astra

www.getastra.com

April 05, 2021

SECURITY AUDIT REPORT FOR MY BUSINESS

hello@getastra.com

Confidential

ATTENTION: This document contains information from Astra IT, Inc. & Czar Securities Pvt. Ltd. that is confidential and privileged. The information is intended for private use of the client. By accepting this document you agree to keep the contents in confidence and not copy, disclose, or distribute this without written request to and written confirmation from Astra IT, Inc. & Czar Securities Pvt. Ltd. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of the contents of this document is prohibited.

COPYRIGHT © ASTRA IT, INC. All rights reserved.

Document Details

Title	Details
COMPLETED ON:	APRIL 02, 2021
REPORT TYPE:	MANUAL SCAN
VALIDITY:	30 DAYS

Table of Contents

1. Executive Summary

This document contains the initial security assessment report for :

{Goldcast Web application and its backend dashboards.}

The purpose of this assessment was to point out security loopholes, business logic errors, and missing best security practices. The tests were carried out assuming the identity of an attacker or a malicious user but no harm was made to the functionality or working of the application/network.

1.1 Scope of Testing

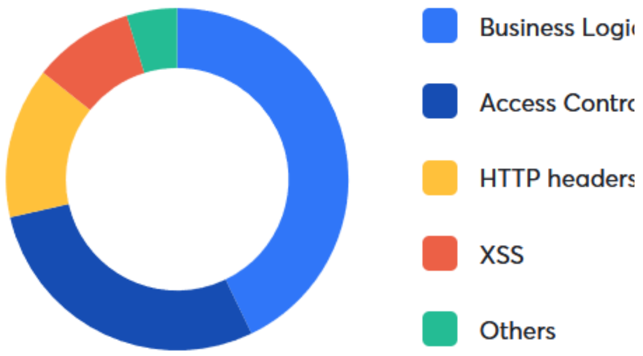
Security assessment includes testing for security loopholes in the scope defined below. Apart from the following, no other information was provided. Nothing was assumed at the start of the security assessment.

The following was the scope covered under the security audit:

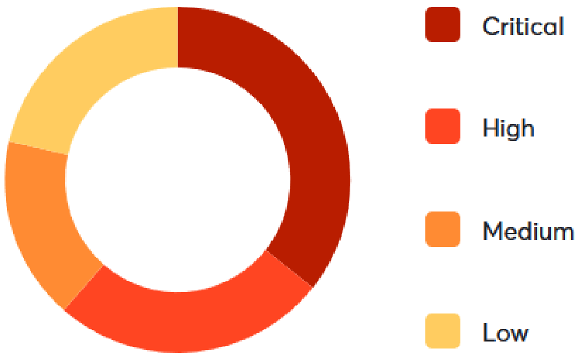
Application 1: {URL1}
Application 2: {URL2}

1.2 Graphical Summary

The below graphical representations from Astra's VAPT dashboard will provide you an overall summary of the security audit scan results, including, vulnerabilities discovered, severity, respective CVSS Score, and other vulnerability details such as its impact, detailed PoC, steps to reproduce, affected URLs/network parameters, and recommended fixes.



Graph 1: Issues Type



Graph 2: Severity Type

1.3 List of Vulnerabilities

#	Vulnerability	Severity	CVSS Score	Status
1	Voluptas voluptates ipsa eos natus.	Low	6	Closed
2	Voluptas voluptates ipsa eos natus.	Medium	7	Closed
3	Voluptas voluptates ipsa eos natus.	Low	5	Closed
4	Voluptas voluptates ipsa eos natus.	High	9	Closed
5	Voluptas voluptates ipsa eos natus.	Low	6	Closed
6	Voluptas voluptates ipsa eos natus.	Medium	7	Closed
7	Voluptas voluptates ipsa eos natus.	Low	5	Closed

Vulnerability Severity	No. of Vulnerability found
Critical	0
High	1
Medium	2
Low	4
Recommendations	0

2. Discovered Vulnerabilities Details

Vulnerability #1

Missing API Security Headers

CVSS Score

5.4

Severity: Medium

Status: Unsolved

Affected URL: **Sitewide**

Details of Vulnerability:

We were able to detect that the following API security headers are missing

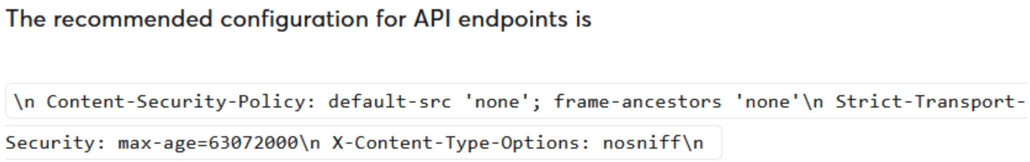
1. Content Security Polic
2. Strict Transport Securit
3. X-Content-Type-Optio

A CSP is an important standard by the W3C that is aimed to prevent a broad range of content injection attacks such as cross-site scripting (XSS), data injection attacks, packet sniffing attacks etc. It is a declarative policy that informs the user agent what are valid sources to load resources from

Impact:

- Missing Content-Type header means that this website could be at risk of a MIME-sniffing attacks.
- Missing Strict Transport Security header means that the application fails to prevent users from connecting to it over unencrypted connections. An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users.

Suggested Fixes:



The recommended configuration for API endpoints is

```
\n Content-Security-Policy: default-src 'none'; frame-ancestors 'none'\n Strict-Transport-
Security: max-age=63072000\n X-Content-Type-Options: nosniff\n
```

Additional References:

https://www.example.com/reference
https://test.com/reference

Vulnerability #2

Stored Cross-Site Scripting (XSS)

CVSS Score

7.7

Severity: High

Status: Resolved

Affected URL:

- http://example.com/test1\n
- https://example.com/test2

Details of Vulnerability:

Stored XSS Vulnerability was found on the affected URLs. This allows an attacker to inject a script which gets stored in the application. When a victim navigates to the affected web page in a browser, the XSS payload will be served as part of the web page. This means that victims will inadvertently end-up executing the malicious script once the page is viewed in a browser.

Impact:

The attacker-supplied code can perform a wide variety of actions, such as

- Stealing the victim's session token
- Stealing Login credential
- Stealing customer Credit Card Information

Suggested Fixes:

- In order to prevent Stored XSS attacks, the best way is to handle the input securely in both client-side and server-side code in a proper manner before it gets stored permanently on the web server.
- Suggested Fix 2

Additional References:

https://www.example.com/reference
https://test.com/reference

Vulnerability #3

SQL Injection

CVSS Score

5

Severity: Medium

Status: Resolved

Affected URL: https://www.example.co/form7

Details of Vulnerability:

SQL injection vulnerabilities arise when user-controllable data is incorporated into database SQL queries in an unsafe manner. An attacker can supply crafted input to break out of the data context in which their input appears and interfere with the structure of the surrounding query.

Steps to reproduce:

The JSON parameter appears to be vulnerable to SQL injection attacks. A single quote was submitted in the JSON parameter, and a general error message was returned. Two single quotes were then submitted and the error message disappeared. You should review the contents of the error message, and the application's handling of other input, to confirm whether a vulnerability is present.### HTTP Requests ###/CRUX/UIDL/

Suggested Fixes:

- The most effective way to prevent SQL injection attacks is to use parameterized queries (also known as prepared statements) for all database access. This method uses two steps to incorporate potentially tainted data into SQL queries: first, the application specifies the structure of the query, leaving placeholders for each item of user input; second, the application specifies the contents of each placeholder. Because the structure of the query has already been defined in the first step, it is not possible for malformed data in the second step to interfere with the query structure. You should review the documentation for your database and application platform to determine the appropriate APIs which you can use to perform parameterized queries.
- Suggested Fix 2

Additional References:

https://www.example.com/reference
https://test.com/reference

Vulnerability #4

Incorrect Contructor Name

CVSS Score

6.3

Severity: Medium

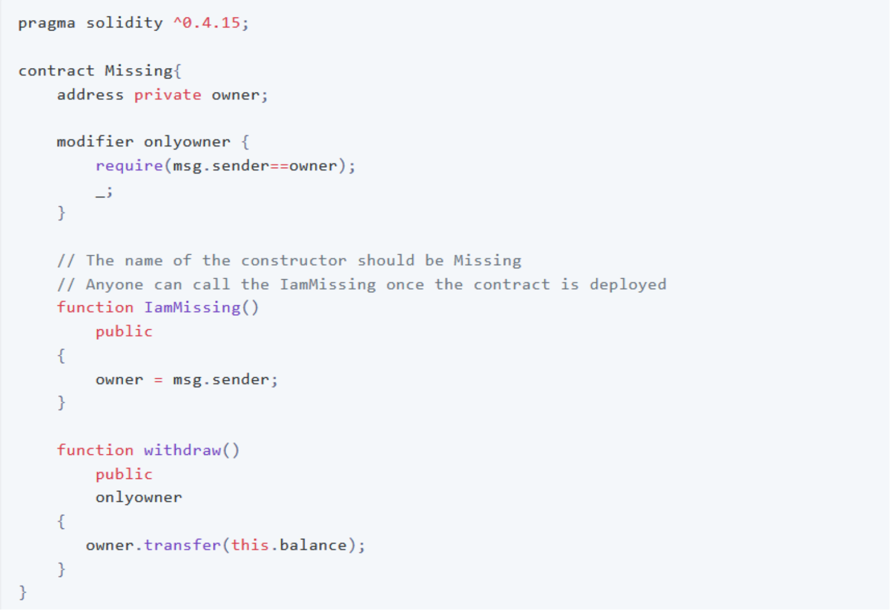
Status: Unsolved

Affected URL: https://www.hacked.co

Details of Vulnerability:

Constructors are special functions that are called only once during the contract creation. They often perform critical, privileged actions such as setting the owner of the contract. Before Solidity version 0.4.22, the only way of defining a constructor was to create a function with the same name as the contract class containing it. A function meant to become a constructor becomes a normal, callable function if its name doesn't exactly match the contract name. This behavior sometimes leads to security issues, in particular when smart contract code is re-used with a different name but the name of the constructor function is not changed accordingly.

Steps to reproduce:



```
pragma solidity ^0.4.15;

contract Missing{
    address private owner;

    modifier onlyowner {
        require(msg.sender==owner);
        _;
    }

    // The name of the constructor should be Missing
    // Anyone can call the IamMissing once the contract is deployed
    function IamMissing()
        public
    {
        owner = msg.sender;
    }

    function withdraw()
        public
        onlyowner
    {
       owner.transfer(this.balance);
    }
}
```

Suggested Fixes:

Solidity version 0.4.22 introduces a new constructor keyword that make a constructor definitions clearer. It is therefore recommended to upgrade the contract to a recent version of the Solidity compiler and change to the new constructor declaration.

Additional References: https://swcregistry.io/docs/SWC-118

3. List of VAPT Tests Performed

The following lists of tests are suggestive & not limited to the ones listed. Most importantly, every test case has multiple sub-test cases ranging from a few to sometimes 1000+ sub tests.

Additional test cases will be performed based on factors such as:

1. Technology Stack
2. Server Side Programming Language, Front-end frameworks
3. Framework/CMS/APIs
4. Type of application (Payment integrations, external integrations)

3.1 OWASP Top 10

#	OWASP Top 10
	for **Web Applications**
1	SQL Injection
2	Broken Authentication
3	Sensitive Data Exposure
4	XML External Entities (XXL)
5	Broken Access Control
6	Security Misconfiguration
7	Cross-Site Scripting (XSS)
8	Insecure Deserialization
9	Using Components with Known Vulnerabilities
10	Insufficient Logging and Monitoring
	for **Mobile Applications**
1	Improper Platform Usage
2	Insecure Data Storage
3	Insecure Communication
4	Insecure Authentication
5	Insufficient Cryptography
6	Insecure Authorization
7	Client Mode Quality
8	Code Tampering
9	Reverse Engineering
10	Extraneous Functionality

3.2 SANS 25 Software Errors/Tests

#	SANS 25
1	Improper Restriction of Operations within the Bounds of a Memory Buffer
2	Improper Neutralization of Input During Web Page Generation ('XSS')
3	Improper Input Validation
4	Information Exposure
5	Out-of-bounds Read
6	Improper Neutralization of Special Elements used in an SQL Command (SQLi)
7	Use After Free
8	Integer Overflow or Wraparound
9	Cross-Site Request Forgery (CSRF)
10	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
11	Improper Neutralization of Special Elements used in an OS Command
12	Out-of-bounds Write
13	Improper Authentication
14	NULL Pointer Dereference
15	Incorrect Permission Assignment for Critical Resource
16	Unrestricted Upload of File with Dangerous Type
17	Improper Restriction of XML External Entity Reference
18	Improper Control of Generation of Code ('Code Injection')
19	Use of Hard-coded Credentials
20	Uncontrolled Resource Consumption
21	Missing Release of Resource after Effective Lifetime
22	Untrusted Search Path
23	Deserialization of Untrusted Data
24	Improper Privilege Management
25	Improper Certificate Validation

3.3 174 Other Test Cases

#	Other Tests	Typical Severity
1	OS Command Injection	High
2	SQL Injection (Second Order)	High
3	XML External Entity Injection	High
4	LDAP Injection	High
5	XPath Injection	High
6	XML Injection	High
7	ASP.NET Debugging Enabled	High
8	DoS Locking Customer Accounts	Medium
9	DoS Buffer Overflows	Medium
10	Storing too much data in session (DoS)	High
11	Writing user-provided data to disk (DoS)	High
12	HTTP Insecure methods available on Server	High
13	Out of band resource load (HTTP)	High
14	File path manipulation	High
15	Server-site JavaScript code injection	High
16	Perl code injection	High
17	Ruby code injection	High
18	Python code injection	High
19	Expression Language injection	High
20	Unidentified code injection	High
21	Server-side template injection	High
22	SSL injection	High
23	Stored XSS	High
24	HTTP response header injection	High
25	Reflected XSS	High
26	Client-side template injection	High
27	DOM-based XSS	High
28	Reflected DOM-based XSS	High
29	Stored DOM-based XSS	High
30	DOM-based JavaScript Injection	High
31	Reflected DOM-based JavaScript Injection	High
32	Stored DOM-based JavaScript Injection	High
33	Path-relative style sheet import	Information
34	Client-side SQLi (DOM-based)	High
35	Client-side SQLi (Reflected DOM-based)	High
36	Client-side SQLi (Stored DOM-based)	High

#	Other Test performed	Typical Severity
37	WebSocket Hijacking (DOM-based)	High
38	WebSocket Hijacking (Reflected DOM-based)	High
39	WebSocket Hijacking (Stored DOM-based)	High
40	Local Path Manipulation (DOM-based)	High
41	Local Path Manipulation (Reflected DOM)	High
42	Local Path Manipulation (Stored DOM-based)	High
43	Client-side XPATH Injection (DOM-based)	Low
44	Client-side XPATH Injection (Reflected DOM)	Low
45	Client-side XPATH Injection (Stored DOM)	Low
46	Client-side JSON Injection (DOM-based)	Low
47	Client-side JSON Injection (Reflected DOM)	Low
48	Client-side JSON Injection (Stored DOM-based)	Low
49	Flash cross-domain policy	High
50	Cross-origin resource sharing	Information
51	Cross-origin resource sharing (arbitrary)	High
52	Cross-origin resource sharing (encrypted)	Low
53	Cross-origin resource sharing (all sub-domains)	Low
54	Cross-site Request Forgery (CSRF)	Medium
55	SMTP header injection	Medium
56	Cleartext submission of password	High
57	External service interaction (DNS)	High
58	External service interaction (HTTP)	High
59	External service interaction (SMTP)	Information
60	Referrer dependent response	Information
61	Spoofable client IP address	Information
62	User-agent dependent response	Information
63	Password returned in a later response	Medium
64	Password submitted using GET method	Low
65	Password returned in URL query string	Low
66	SQL statement in request parameter	Medium
67	Cross-domain POST	Information
68	ASP.NET ViewState without MAC Enabled	Low
69	XML entity expansion	Medium
70	Long redirection response	Information
71	Serialized object in HTTP message	High
72	Duplicate cookies set	Information

#	Other Test performed	Typical Severity
73	WebSocket Hijacking (DOM-based)	High
74	WebSocket Hijacking (Reflected DOM-based)	High
75	WebSocket Hijacking (Stored DOM-based)	High
76	Local Path Manipulation (DOM-based)	High
77	Local Path Manipulation (Reflected DOM)	High
78	Local Path Manipulation (Stored DOM-based)	High
79	Client-side XPATH Injection (DOM-based)	Low
80	Client-side XPATH Injection (Reflected DOM)	Low
81	Client-side XPATH Injection (Stored DOM)	Low
82	Client-side JSON Injection (DOM-based)	Low
83	Client-side JSON Injection (Reflected DOM)	Low
84	Client-side JSON Injection (Stored DOM-based)	Low
85	Flash cross-domain policy	High
86	Cross-origin resource sharing	Information
87	Cross-origin resource sharing (arbitrary)	High
88	Cross-origin resource sharing (encrypted)	Low
89	Cross-origin resource sharing (all sub-domains)	Low
90	Cross-site Request Forgery (CSRF)	Medium
91	SMTP header injection	Medium
92	Cleartext submission of password	High
93	External service interaction (DNS)	High
94	External service interaction (HTTP)	High
95	External service interaction (SMTP)	Information
96	Referrer dependent response	Information
97	Spoofable client IP address	Information
98	User-agent dependent response	Information
99	Password returned in a later response	Medium
100	Password submitted using GET method	Low
101	Password returned in URL query string	Low
102	SQL statement in request parameter	Medium
103	Cross-domain POST	Information
104	ASP.NET ViewState without MAC Enabled	Low
105	XML entity expansion	Medium
106	Long redirection response	Information
107	Serialized object in HTTP message	High
108	Duplicate cookies set	Information

#	Other Test performed	Typical Severity
109	Input returned in response (stored)	Information
110	Input returned in response (reflected)	Information
111	Suspicious input transformation (reflected)	Information
112	Suspicious input transformation (stored)	Information
113	Open redirection (stored)	Low
114	Open redirection (reflected)	Medium
115	Open redirection (DOM-based)	Low
116	Open redirection (Stored DOM-based)	Low
117	Open redirection (Reflected DOM-based)	Medium
118	SSl cookie without secure flag set	Medium
119	Cookie scoped to parent domain	Low
120	Cross-domain referrer leakage	Information
121	Cross-domain script include	Information
122	Cookie without HTTPOnly flag set	Low
123	Session token in URL	Medium
124	Password field with autocomplete enabled	Low
125	Password value set in cookie	Medium
126	Browser cross-site scripting disabled	Infomration
127	HTTP TRACE method is enabled	Information
128	Cookie manipulation (DOM-based)	Low
129	Cookie manipulation (reflected DOM-based)	Low
130	Cookie manipulation (DOM-based)	Low
131	Ajax request header manipulation (DOM-based)	Low
132	Ajax request header manipulation (reflected)	Low
133	Ajax request header manipulation (stored DOM)	Low
134	Denial of service (DOM-based)	Information
135	Denial of service (reflected DOM-based)	Information
136	Denial of service (stored DOM-based)	Low
137	HTML5 web message manipulation DOM-based	Information
138	HTML5 web message manipulation (reflected)	Information
139	HTML5 web message manipulation (stored DOM)	Information
140	HTML5 storage manipulation (DOM-based)	Information
141	HTML5 storage manipulation (reflected DOM)	Information
142	HTML5 storage manipulation (stored DOM)	Information
143	Link manipulation (DOM-based)	Low

#	Other Test performed	Typical Severity
144	Link manipulation (reflected DOM-based)	Low
145	Link manipulation (stored DOM-based)	Low
146	Link manipulation (reflected & stored)	Information
147	Document domain manipulation (DOM-based)	Medium
148	Document domain manipulation reflected DOM	Medium
149	Document domain manipulation (stored DOM)	Medium
150	DOM data manipulation (DOM-based)	Information
151	CSS Injection (reflected & stored)	Medium
152	Client-side HTTP parameter pollution (reflected)	Low
153	Client-side HTTP parameter pollution (Stored)	Low
154	Form action hijacking (reflected)	Medium
155	Form action hijacking (stored)	Medium
156	Database connection string disclosed	Medium
157	Source code disclosure	Low
158	Directory listing	Information
159	Email addresses disclosed	Information
160	Private IP addresses disclosed	Information
161	Social security numbers disclosed	Information
162	Credit card numbers disclosed	Information
163	Private key disclosed	Information
164	Cacheable HTTPS response	Information
165	Base64 encoded data in parameter	Information
166	Multiple content types specified	Information
167	HTML does not specify charset	Information
168	HTML uses unrecognized charset	Information
169	Content type incorrectly stated	Low
170	Content ty[e is not specified	Information
171	SSL certificate	Medium
172	Unencrypted communications	Low
173	Strict transport security not enforced	Low
174	Mixed content	Information

3.4 Server - Level Test Cases

Server - Level Testing	
Information Gathering	
1	Fingerprint Web Server
2	Test Network/Infrastructure Configuration
3	Test HTTP Methods
4	Test HTTP Strict Transport Security (HSTS)
5	Testing for Cookies Attributes
6	Test RIA Cross-domain Policy
SSL/TLS Testing	
7	HeartBleed
8	POODLE SSL Vulnerability
9	ChangeCipherSpec Injection
10	BREACH
11	BEAST
12	Forward Secrecy Support
13	RC4 Support
14	CRIME & Time Vulnerabilities
15	Lucky13
16	HSTS: Check for header
17	HSTS: Reasonable duration of MAX-AGE
18	HSTS: Check for SubDomains support
19	Certificate expiration
20	Insufficient public key length

21	Host Name mismatch
22	Weak/Insecure Hashing Algorithm
23	SSLv2 support
24	Weak ciphers check (Low, Anon, Null, Export)
25	Null prefix in the certificate
26	HTTPS stripping
27	SurfJacking
28	Non-SSL elements/content embedded in SSL Page
29	Cache control

Configuration and Deploy Management Testing

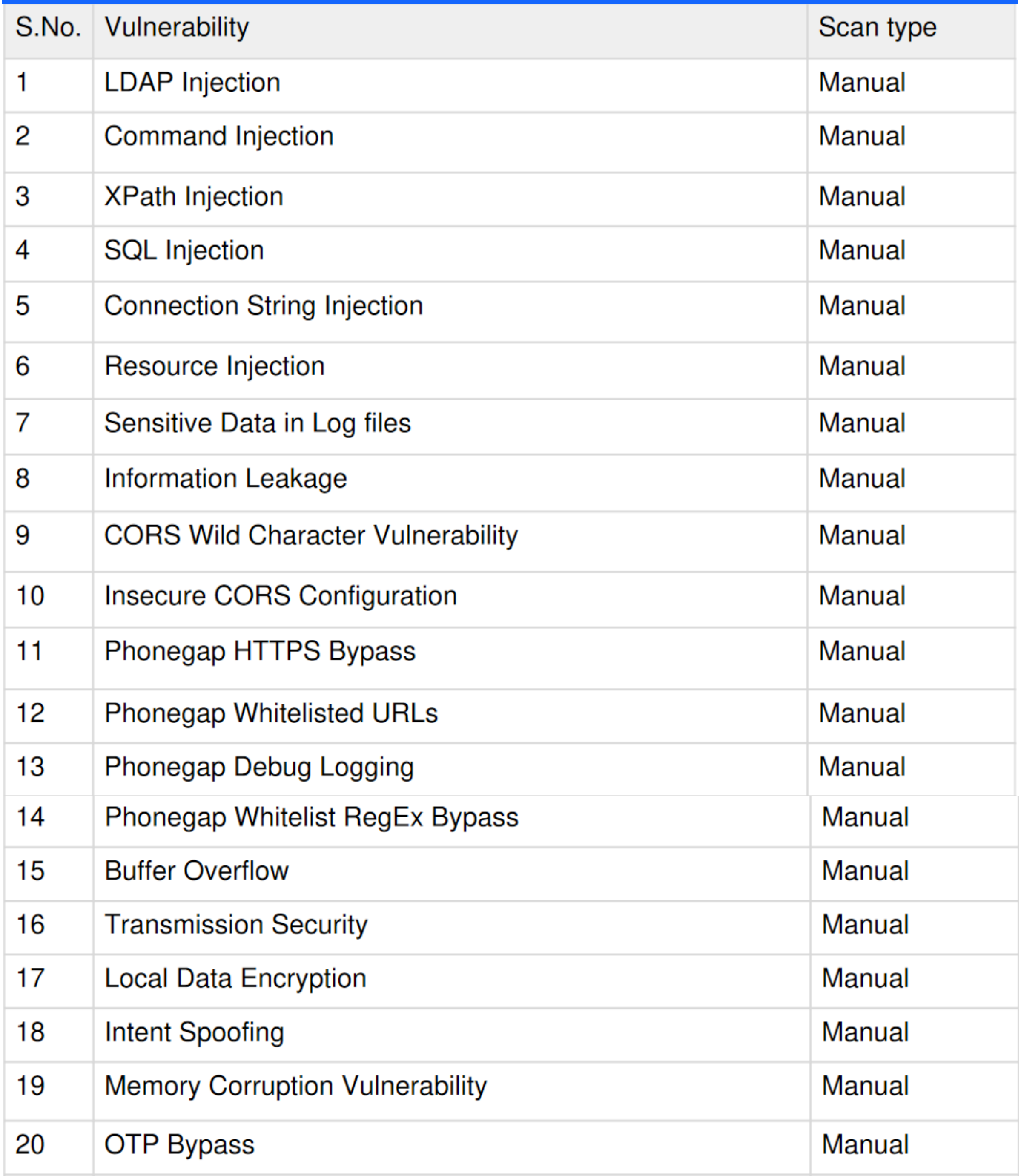
30	Test Network/Infrastructure Configuration
31	Test HTTP Methods
32	Test HTTP Strict Transport Security
33	Testing for Cookies Attributes
34	Test RIA cross domain policy

Cryptography

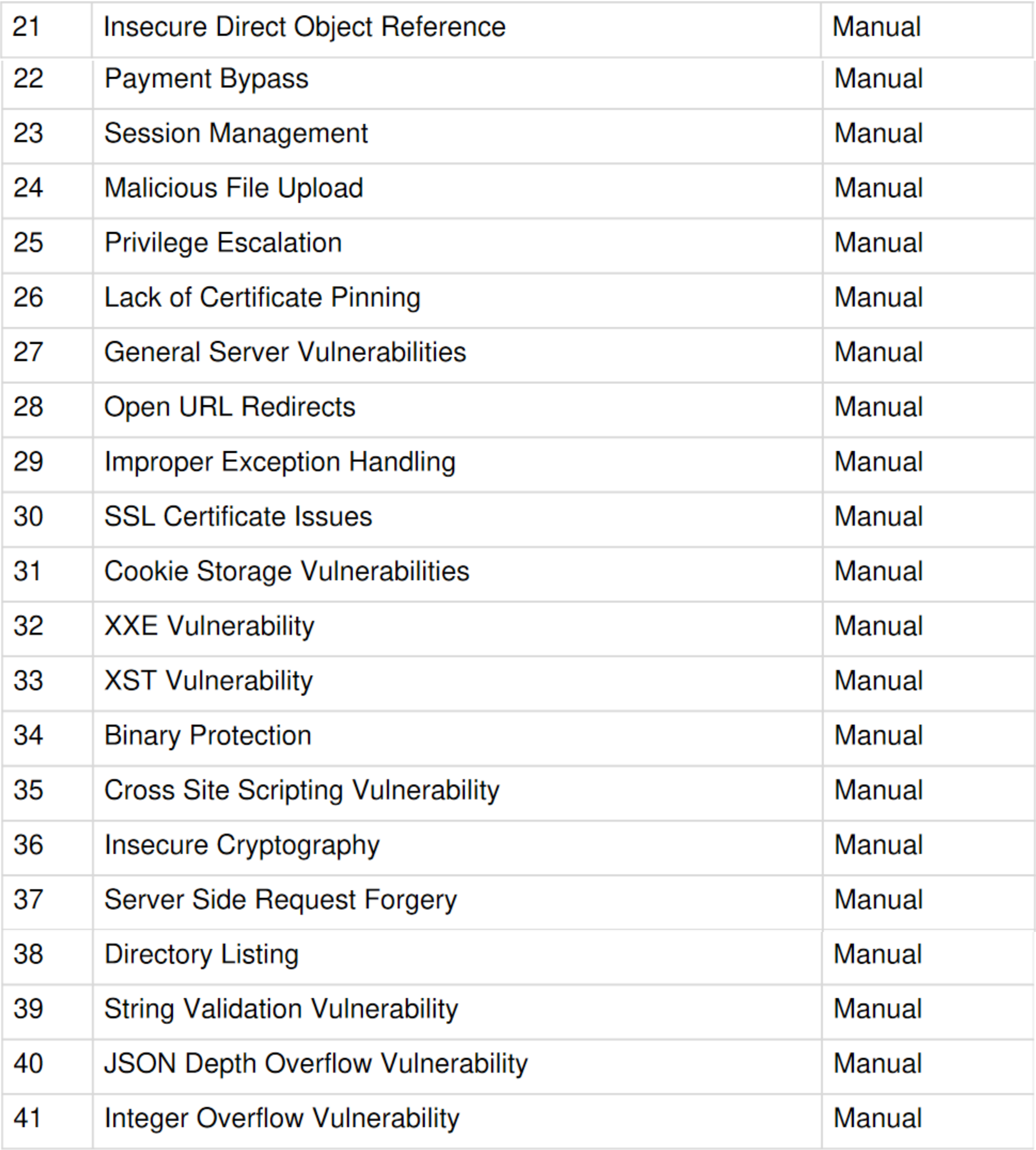
35	Testing of Weak SSL/TLS Ciphers, Insufficient Transport layer
36	Test HTTP Methods
37	Test HTTP Strict Transport Security

3.5 Test Cases for Windows

Test Cases for Windows



S.No.	Vulnerability	Scan type
1	LDAP Injection	Manual
2	Command Injection	Manual
3	XPath Injection	Manual
4	SQL Injection	Manual
5	Connection String Injection	Manual
6	Resource Injection	Manual
7	Sensitive Data in Log files	Manual
8	Information Leakage	Manual
9	CORS Wild Character Vulnerability	Manual
10	Insecure CORS Configuration	Manual
11	Phonegap HTTPS Bypass	Manual
12	Phonegap Whitelisted URLs	Manual
13	Phonegap Debug Logging	Manual
14	Phonegap Whitelist RegEx Bypass	Manual
15	Buffer Overflow	Manual
16	Transmission Security	Manual
17	Local Data Encryption	Manual
18	Intent Spoofing	Manual
19	Memory Corruption Vulnerability	Manual
20	OTP Bypass	Manual



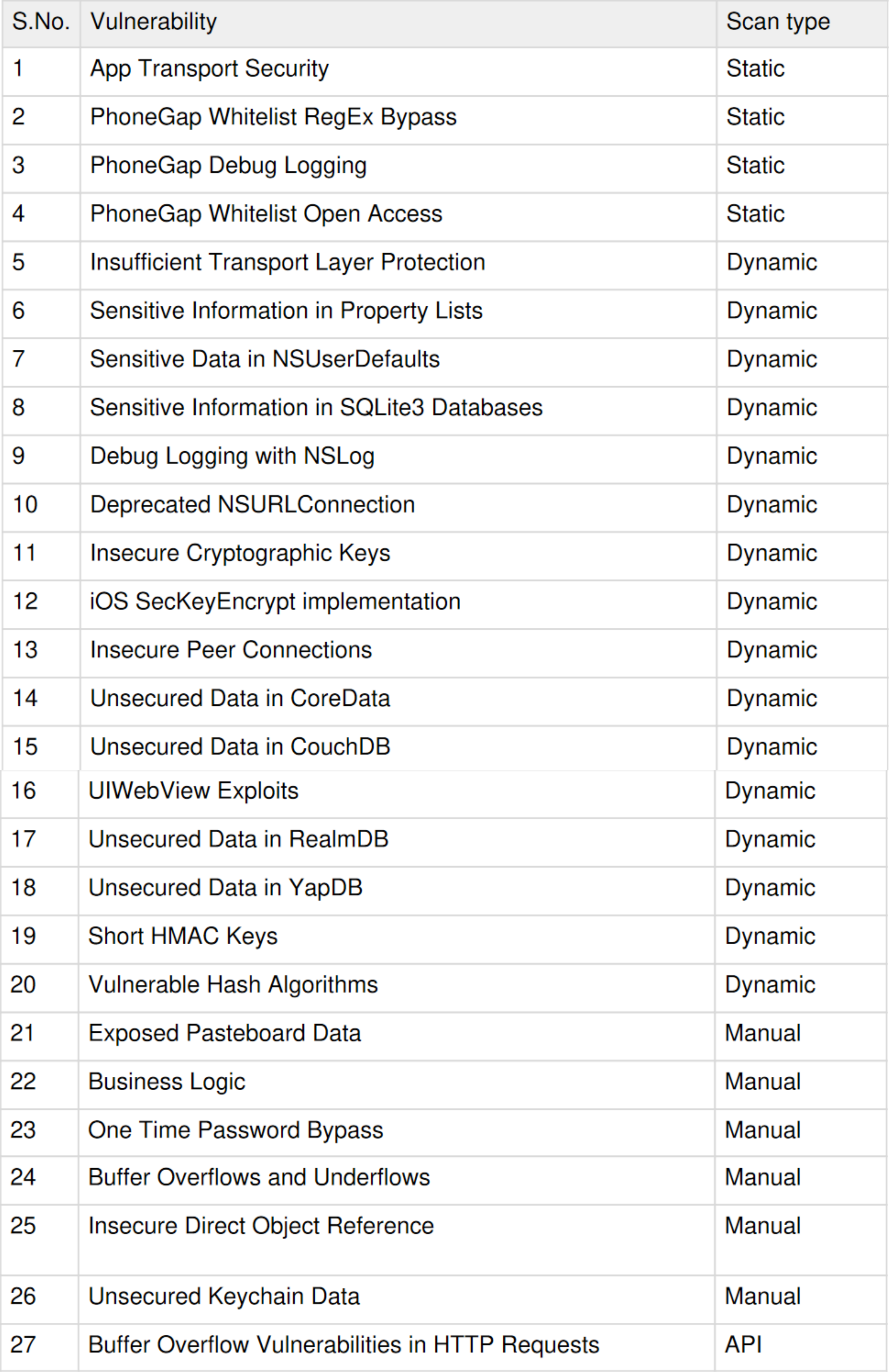
21	Insecure Direct Object Reference	Manual
22	Payment Bypass	Manual
23	Session Management	Manual
24	Malicious File Upload	Manual
25	Privilege Escalation	Manual
26	Lack of Certificate Pinning	Manual
27	General Server Vulnerabilities	Manual
28	Open URL Redirects	Manual
29	Improper Exception Handling	Manual
30	SSL Certificate Issues	Manual
31	Cookie Storage Vulnerabilities	Manual
32	XXE Vulnerability	Manual
33	XST Vulnerability	Manual
34	Binary Protection	Manual
35	Cross Site Scripting Vulnerability	Manual
36	Insecure Cryptography	Manual
37	Server Side Request Forgery	Manual
38	Directory Listing	Manual
39	String Validation Vulnerability	Manual
40	JSON Depth Overflow Vulnerability	Manual
41	Integer Overflow Vulnerability	Manual

3.6 Test Cases for Android and iOS

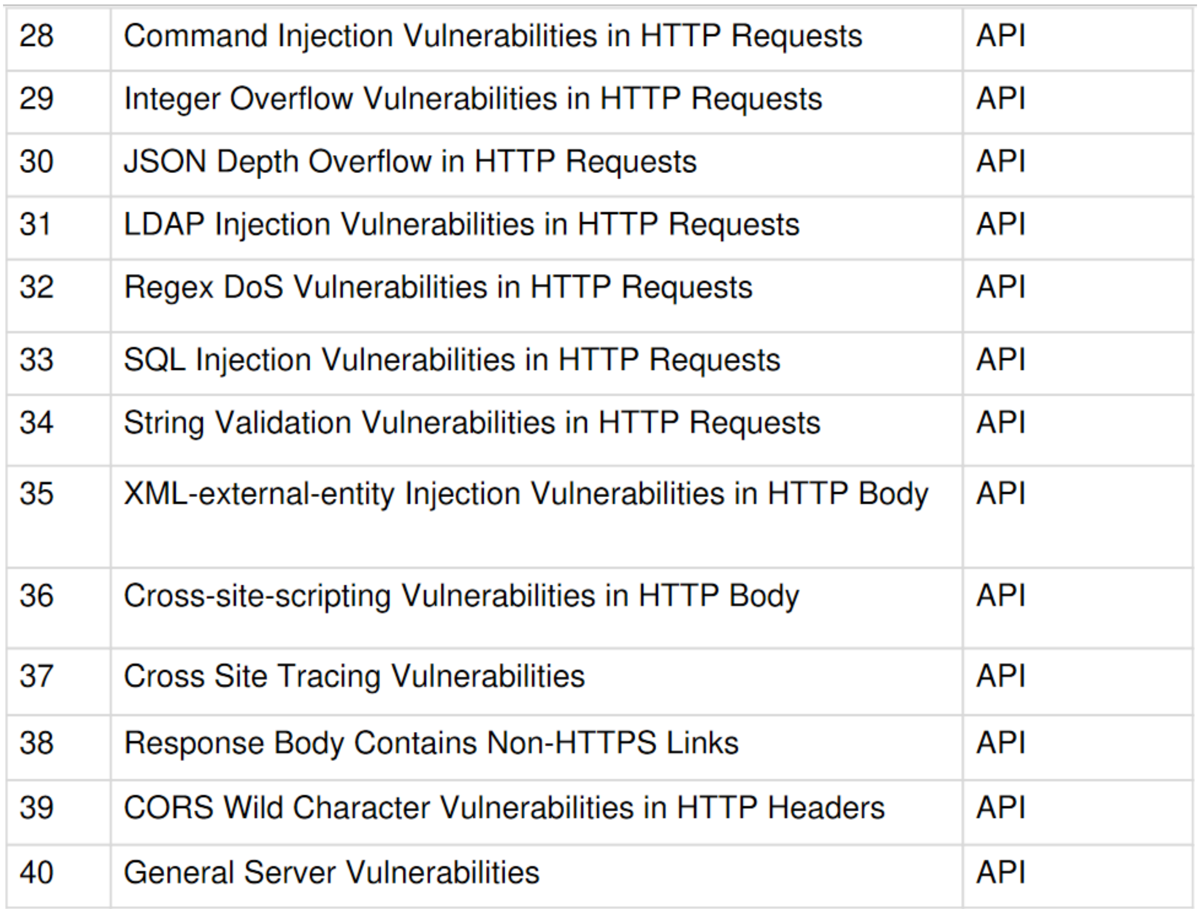
Test Cases for Android		
S.No.	Vulnerability	Scan type
1	Unprotected Services	Static
2	Improper Content Provider Permissions	Static
3	Improper Custom Permissions	Static
4	Remote URL Redirection Vulnerability	Static
5	PhoneGap Error URL Redirection Vulnerability	Static
6	PhoneGap HTTPS Bypass Vulnerability	Static
7	Cordova Remote Start Page Manipulation Vulnerability	Static
8	Connection to External Redis Server	Static
9	Unprotected Exported Activities	Static
10	Unprotected Exported Receivers	Static
11	Unprotected Exported Service	Static
12	Unprotected Exported Provider	Static
13	Non-signature Protected Exported Activities	Static
14	Non-signature Protected Exported Receivers	Static
15	Non-signature Protected Exported Services	Static
16	Non-signature Protected Exported Providers	Static
17	Content Provider File Traversal Vulnerability	Static
18	Broken SSL Trust Manager	Static
19	Broken HostnameVerifier for SSL	Static
20	Insecure SSLSocketFactories	Static
21	HostnameVerifier Allowing All Hostnames	Static
22	App Extending WebViewClient	Static
23	PhoneGap HTTPS Whitelist Bypass	Static
24	PhoneGap Whitelisted URLs	Static
25	JavascriptInterface Remote Code Execution	Static
26	AddressBook Expose Vulnerability	Static

27	SSL Pinning Detection	Static
28	Surreptious Sharing Vulnerability	Static
29	Webview Fileschema Vulnerability	Static
30	Fragment Injection Vulnerability	Static
31	WebView Exploits	Dynamic
32	Unused Permissions	Static
33	PhoneGap JavaScript Injection	Static
34	Application Debugging	Static
35	Application Logs	Static / Dynamic
36	PhoneGap Debug Logging	Static
37	Storing Information in Shared Preferences	Dynamic
38	Derived Crypto Keys	Static / Dynamic
39	Buffer Overflow Vulnerabilities in HTTP Requests	API
40	Command Injection Vulnerabilities in HTTP Requests	API
41	Integer Overflow Vulnerabilities in HTTP Requests	API
42	JSON Depth Overflow in HTTP Requests	API
43	LDAP Injection Vulnerabilities in HTTP Requests	API
44	Regex DoS Vulnerabilities in HTTP Requests	API
45	SQL Injection Vulnerabilities in HTTP Requests	API
46	String Validation Vulnerabilities in HTTP Requests	API
47	XML-external-entity Injection Vulnerabilities in HTTP Body	API
48	Cross-site-scripting Vulnerabilities in HTTP Body	API
49	Cross Site Tracing Vulnerabilities	API
50	Response Body Contains Non-HTTPS Links	API
51	CORS Wild Character Vulnerabilities in HTTP Headers	API
52	General Server Vulnerabilities	API
53	Misconfigured AWS S3 Buckets	Manual
54	Business Logic	Manual
55	One Time Password Bypass	Manual
56	Insecure Direct Object Reference	Manual

Test Cases for iOS



S.No.	Vulnerability	Scan type
1	App Transport Security	Static
2	PhoneGap Whitelist RegEx Bypass	Static
3	PhoneGap Debug Logging	Static
4	PhoneGap Whitelist Open Access	Static
5	Insufficient Transport Layer Protection	Dynamic
6	Sensitive Information in Property Lists	Dynamic
7	Sensitive Data in NSUserDefaults	Dynamic
8	Sensitive Information in SQLite3 Databases	Dynamic
9	Debug Logging with NSLog	Dynamic
10	Deprecated NSURLConnection	Dynamic
11	Insecure Cryptographic Keys	Dynamic
12	iOS SecKeyEncrypt implementation	Dynamic
13	Insecure Peer Connections	Dynamic
14	Unsecured Data in CoreData	Dynamic
15	Unsecured Data in CouchDB	Dynamic
16	UIWebView Exploits	Dynamic
17	Unsecured Data in RealmDB	Dynamic
18	Unsecured Data in YapDB	Dynamic
19	Short HMAC Keys	Dynamic
20	Vulnerable Hash Algorithms	Dynamic
21	Exposed Pasteboard Data	Manual
22	Business Logic	Manual
23	One Time Password Bypass	Manual
24	Buffer Overflows and Underflows	Manual
25	Insecure Direct Object Reference	Manual
26	Unsecured Keychain Data	Manual
27	Buffer Overflow Vulnerabilities in HTTP Requests	API



28	Command Injection Vulnerabilities in HTTP Requests	API
29	Integer Overflow Vulnerabilities in HTTP Requests	API
30	JSON Depth Overflow in HTTP Requests	API
31	LDAP Injection Vulnerabilities in HTTP Requests	API
32	Regex DoS Vulnerabilities in HTTP Requests	API
33	SQL Injection Vulnerabilities in HTTP Requests	API
34	String Validation Vulnerabilities in HTTP Requests	API
35	XML-external-entity Injection Vulnerabilities in HTTP Body	API
36	Cross-site-scripting Vulnerabilities in HTTP Body	API
37	Cross Site Tracing Vulnerabilities	API
38	Response Body Contains Non-HTTPS Links	API
39	CORS Wild Character Vulnerabilities in HTTP Headers	API
40	General Server Vulnerabilities	API

3.7 Tests Cases for Cloud (AWS, Azure, GCP, and Other)

#	Test Cases for Cloud Services
1	Test for Unauthenticated database access
2	Test for Improper permissions for Database
3	Test for compromising access keys
4	Test for extracting keys from a VM / instance
5	Test for exploits due to improper configs.
6	Testing for public exploits in VM / instances
7	Test for backdoors exploitation internally
8	Test for Subdomain Takeover
9	Test for access mgmt. Privilege Escalation
10	Test for Remote Code Execution (RCE)
11	Test for Role Enumeration
12	Test for VM service Privilege Escalation
13	Test for IAM Enumeration
14	Test for BitBucket Server Data for credentials
15	Test for cloud compromise by DNS rebinding
16	Test for local Windows/Linux logs change
17	Test for loopholes that add root certificates and SSH private keys to VMs and users
18	Test for loopholes that assign a secondary private IP address to an instance / VM when you launch the instance / VM
19	Test for unauthenticated obtaining of the VM images from storage accounts and do an analysis for passwords, keys, certificates to penetrate and access live resources
20	Test for penetrating OS-level access to Instances/VMs via Workload
21	Test for Management Service Privileges
22	Test to run or deploy a workload with an assigned service or role and export instance credentials for those privileges
23	Test for server and application versions & frameworks fingerprinting and detect exposed sensitive PII in server/application logs
24	Test for CSV injection
25	Test for MITM attack penetration on Elastic Load Balancer (ELB) for session hijacking
26	Test for credential stealing attack on credentials
27	Test for credential stealing attack on cloud workload
28	Test for credential stealing attack on operation of a cloud key management service

29 Test to alter data in datastore for fraudulent transactions or static website compromise

30 Test to alter a serverless function, logic app or otherwise a business logic implementation for action on objective or escalation

31 Test to alter a DNS Record record set in a trusted zone and/or certificates for the resource record set to divert traffic, create phishing sites & abuse the brand (AWS ACM, AWS Route53, Azure DNS Service)

32 Test to alter data in local SQL or MySQL databases

33 Operate in regions where logging is not enabled or disable global logging (like CloudTrail)

34 Test to alter log files in a non-validated log store or disable validation (like CloudTrail Log Validation)

35 Test for Disable network traffic analysis/logging (VPC Flow Logs)

36 Test for Disable Cloud Alerting to prevent detection and response

37 Test for Disable data store access logging to prevent detection and response (CloudTrail Data Access, S3 Access Logging, etc.)

38 Test to alter log retention or damage the integrity of logs (S3 lifecycle, KMS decryption, CMK key deletion/role privilege lockout)

39 Process hooking, process injection, Windows access token manipulation, leveraging misconfigured sudo capabilities

40 Test to create or reset a login, access key, or temporary credential belonging to a high privilege user (like IAM: CreateAccessKey, STS, or IAM: UpdateLoginProfile)

41 Test to Change the default policy for a user or new users to include additional privileges (like Set-Default-Policy-Version)

42 Leverage data or code pipelines to execute operations on behalf of their assumed roles (AWS data pipeline Shell-Command-Activity, inject python code into a pickle celery SQS queue)

3.7 Tests Cases for Blockchain

#	Test Cases for Cloud Services
1	Test for Unencrypted Private Data On-Chain
2	Test for Code With No Effects
3	Test for Message call with hardcoded gas amount
4	Test for Unexpected Ether balance
5	Test for Hash Collisions With Multiple Variable Length Arguments
6	Testing for Presence of unused variables
7	Test for Right-To-Left-Override control character (U+202E)
8	Test for Typographical Error
9	Test for DoS With Block Gas Limit
10	Test for Arbitrary Jump with Function Type Variable
11	Test for Insufficient Gas Griefing
12	Test for Incorrect Inheritance Order
13	Test for Writing to Arbitrary Storage Location
14	Test for Requirement Violation
15	Test for Lack of Proper Signature Verification
16	Test for local Missing Protection against Signature Replay Attacks
17	Test for Weak Sources of Randomness from Chain Attributes
18	Test for Shadowing State Variables
19	Test for Incorrect Constructor Name
20	Test for Signature Malleability
21	Test for Blocking values as a proxy for time
22	Test for Authorization through tx.origin
23	Test for Transaction Order Dependence
24	Test for DoS with Failed Call
25	Test for Delegatecall to Untrusted Callee
26	Test for Use of Deprecated Solidity Functions
27	Test for Assert Violation
28	Test for Uninitialized Storage Pointer
29	Test for State Variable Default Visibility
30	Test for Reentrancy
31	Test for Unprotected SELFDESTRUCT Instruction
32	Test for Unprotected Ether Withdrawal
33	Test for Unchecked Call Return Value
34	Test for Floating Pragma
35	Test for Outdated Compiler Version
36	Test for Integer Overflow and Underflow
37	Test for Function Default Visibility

astra

Your plug & play cyber security suite.

Questions? Contact us at

hello@getastra.com