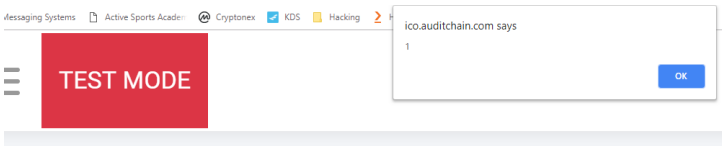
High	Persistent Cross Site Scripting
	This Persistent XSS allows malicious code to be submitted to a web site where it's stored for a period in this case the bounties board. The unsuspecting user is not required to interact with any additional site or link as the javascript is executed by simply viewing the web page containing the code.
Test Payload	**</textarea><script>alert(1);</script><textarea>**
Sample Vulnerable URL	https://ico.auditchain.com/SecretAdmin/EditPage/11
CVSS v3.0 Score	**8.5**
Solution	Assume all input is malicious. Use an "accept known good" input validation strategy via a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications by transforming it into something that does.

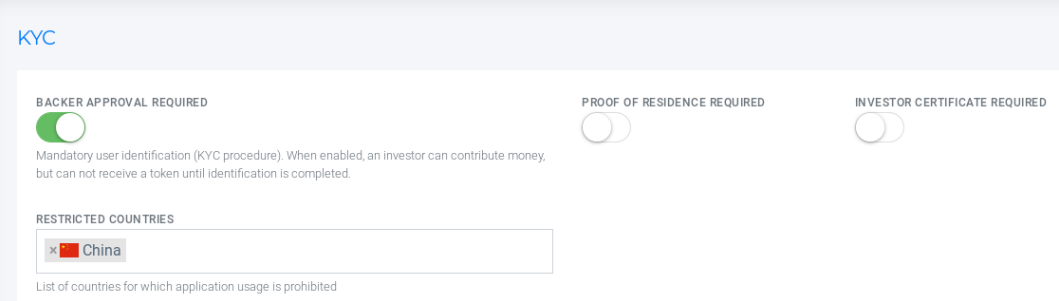
High	Reflected Cross Site Scripting
Description	This Reflected XSS will allow the attacker-supplied javascript to run in the victim's browser instance and compromise the trust relationship between a user and the web site. The attacker can hook into the user's browser within the security zone of the hosting web site and depending on the browser version can read, modify and transmit any sensitive data accessible by the browser. The most likely scenario is that the victim has their cookie stolen after which their browser is redirected to legitimate content encapsulated with the iframe of the originally vulnerable site so that the attacker can maintain control. This is a non-persistent attack that requires the user to visit a specially crafted link laced with malicious code to mount the attack and have a field in the victim page submit automatically without the user's knowledge. Upon clicking the malicious link, the XSS payload gets echoed back and get interpreted by the user's browser and execute.
Test Payload	**javascript:alert(1);**
Sample Vulnerable URL	https://ico.auditchain.com/Bounties/Edit/1143
CVSS v3.0 Score	**8.0**
Solution	Perform input validation by considering length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. Ensure input validation at well-defined interfaces within the application to protect the application even if a component is reused or moved elsewhere.

Screenshot



Persistent XSS is no fun for anyone!

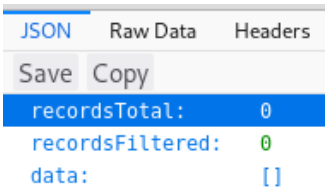
javascript:alert(1);

High	Path Traversal
Description	This Path Traversal allows an attacker access to files and directories that reside outside the web document root directory such as the KYC requirements. A malicious user can utilize special-characters sequences to alter the resource location requested in the URL and tamper with the ICO parameters, for example enabling funds to be collected from United States and create a regulatory risk.
Test Payload	**/Settings/KYC?lang=KYC**
Sample Vulnerable URL	https://ico.auditchain.com/Settings/KYC?lang=KYC
CVSS v3.0 Score	**7.5**
Solution	Perform input validation to accept values, sanitizes superfluous inputs and ensures syntactical conformance to business rules as per stringent whitelist specifications.
Screenshot	

High	SQL Injection
Description	This SQLi was detected by successfully retrieving more data than originally returned, by manipulating the parameter and crafting a customized request to the database which may in the future allow an attacker to compromise the confidentiality of records in the database.
Test Payload	**false OR 1=1 --**
Sample Vulnerable URL	https://tinyurl.com/y7qorpl9 (Tiny URL)
CVSS v3.0 Score	**6.8**
Solution	Escape characters from all client-side inputs, prevent simple concatenation of strings into queries, perform server-side data checks and grant minimum database access to the application
Screenshot	

High	MIME Sniffing
Description	The website allows an attacker to leverage MIME sniffing to determine how a browser renders files in this case allowing the browser to render out json data as output.
Sample Vulnerable URL	**https://tinyurl.com/ybzylhkj (Tiny URL)**
CVSS v3.0 Score	5.1
Solution	**Ensure that the web server sets the X-Content-Type-Options header to 'nosniff' for all web pages**

Screenshot



High	Unsecured Cookie Transmission
Description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
Test Payload	**TwoFactorCookie**
Sample Vulnerable URL	Set-Cookie: .AspNet.TwoFactorCookie
CVSS v3.0 Score	**3.1**
Solution	Put <httpCookies requireSSL="true" /> in <system.web>

High	Password auto-complete is enabled
Description	If an investor chooses to save, data entered in these fields will be cached by the browser. An attacker who can access the victim's browser could steal this information especially if the website has been accessed from a shared computer in a cyber cafe or airport terminal
Sample Vulnerable URL	**<input class="form-control" data-val="true" data-val-required="Password required" id="Password" name="Password" required="true" type="password"/>**
CVSS v3.0 Score	1.8
Solution	**Add the attribute autocomplete="off" to the form tag or to individual "input" fields.**

High	Dysfunctional Checkbox
Description	The check boxes on both Registration and Log In forms are dysfunctional as the tick mark is not visible on click. Moreover, there is no constraint applied on reading the Privacy policy and Terms and Conditions before a user can register
Solution	User validation must be applied to ensure the user has to open and scroll through to the end of both the Privacy Policy and the Terms and Conditions, before they can agree to them and register successfully.