VAPT Lootrix

Tested Games : Hilo and Roulette

1. Currency Parameter Tampering in Roulette and Hilo Endpoint Leading to Business Logic Risk

The currency parameter in the GET request to /roulette/ and /hilo/ is not properly validated or restricted on the server side. An attacker can supply arbitrary or unsupported currency values, potentially affecting financial logic, game behavior, or leading to inconsistent transactional outcomes.

Threats:

- Allows unauthorized currency injection.
- Could be used to exploit financial logic (if the backend uses this for payout calculation).
- Can lead to business logic flaws.
- Potential for bypassing regional restrictions or payment rules.

POC:

CVSS : 6**.5 (Medium)**

Remedies:

- ✓ Implement strict server-side validation for the currency parameter against an allow list of supported currencies.
- ✓ Reject any request containing unsupported currency values with appropriate error responses.
- ✓ Sanitize and normalize all incoming query parameters.

2. AWS Lambda Function URL Exposed Without WAF Protection

The publicly accessible AWS Lambda Function URL lacks any attached Web Application Firewall (WAF) or security gateway in front of it. Since Lambda Function URLs are direct HTTPS endpoints, without a WAF or API Gateway in front of them, they are exposed to the internet, increasing the risk of exploitation through malicious requests, automated attacks, or denial-of-service attempts.

URL : fwiknm2h5fpjwc32oguaevkggi0tibgf.lambda-url.ap-south-1.on.aws

Threats:

- ➢ Publicly exposed endpoint susceptible to abuse.
- ➢ Potential for Denial of Service (DoS) attacks by flooding requests.
- ➢ Increased injection attacks if input validation is weak.
- ➢ Loss of control over legitimate vs. malicious traffic.
- ➢ Attackers can bypass other application-level security controls.

CVSS: **6.5 (Medium)**

Steps To Reproduce:

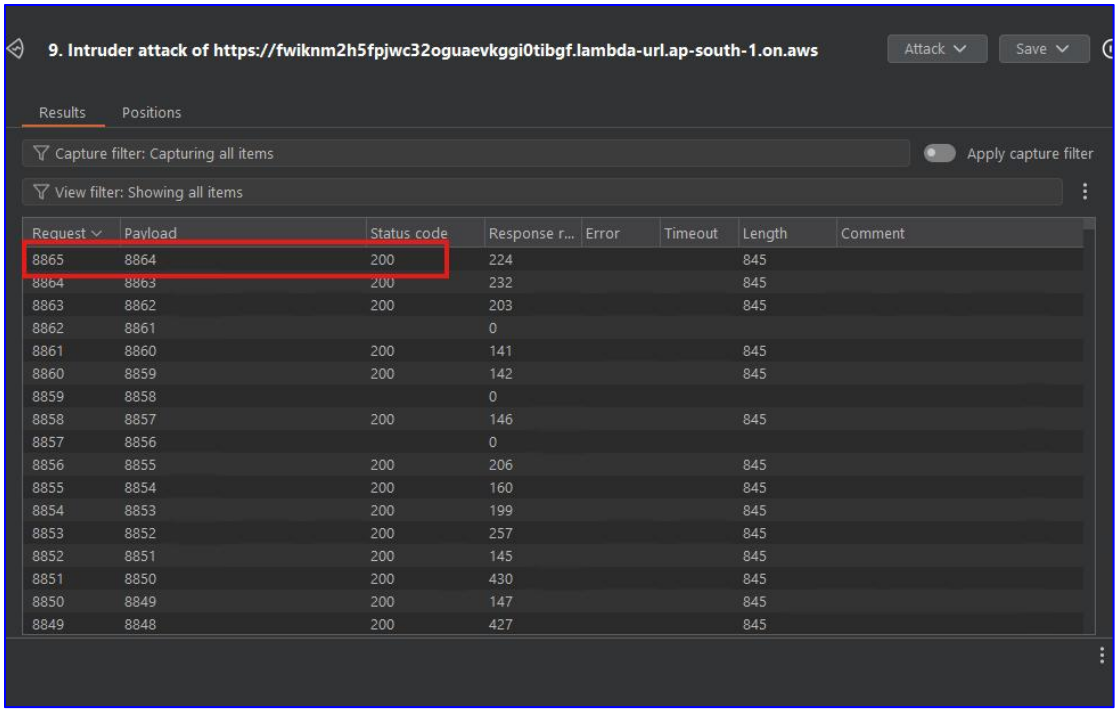
1. Identify the publicly accessible Lambda URL:

   fwiknm2h5fpjwc32oguaevkggi0tibgf.lambda-url.ap-south-1.on.aws

2. Use a tool like Burp Suite, cURL, or Postman to send direct requests to the endpoint.

3. Observe that no rate limiting, WAF-based filtering, or blocking mechanisms are in place.

4. Confirm endpoint responds to these requests without restrictions or security checks.



Remedies:

- ✓ Move the Lambda Function URL behind an API Gateway HTTP API.
- ✓ Set up an API Gateway HTTP API to serve as the new managed front-end for your Lambda function.
- ✓ Integrate your existing Lambda function behind the API Gateway HTTP API.
- ✓ Attach an AWS WAF Web ACL to the API Gateway, applying:
  - AWSManagedRulesCommonRuleSet
  - AWSManagedRulesSQLiRuleSet
  - AWSManagedRulesAnonymousIpList
- ✓ Custom rate-limiting and IP reputation rules.
- ✓ Redirect all client traffic to the API Gateway URL instead of the public Lambda Function URL to isolate backend infrastructure.
- ✓ Disable or restrict the public Lambda Function URL.
- ✓ Disable it entirely if not required, or Set its AuthType to AWS_IAM if access is needed internally within AWS services.
- ✓ **Alternative option**: Route traffic via CloudFront distribution with the Lambda URL as origin, and attach AWS WAF to the CloudFront distribution if API Gateway isn't suitable.
- ✓ Apply Geo-blocking rules via AWS WAF if the game/API is region-specific, to reduce global exposure.
- ✓ Implement rate-limiting and IP-based blocking using AWS WAF rules to prevent abuse, denial-of-service attacks, and automated bot traffic.

3. Open Redirect via Unvalidated return_url Parameter in Game Launch Request

The application allows users to provide a return_url parameter during the game initialization request. This URL is later used for redirection when the user exits the game. However, the application fails to properly validate this parameter, allowing attackers to redirect users to arbitrary external domains after gameplay.

This can be exploited for phishing, malware delivery, or social engineering attacks by luring users into playing a game session and then redirecting them to a malicious website upon exit.

Threats:

- ✓ Could redirect legitimate users to phishing/malware sites.
- ✓ Facilitates social engineering attacks.
- ✓ Can damage application reputation.
- ✓ Bypass security controls by redirecting to attacker-controlled domains.

CVSS : **6.8 (Medium)**

Steps to Reproduce:

1. Start a game by sending a GET request to:

https://game.lootrix.live/hilo/?name=Liam&user=<user-id>&token=<token>&operator=demo&currency=USD&return_url=https://google.com/exit/2025-04-30/11:44%20am

2. Play the game as normal.
3. Upon clicking the "Exit" button, the user is redirected to the domain specified in return_url.
4. Replace https://google.com with https://attacker.com and observe the redirection after exiting.

Remedies:

- ✓ Validate the return_url parameter against an allowlist of trusted domains or only permit redirects to specific, application-owned URLs.
- ✓ Implement strict URL parsing and sanitization to ensure untrusted external URLs cannot be passed.
- ✓ If redirects must happen, use relative URLs or assign pre-approved destination keys on the server-side (e.g., return=dashboard and server resolves it internally).
- ✓ Log and monitor redirect events to detect abnormal patterns or frequent redirects to external domains.
- ✓ Implement Strict CSP (Content Security Policy) headers to restrict unexpected redirects or navigations where possible.

4. Custom 404 Error Page Redirects to External Environment (Potential Misconfiguration)

The application's custom 404 error page is misconfigured to redirect users to a different environment (https://lootrix.utwebapps.com/). This unexpected redirect can expose users to unintended environments or potentially lead to information leakage, unauthorized access, or phishing risks. Users encountering the 404 error may unintentionally land on a different version of the site or development environment, which could lead to security issues or confusion.

Threats:

- ➢ Users may be redirected to an unintended or insecure environment.
- ➢ Potential information leakage about the underlying infrastructure or staging environments.
- ➢ Exposure of development or testing environment details to the public.
- ➢ Unauthorized access to environments not meant for production users.

CVSS: **3.1 (Low)**

Steps to Reproduce:

1. Navigate to a non-existent page or resource on the website (https://game.lootrix.live/error.html).

2. Observe that from error.html the user is redirected to another environment)https://lootrix.utwebapps.com).

3. Verify if the redirected environment is a development or staging environment by checking the content or headers.

4. Confirm that the redirect may reveal infrastructure details or lead users to unintended areas.

Remedies:

- ✓ Ensure the custom 404 page does not redirect to external domains, especially not to different environments or staging servers.
- ✓ Review the 404 error page configuration and ensure it serves only user-friendly, static error messages.
- ✓ Redirect to a safe, internal page like the homepage or a generic error page if redirection is necessary.

5. Exposure of Management & Game Server/Panels via Subdomain Enumeration

Multiple management, game build upload, API, game server panels and developenet/stagining environments or other projects were found exposed via publicly enumerable subdomains under the **lootrix.live** domain and its subdomains. These subdomains reveal internal host IPs embedded in DNS records (e.g. gameservernew.10.0.12.254.lootrix.live) and management interfaces such as admin panels, build servers, and client interfaces, which should not be publicly accessible or resolvable externally. **This will leads to bypassing of the current implemented WAF.**

This significantly increases the attack surface for enumeration, unauthorized access attempts, and other pre-exploitation activities.

Threats:

- Information Disclosure: Internal IP addresses and environment structure revealed via DNS.
- Attack Surface Expansion: Exposure of admin, build, and client management panels for potential brute-force or default credential attacks.
- Facilitates Pre-Attack Reconnaissance: Enables adversaries to efficiently enumerate critical services and game servers.
- Insecure Access Points: Potentially unprotected or misconfigured management interfaces.
- Risk of DDOS Attacks.

Enumerated Subdomains:

gameserver-test.lootrix.live
gameservernew.10.0.12.254.lootrix.live
gameservernew.10.0.30.46.lootrix.live
gameservernew.10.0.1.218.lootrix.live
gameservernew.10.0.14.252.lootrix.live
gameservernew.172.31.40.100.lootrix.live

aviator.10.0.23.119.lootrix.live
aviator.10.0.22.92.lootrix.live
gameservernew.10.0.16.116.lootrix.live
gameservernew.10.0.0.165.lootrix.live
gameservernew.10.0.20.225.lootrix.live
gameservernew.10.0.15.239.lootrix.live
gameservernew.172.31.38.25.lootrix.live
gameservernew.172.31.37.52.lootrix.live
gameservernew.10.0.1.175.lootrix.live
gameservernew.10.0.26.146.lootrix.live
gameservernew.10.0.14.4.lootrix.live
gameservernew.10.0.15.225.lootrix.live
gameservernew.172.31.7.142.lootrix.live
ssl.lootrix.live
gameservernew.172.31.36.140.lootrix.live
gameservernew.172.31.34.177.lootrix.live
gameservernew.10.0.1.36.lootrix.live
na.10.0.30.123.lootrix.live
status.lootrix.live
gameservernew.10.0.21.34.lootrix.live
gameservernew.10.0.27.248.lootrix.live
gameserver.172.31.4.195.lootrix.live
domaintest.lootrix.live
client.lootrix.live
gameservernew.10.0.10.18.lootrix.live
gameservernew.10.0.12.50.lootrix.live
gameservernew.10.0.9.242.lootrix.live
gameservernew.10.0.29.255.lootrix.live
gameservernew.10.0.24.51.lootrix.live
gameservernew.10.0.24.125.lootrix.live
gameservernew.172.31.4.195.lootrix.live
na.10.0.31.185.lootrix.live
lootrixlivenjserver.lootrix.live
gameservernew.10.0.22.115.lootrix.live
gameservernew.10.0.22.189.lootrix.live
gameservernew.172.31.12.253.lootrix.live
www.lootrix.live
admin.lootrix.live
gameservernew.10.0.12.234.lootrix.live
gameservernew.10.0.3.125.lootrix.live
gameservernew.10.0.9.86.lootrix.live
gameservernew.10.0.8.36.lootrix.live
gameservernew.10.0.16.170.lootrix.live
gameservernew.172.31.33.195.lootrix.live
build.lootrix.live
gameservernew.10.0.25.3.lootrix.live
gameserver.lootrix.live
aviator.10.0.12.174.lootrix.live
builds.lootrix.live
lootrixtest.lootrix.live
gameservernew.10.0.25.92.lootrix.live
shivtest.lootrix.live
gameservernew.10.0.14.45.lootrix.live
gameservertest.lootrix.live
aviator.10.0.26.174.lootrix.live
gameservernew.10.0.2.198.lootrix.live
gameservernew.10.0.29.64.lootrix.live

test.admin.lootrix.live
gameservernew.10.0.13.49.lootrix.live
gameservernew.10.0.9.252.lootrix.live
gameservernew.10.0.29.18.lootrix.live
gameservernew.10.0.1.171.lootrix.live
test.client.lootrix.live
gameservernew.10.0.4.31.lootrix.live
gameservernew.10.0.4.212.lootrix.live
gameservernew.10.0.18.3.lootrix.live
gameservernew.10.0.5.27.lootrix.live
gameservernew.10.0.8.56.lootrix.live
gameservernew.10.0.22.137.lootrix.live
gameservernew.10.0.10.107.lootrix.live
gameservernew.10.0.16.207.lootrix.live
gameservernew.172.31.12.217.lootrix.live
game.lootrix.live
gameservernew.10.0.7.56.lootrix.live
gameservernew.10.0.20.183.lootrix.live
gameservernew.10.0.25.68.lootrix.live
gameservernew.10.0.5.224.lootrix.live
gameservernew.10.0.6.204.lootrix.live
gameservernew.10.0.0.179.lootrix.live
gameservernew.10.0.3.24.lootrix.live
gameservernew.10.0.14.41.lootrix.live
gameservernew.172.31.45.223.lootrix.live
aviator.10.0.17.123.lootrix.live
gameservernew.172.31.5.145.lootrix.live
gameservernew.10.0.27.73.lootrix.live
gameservernew.10.0.11.160.lootrix.live
gameservernew.10.0.27.211.lootrix.live
gameservernew.10.0.5.196.lootrix.live
gameservernew.10.0.9.115.lootrix.live
gameservernew.10.0.4.18.lootrix.live
lootrixlivegames.lootrix.live
rumblebets.lootrix.live
gameservernew.10.0.6.76.lootrix.live
gameservernew.10.0.13.199.lootrix.live
gameservernew.10.0.24.66.lootrix.live
gameservernew.10.0.11.18.lootrix.live
livegames.lootrix.live
gameservernew.10.0.25.205.lootrix.live
lootrix.live
frontendnode.hrms.utwebapps.com

Remedies:

- ✓ Remove Public DNS Records for Internal Services: Eliminate subdomains mapping to internal IPs.
- ✓ Block DNS Zone Transfers.
- ✓ Implement Access Control on Critical Subdomains.
- ✓ Disable Wildcard DNS Entries.
- ✓ Clean Up Old & Unused Subdomains.
- ✓ Avoid Predictable Subdomain Naming Patterns.
- ✓ Restrict Access to Admin & Management Panels: Implement IP whitelisting, VPN access, or authentication gateways (e.g., Cloudflare Access, AWS WAF + Cognito).
- ✓ Implement a Subdomain Firewall/Reverse Proxy Policy
- ✓ Properly Segregate Dev & Test Environments, Ensure staging/dev environments are isolated and not exposed publicly.

- ✓ Regularly Review DNS & Reverse Proxy Configurations.
- ✓ Continuous Asset Discovery & Monitoring.