ORACLE

Siebel eScript Memory Leak Analyzer

AI-driven tool designed to analyze Siebel eScript code

July 2026
Document version 1.0

What Is the Siebel eScript Memory Leak Analyzer?

The Siebel eScript Memory Leak Analyzer uses AI-driven analysis to identify potential memory-leak risks in Siebel eScript before they affect test, staging, or production environments. The utility is implemented as an Oracle Cloud Infrastructure (OCI) Function and is exposed through OCI API Gateway.

How the eScript Memory Leak Analyzer Works

The analyzer accepts Siebel eScript source code, validates the request, and screens the input for prompt-injection risks. It then submits the code to a Generative AI Agent supported by a retrieval-augmented generation (RAG) knowledge base of Siebel eScript memory-leak patterns.

The analyzer returns identified issues and recommended improvements.

OCI API Gateway provides the public entry point for authentication, authorization, request processing, monitoring, and rate limiting. The backend OCI Function is not directly exposed to the internet.

Analyzer Workflow

Step	Stage	What Happens
1	Caller submits request	Client sends a JSON request containing the script field to the API Gateway endpoint.
2	Gateway checks access	JWT bearer-token and client-token controls are evaluated before the function is invoked.
3	Function validates input	The function checks the input type, blank content, ASCII content, and the 20,000-character limit.
4	Guardrails run	Prompt-injection protection is applied before the script is passed to the agent.
5	Agent analysis runs	The function creates an AI Agent session, sends the prompt and script, receives the analysis, then deletes the session.
6	Response is returned	The caller receives the analyzer output or an error payload with an HTTP status code.

Requesting Access

1. Email siebel_escript_analyser_support_grp@oracle.com to request that your email domain be added to the allowlist.
2. After Oracle processes the request, the requester receives the User Guide, Client ID, Client Secret, X-Client-Token, and self-registration instructions.
3. Open the self-registration URL and create a user account using an email address from the allowlisted domain.

Note: Access is managed through an IAM Identity Domain and OCI API Gateway. Oracle provides the self-registration URL only to invited users. Registration is restricted to email domains allowlisted in the self-registration profile. After Oracle confirms that your email domain is allowlisted, use the provided self-registration URL to create your account.

Self-Registration Steps

Step	Action
1	Open the self-registration URL provided by Oracle.
2	Register with an email address that belongs to a whitelisted domain.
3	Create a password that satisfies the password policy in this guide.
4	Complete the registration flow and retain the username and password. These credentials are used only to generate the OAuth2 bearer token.
5	If the UI displays [object Object], use the troubleshooting checklist before retrying.

Password Requirements

- Password length must be between 12 and 40 characters.
- Password must include at least one numeric character.
- Password must include at least one lowercase alphabetic character.
- Password must include at least one uppercase alphabetic character.
- Password cannot contain the username, first name, or last name.

Using the eScript Memory Leak Analyzer

Generate an OAuth2 Bearer Token

The analyzer API requires a bearer token issued by the IAM Identity Domain. The token-generation call uses the password grant type with the registered username and password.

Placeholder	Required?	Purpose	Value
<BASIC_AUTH_TOKEN>	Required	Use in the Authorization: Basic header for token generation.	Base64 value generated from the Client ID and Client Secret shared in the Oracle access email.
<SELF_REG_USERID>	Required	The username or email used during self-registration.	Provided by the registered user
<SELF_REG_PASSWORD>	Required	The password selected during self-registration.	Provided by the registered user

```
curl --location --request POST   'https://idcs-692ab7052c1a4b9bbf009494da64e6a2.identity.oraclecloud.com/oauth2/v1/token'   --
header 'Authorization: Basic <BASIC_AUTH_TOKEN>'   --header 'Content-Type: application/x-www-form-urlencoded'   --data-
urlencode 'grant_type=password'   --data-urlencode 'scope=urn:opc:idm:__myscopes__'   --data-urlencode
'username=<SELF_REG_USERID>'   --data-urlencode 'password=<SELF_REG_PASSWORD>'
```

Copy the access_token value from the token response and use it as <ACCESS_TOKEN> in the analyzer API request. The bearer token is expected to be valid for 60 minutes; programmatic callers should regenerate the token when it expires or becomes invalid.

Call the Analyzer API

Placeholder	Required?	Purpose	Value
<ACCESS_TOKEN>	Required	Bearer token returned by the token-generation call.	Generated per user session
<X_CLIENT_TOKEN>	Required	Client-token header checked by the API gateway or gateway policy.	Provided by Oracle after the sender requests domain whitelisting.
<SCRIPT_TEXT>	Required	Stringified Siebel eScript code to analyze.	Caller-provided script text

```
curl --location 'https://escript-siebelgenie.oraclecloud.com/v1/escriptassist' --header 'Content-Type: application/json' --header 'Authorization: Bearer <ACCESS_TOKEN>' --header 'X-Client-Token: <X_CLIENT_TOKEN>' --data '{"script":"<SCRIPT_TEXT>"}'
```

Sample Stringified JSON Request

```
{
  "script": "function SearchActivity(FieldName, FieldValue) {\nvar boBusObj = TheApplication().GetBusObject(\"Opportunity\");\nvar bcOppty = boBusObj.GetBusComp(\"Opportunity\");\nvar rowId = GetFieldValue(\"Id\");\nwith (bcOppty) {\n  SetViewMode(AllView);\n  ActivateField(\"Opportunity Id\");\n  SetSearchSpec(\"Id\", rowId);\n  ExecuteQuery(ForwardOnly);\n}\nboBusObj = null;\n}"
}
```

Request Schema

Field	Type	Required?	Validation / Notes
script	String	Required	Siebel eScript code as a JSON string. Must be non-empty, ASCII-only, and no more than 20,000 characters.

Successful Response

A successful call returns HTTP 200. The response body is text-oriented analyzer output, not a formal JSON object. It may include HTML-formatted bullets, pattern names, code suggestions, and status phrases such as Memory Leak Detected or No Memory Leak Detected.

Sl No	Sample Script	Sample Success Response

1	function SearchActivity (FieldName, FieldValue)
	{
	var bo = TheApplication().GetBusObject("Action");
	var bc = boBusObj.GetBusComp("Action"); /*this activates the Business Component for Activity , which is Action */
	var rowId = GetFieldValue("Id"); //accept the input Activity ID
	with (bc)
	{
	SetViewMode(AllView); //This gives access to data from all Organisations
	ActivateField("Account Name"); //Get the Account Name
	ActivateField("Account Id"); //Get the Account Id
	ActivateField("Activity UID"); //Get the Activity UID
	SetSearchSpec("Id", rowId); // Search based on input ID = Activity rowID
	ExecuteQuery(ForwardOnly); //We do not need to go back to row 1, better for performance
	}
	bc = null;
	}
		• Pattern 2: Business Object Memory Leak - - Memory Leak Detected -
		• Code Suggestion: var bo = TheApplication().GetBusObject("Action"); var bc = bo.GetBusComp("Action"); var rowId = GetFieldValue("Id"); with (bc) { SetViewMode(AllView); ActivateField("Account Name"); ActivateField("Account Id"); ActivateField("Activity UID"); SetSearchSpec("Id", rowId); ExecuteQuery(ForwardOnly); } bc = null; bo = null;

Sl No	Sample Script	Sample Success Response

2	//1. Get Access token from Token Relay Service of Fusion KM server//
	var Trasnport = TheApplication().GetService("EAI HTTP Transport");
	var Httpin = TheApplication().NewPropertySet();
	var Httpout = TheApplication().NewPropertySet();
	Httpin.SetProperty("HTTPRequestURLTemplate", "");
	Httpin.SetProperty("HTTPContentType", "application/json");
	Httpin.SetProperty("HDR.Authorization", "Basic XXXXXXX");
	Httpin.SetProperty("HTTPRequestMethod", "GET");
	Httpin.SetProperty("HTTPAccept", "*/*");
	Httpin.SetProperty("CharSetConversion", "UTF-8");
	Trasnport.InvokeMethod("SendReceive", Httpin, Httpout);
	// Can be removed or commented after debugging complete
	TheApplication().TraceOn("", "Allocation", "All");
	TheApplication().Trace("START - TEST v1.0");
	TheApplication().Trace(Httpout.GetValue());
	TheApplication().Trace("END - TEST");
	TheApplication().TraceOff();
	var access_token = Httpout.GetValue();
	//3. Add access token to Custom Filter Service BS input property set//
	var j;
	if(MethodName == "UpdateInput")
	{
	for (var i = 0; i < Inputs.GetChildCount(); i++)
	{
	j = Inputs.GetChild(i);
	if(j.GetType() == "header")
	{
	j.SetProperty("Authorization", "Bearer " + access_token);
	break;
	}
	}
	}
	if(MethodName == "UpdateOutput")
	{
	// write your script code here to update the response
	}
	Trasnport = null;
	return (CancelOperation);
		• Pattern 1: Business Service Memory Leak - - Memory Leak Detected -. Set business service object Trasnport = null; after usage, however it is already set to null at the end of the function.
		• Pattern 7: Siebel PropertySet Leaks 2 - - Memory Leak Detected -. Set PropertySet Httpin = null; and Httpout = null; at end of function or before return.
		• Code Suggestion: var Trasnport = TheApplication().GetService("EAI HTTP Transport"); var Httpin = TheApplication().NewPropertySet(); var Httpout = TheApplication().NewPropertySet(); // ... Trasnport.InvokeMethod("SendReceive", Httpin, Httpout); // ... Trasnport = null; Httpin = null; Httpout = null; return (CancelOperation);
		• Summary: Memory leaks detected in the given Siebel eScript code due to Business Service and PropertySet objects not being properly nullified. The suggested code fixes these issues by setting the objects to null at the end of the function.

HTTP Return Codes

Code	Category	Typical Cause	Caller Action
200	Success	Request was accepted and analyzed.	Read the analyzer body for memory-leak findings and code suggestions.
400	Validation error	Missing script field, wrong script type, blank or whitespace-only script, Unicode input, or prompt-injection/malicious input.	Correct the request body and retry.
401	Authentication failure	Bearer token is missing, expired, invalid, or rejected by API Gateway.	Generate a fresh token and retry.
403	Permission denied	Required client token is missing/invalid, user is not allowed, or domain/user access is not provisioned.	Confirm X-Client-Token, registration, and whitelist/domain access.
404	Not found	Endpoint path or route is incorrect, or the deployment is unavailable.	Check the API URL and route version.
409	Conflict	Request conflicts with current server state or duplicate operation handling.	Retry only after confirming the operation state.
413	Payload too large	Script exceeds the 20,000-character limit.	Submit one function at a time or reduce the payload.

Code	Category	Typical Cause	Caller Action
429	Throttled	API Gateway rate limit or usage limit has been exceeded.	Stop batch processing, wait for the allowed window, then retry later.
424 / 502 / 503	Dependency failure	Gateway, function, guardrail, AI agent, or downstream dependency cannot complete the request.	Retry with backoff if the request is otherwise valid; escalate if the issue persists.
500	Server error	Unexpected server-side failure.	Retry later. If the issue persists, email siebel_escript_analyser_support_grp@oracle.com with the request time, endpoint, user, and available logs or correlation details.

Rate Limits, Usage Limits, and Retry Guidance

When the API Gateway rate limiter or usage limit is exceeded, the caller receives HTTP 429 with a response such as {"code":429,"message":"Too Many Requests"}. This is the user-visible signal that the current rate or quota has been exceeded.

Rate Limit Policy

To safeguard the API hosted on Oracle Cloud Infrastructure (OCI), the following rate limiting and quota policies are currently enforced.

API requests are rate-limited on a per-client IP basis:

- Maximum requests per second: 1 request per second per client IP
- Usage plan limit: 5 requests per second

The following quota limits apply. Requests that exceed these limits are rejected.

- 20 requests per minute
- 100 requests per day

Troubleshooting HTTP Return Codes

No formal automatic server-side retry contract is documented for the public API. Callers should implement client-side retry behavior conservatively:

- Do not retry HTTP 400 validation errors until you correct the request payload.
- For HTTP 401 responses caused by an invalid or expired token, generate a new bearer token and retry the request once.
- Do not retry HTTP 403 permission or client-token errors until you correct the access configuration, domain allowlisting, or X-Client-Token.
- For HTTP 429 Too Many Requests responses, stop batch processing, wait for the applicable rate-limit window, and retry later. Use exponential backoff when supported by the calling utility.
- For HTTP 424, 500, 502, or 503 responses, retry a limited number of times using exponential backoff with jitter. If the issue persists, escalate the issue with the request time, user, endpoint, and any available correlation IDs or log details.
- Bearer tokens are expected to remain valid for 60 minutes. Programmatic clients should renew tokens automatically, where supported.

Troubleshooting Scenarios

Scenario	User-visible Symptom	Next Steps
Password policy	Registration or password reset fails.	Use a password that meets the Password Requirements in this guide.
Self-registration shows [object Object]	The self-registration UI displays [object Object].	Ensure the email address belongs to a whitelisted domain. Ensure the password meets policy. Ensure the same email address has not already been registered.
Expired password	{"error":"invalid_grant","error_description":"Your password is expired.","ecid":"704b3761c5424c08934b370324e46594"}	Open https://idcs-692ab7052c1a4b9bbf009494da64e6a2.identity.oraclecloud.com/ui/v1/myconsole and follow the flow to change the password.
Unauthorized token	{"code":401,"message":"Unauthorized"}	Generate a fresh bearer token and retry with Authorization: Bearer <ACCESS_TOKEN>.
Missing X-Client-Token	{"code":403,"message":"The token is not found at the configured location"}	Send the API request with header X-Client-Token: <X_CLIENT_TOKEN>.
Too Many Requests	{"code":429,"message":"Too Many Requests"}	A rate limit or usage limit was exceeded. Stop processing and retry after the allowed window.
Empty script	Input is empty. Please provide a code snippet.	Send a non-empty script value. Whitespace-only content is also rejected.
Missing or wrong script field	Input is not the correct type. Please provide a valid stringified eScript...	Send JSON with script as a string field, for example {"script":"function ..."}.
Unicode characters	Input contains Unicode characters. Please remove them.	Remove non-ASCII characters from the submitted script.
Oversized script	Script exceeds 20,000-characters limit. Please submit one function at a time.	Reduce the request size and submit one function or focused snippet at a time.
Prompt injection attempt	Unauthorized access or malicious input found.	Remove instructions or content that appears to manipulate the analyzer or bypass the intended task.
Client authentication failed	Token generation fails before the analyzer API call.	The confidential client secret may have rotated. Contact Oracle to receive the current Basic authorization value through the approved channel.

ORACLE

Frequently Asked Questions (FAQ)

Q: Are any request fields optional?
A: No. The public request contract does not document optional fields. The request body must include the required script string.

Q: Can I submit multiple functions in one request?
A: Submit one function or focused code snippet per request. The service enforces a 20,000-character limit.

Q: Does the API return JSON?
A: Successful analyzer output is text-based and can include HTML tags. Gateway error responses can be JSON. Always evaluate the HTTP status code before processing the response body.

Q: How do I know when a limit is exceeded?
A: HTTP 413 indicates that the submitted script exceeds the supported size. HTTP 429 indicates that a rate or usage limit has been exceeded. Batch utilities should log the event and stop further processing.

Q: Are prompt-injection checks performed?
A: Yes. The function invokes PromptInjectionProtectionResult before passing code to the AI Agent.

Q: Does Oracle Siebel CRM provide a utility to extract repository scripts and invoke the service programmatically?
A: Yes. Oracle provides a utility for this purpose. To request access, open a Technical Support Service Request. Oracle shares the utility and its corresponding User Guide through the Service Request.

Q: How is customer data protected?
A: The utility does not store customer data. It is designed to operate without data persistence and in accordance with applicable Oracle security and data-handling policies.