Oracle® Real User Experience Insight
Administration Guide

13.5
F37655-02
February 2023

ORACLE

Oracle Real User Experience Insight Administration Guide, 13.5

F37655-02

Contents

C Connecting a Collector to a GRE Tunnel

Preface

Oracle Real User Experience Insight (RUEI) provides you with powerful analysis of your network and business infrastructure. RUEI helps you to monitor real-user experience, set Key Performance Indicators (KPIs) and Service Level Agreements (SLAs), and send alerts when the thresholds are reached.

Audience

This document is intended for the following audience:

- System administrators responsible for the installation of RUEI. This assumes a sound understanding of the Linux operating system.
- The person within your organization designated as RUEI Super Administrator (that is, the `admin` user). They are responsible for post-installation configuration, and system maintenance.

Some familiarity with network and web technology is preferred. In particular, you should have a sound understanding of network topology, and a good operational knowledge of your organization's network and application environment.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Real User Experience Insight (RUEI) documentation set:

- *Oracle Real User Experience Insight Release Notes*
- *Oracle Real User Experience Insight User's Guide*
- *Oracle Real User Experience Insight Installation Guide*

For the latest version of this document and other RUEI documentation, see https://docs.oracle.com/cd/E73210_01/nav/associated_products.htm.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Controlling Reporting

This chapter provides information about the settings that help you to optimize the reports created by monitored traffic. These settings include increasing the amount of information available within the Failed Data Browser groups, increasing the default user flow limits, and obtaining user event information.

Obtaining User Event Information

The RUEI database contains information about user events (such as when a user opens a report, consults a KPI alert log, or logs on and off). This information can be used for various purposes, such as determining how often a particular report is opened or downloaded by users, or which is the most frequently accessed Data Browser group. By collecting and analyzing this information, you can optimize your RUEI installation to more accurately meet user requirements.

The recording of user events is controlled by the `user_events_enabled` setting, in the `C_config` table. When set to 1 (the default), user events are recorded; when set to 0, user events are not recorded.

By default, information about user events is stored in the database for a maximum of 31 days. This is controlled by the `db_max_user_events` entry in the `C_config` table. To modify either of these settings, do the following:

1. Get access as a `RUEI_USER` user
2. Run the following command to modify the user event retention setting:

   ```
   execsql config_set_value processor db_max_user_events days
   ```

where, *`days`* specifies the maximum number of days for which user event information should be stored. This setting has an impact on database usage.

User Event Table Structure

The following table lists the `user` event information.

Table 1-1 C_EVENTS Table

Column	Type	Description
`ID`	NUMBER	Unique ID used to identify the user event.
`STAMP`	TIMESTAMP	Time (in UTC format) when the event was performed by user.
`USERNAME`	VARCHAR2 (255 BYTE)	Login name of user.
`CODE`	NUMBER	The event code.
`EVENT`	VARCHAR2 (4000 BYTE)	Description of the event.

Event Codes and Descriptions

The following tables lists the possible `CODE` events and their associated descriptions.

Table 1-2 C_LANG_CATALOG_DATA Table

Code	Description
0	User logon.
1	User logout.
2	Load/reload Dashboard tab.
3	Added dashboard (%1%s).
4	Updated dashboard (%1$s).
5	Removed dashboard (%1$s).
6	Load/reload Report tab.
7	View report (%1$s).
8	Load/reload preview report (%1$s).
9	Save report (%1$s).
10	Save report as new (%1$s).
11	Download report as PDF (%1$s).
12	Download report as CSV (%1$s).
13	Download report as TSV (%1$s).
14	Download report as XLS (%1$s).
15	Download report as XML (%1$s).
16	Add report to Favorites (%1$s).
17	Remove report from Favorites (%1$s).
18	Toggle report %1$s mailing (%2$s).
19	Remove report from %1$s mailing (%2$s).
20	Send %1$s mailing now.
21	Load/reload Browse tab.
22	Select graph (%1$s).
23	Select graph category (%1$s).
24	Select group (%1$s).
25	Load/reload diagnostics.
26	Browse report (%1$s).
27	Load/reload KPI overview tab (%1$s).
28	Load/reload KPI overall alert log.
29	Show KPI specific alert log (%1$s).
30	Load/reload KPI correlation (%1$s).
31	User %1$s has been added (%2$s, disabled: %3$d, locked: %4$d, admin: %5$d, sec officer: %6$d).
32	User %1$s has been removed.
33	Application %1$s has been added.

Table 1-2 (Cont.) C_LANG_CATALOG_DATA Table

Code	Description
34	Application %1$s has been removed.
35	Service %1$s has been added.
36	Service %1$s has been removed.
37	Suite %1$s has been added.
38	Suite %1$s has been removed.
39	Collector profile %1$s has been added.
40	Collector profile %1$s has been removed.
41	Collector %1$s has been registered in profile %2$s.
42	Collector %1$s from profile %2$s has been unregistered.
43	Collector %1$s in profile %2$s has been restarted.
44	Collector %1$s in profile %2$s has been disabled.
45	Collector %1$s has been moved to profile %2$s.
46	Traffic filter in profile %1$s has been changed to %2$s.
47	VLAN filter in profile %1$s has been changed to %2$s.
48	Port numbers (%1$s) in profile %2$s has been added.
49	Port numbers (%1$s) in profile %2$s has been removed.
50	The IP filter (%1$s) has been added in profile %2$s.
51	The IP filter (%1$s) has been removed from profile %2$s.
52	User account %1$s has been enabled.
53	User account %1$s has been disabled.
54	User account %1$s has been locked.
55	User account %1$s has been unlocked.
56	Maximum login attempt reached for user account %1$s.
57	The password for user %1$s has been expired.
58	The initial password for user %1$s has expired.
59	The minimum password length has been changed to %1$s.
60	The maximum password duration has been changed to %1$s days.
61	Remove report (%1$s).
62	URL prefix %1$s with action: %2$s has been added.
63	URL prefix %1$s with action: %2$s has been removed.
64	URL prefix %1$s with action: %2$s has been updated.
65	Default replay action has been changed to %1$s.
66	Replay IP range action has been changed to %1$s.
67	Replay IP range %1$s has been added.
68	Replay IP range %1$s has been removed.
69	Replay all IP ranges have been removed.
70	Replay IP range %1$s has been changed.
71	Replay IP ranges uploaded.

Table 1-2 (Cont.) C_LANG_CATALOG_DATA Table

Code	Description
72	%1$s action with source value: %2$s and action: %3$s has been added.
73	%1$s action with source value: %2$s and action: %3$s has been removed.
74	%1$s action source value: %2$s and action: %3$s has been updated.
75	Default action for %1$s has changed to %2$s.
76	User account %1$s has been renamed to %2$s.
78	User account %1$s password has been changed.
79	User account %1$s has been set as administrator.
80	User account %1$s has been unset as administrator.
81	User account %1$s has been set as security officer.
82	User account %1$s has been unset as security officer.
83	The initial password duration has been changed to %1$s days.
84	The number of allowed login attempts has been changed to %1$s.
85	The SSL key (%1$s) valid from %2$s to %3$s in profile %4$s has been added.
86	The SSL key (%1$s) valid from %2$s to %3$s in profile %4$s has been removed.
87	SSL certifcate masking in profile %1$s has been changed to %2$s.
88	KPI %1$s (%2$s) has been added.
89	KPI %1$s (%2$s) has been removed.
90	KPI %1$s (%2$s) has been updated.
91	KPI category %1$s has been removed.
92	KPI category %1$s has been renamed to %2$s.
93	Naming scheme of %1$s has been updated.
94	Loading satisfaction of %1$s has been updated.
95	System account has been set to not expire.
96	System account has been set to expire.

Increasing the Size of the Failed Groups

The Failed URLs, Failed services, and Failed pages groups do not use the maximum group size setting. Instead, their size is controlled through the `event_max_fail` setting. This specifies the maximum number of rows that can added to the group's main database table during a 1-minute period. By default, this is 1000 rows. For the Slow URLs group, the `event_max_slow` setting is used, and specifies the number of the slowest URLs that are recorded within each 1-minute period. By default, this is 1000 rows.

If you change the `event_max_fail` or the `event_max_slow` setting, you should also review the `daily_max_fail` setting. This specifies the maximum number of rows that the groups' tables can contain. This is derived from the formula 1440 * `event_max_fail`. By default, this is 1.4 million rows.

To modify the size settings, run the following commands

```
execsql config_set_value processor event_max_fail 10000
execsql config_set_value processor daily_max_fail 4320000
```

The `event_max_fail` setting is limited to a maximum of 10,000 rows.

Pre-requisites:

- Confirm that more than 1000 error pages are *actually* reported for a 1-minute period within the All sessions group.
- Ensure that replay viewer functionality is enabled. To verify, select **Configuration** >**Security** >**Replay logging policy**>**Default replay action** setting, and then select **Complete logging** option.

> **Note:**
>
> Before changing the default of 1000 error pages, you should consider the following:
>
> - Consider whether you need to increase this limit. Typically, if a high number of error pages are reported within a 1-minute period, it is unlikely that they refer to different problems. Hence, having a large number of recordings for the same page errors might not help with root-cause analysis.
> - Increasing the limit imposes a considerable I/O overhead on both the Reporter and Collector systems. Therefore, you should carefully consider the limits of these systems before modifying the default limit.
> - Each group within the Data Browser has a maximum size. This is 1.5 times its "condense limit" (as specified by the `cube_max_size` option in the `C_CONFIG` table). The effect of trying to merge more than 5000 error pages within a 5-minute period can be that the system stops merging data at some point during the day. The more error pages that are encountered, the sooner the Data Browser group will become full. You can diagnose this in the error log file (*`RUEI_DATA`*`/processor/log/error.log`) by searching for errors containing the string "wg_failpg_dy_*" starting with the string "no merge:".
> - The `event_max_fail` settings is used by both the Failed pages and Failed URLs, and Failed services groups.

Increasing the Default Limits for User Flows

The default maximum number of steps that can be defined within a user flow is 15. This can be modified by using the `txn_max_steps` setting. The default maximum number of user flows that can be defined is 200. This can be modified by using the `txn_max_trans` setting. To change either setting, do the following:

1. Login to the Reporter system as the *`RUEI_USER`* user.
2. Run the following commands:

   ```
   execsql config_set_value processor txn_max_steps steps
   execsql config_set_value processor txn_max_trans flows
   ```

   where:

   - *`steps`* specifies the new maximum number of steps allowed with user flows.

- *flows* specifies the new maximum number of user flows that can be defined.

> **Note:**
>
> Increasing the default maximum values might result in a performance overhead. In addition, if the maximum number of steps within user flows is significantly increased, the graphical reporting of user flows (such as the Flow status and Flow transitions) may become difficult to read.

Obtaining Client IP Addresses within Desktop Virtualization Environments

By default, the client IP address is obtained from the IP header packet sent from the client. The IP packet contains, among other things, the numerical source and destination address of the packet. If RUEI has been placed after a NAT device (such as a load balancer), you can configure RUEI to look in a specified header (set by the NAT device) rather than the IP packet. For more information on this procedure, see **Monitoring NATed Traffic** in the *Oracle Real User Experience Insight User's Guide*. However, if monitored clients are using a desktop virtualization environment (such as a Citrix server), the IP address of the server is returned as the client IP address.

The following important points need to be considered:

- In desktop virtualization environments, you connect to the Internet using a browser running on the Desktop Virtualization Server (citrix for example) rather than on the client machine. RUEI sees the IP from the Virtualization server and not the real originating client IP from the user. However, RUEI provides mapping of user-id to client-ip to provide some way of reporting on the real originating client IP. You can upload this mapping, but this has limited functionality.
- The map-ranges file contains the originating server IP ranges from which the user-id to client-ip mapping is done.
- The map-users file contains the user-id to real originating client-IP. For example: A set of Citrix Servers have IP addresses in the ranges 10.0.1.2 - 10.0.1.254 (10.0.1.0/24). Citrix Clients connecting to the Citrix Server have IP-addresses for example in the range 192.168.1.2 to 192.168.1.254 (192.168.1.0/24). Users on these Citrix clients are using a web-application monitored by RUEI. In order to configure RUEI to report on the real client-ip instead of the Citrix Server IP, the following configuration is used:

```
RANGE
10.0.1.0/24

USER_ID\tCLIENT_IP
JohnSmith\t192.168.1.10
FredWhite\t192.168.1.10
SteveBrown\t192.168.1.10
```

- When a session with a client-ip (the Citrix Server IP) within one of the ip-ranges in the RANGE file is found, RUEI will attempt to map the user-name from that session to a real-client ip (the citrix-client-pc of the user) by using the `USER_ID-CLIENT_IP` mapping file.

So, any functionality or reporting (for example, Client Network views in the data browser) in RUEI that depends on the client-ip will use the mapped client-ip. If no match is found in the `USER_ID\tCLIENT_IP` mapping file, the original client-ip retrieved from TCP/IP layer or from configured header will be used.

> **Note:**
>
> Any user having a client IP in the map-ranges file, but where the user id is not in the map-users file, is not mapped. Pages requested by that user are reported with IP "unknown".

To configure RUEI to report a preferred client IP address, do the following:

1. Create a file containing a list of the IP address range(s) that you want to be remapped. Each range must be specified using the format 10.1.1.0/24. It is recommended that you call the file `ip-map-ranges-file.tsv`, this file can also contain IPv6 based CIDR notation. For example:

   ```
   RANGE
   169.254.0.0/16
   172.16.0.0/12
   ```

2. Create a tab-separated file containing a list of the required user IDs and client IP addresses. It is recommended that you call the file `ip-map-users-file.tsv`, this file can also contain IPv6 based CIDR notation. For example:

   ```
   USER_ID\tCLIENT_IP
   JohnSmith\t10.10.10.50
   FredWhite\t10.10.10.51
   SteveBrown\t10.10.10.52
   ```

   In the above example `\t` indicates a tab character. Ensure that both files do not contain any leading or trailing characters, and no lines containing only whitespace or special characters (such as `/n` or `/r)`.

3. Logon to the RUEI Reporter system as the *`RUEI_USER`* user.
4. Import the two created files onto a suitable location on the RUEI Reporter system.
5. Run the `import-ip-map` script (located in the *`RUEI_DATA`*`/processor/bin` directory) by using the following command:

   ```
   import-ip-map -r ip-map-ranges-file -u ip-map-users-file
   ```

   Where, *`ip-map-ranges-file`* and *`ip-map-users-file`* are the two files to be created and imported.

   Any reporting changes made by this facility take effect within appropriately 5 minutes.

Restoring Default functionality

To restore default client IP address reporting, create two files containing only column headers, and repeat the previous procedure.

Controlling the Maximum Session Duration and Idle Time

By default, a visitor session is regarded as terminated if the visitor has been inactive for longer than 60 minutes. This is controlled through the `session_idle_time` setting. In addition, the default number of hours that user IDs and custom dimensions are remembered for a session is 12 hours. This is controlled through the `max_age_session` setting.

Lowering the `session_idle_time` setting will increase Reporter system performance in terms of CPU utilization. It has no impact on memory usage. However, a drawback of lowering this setting is that the identified visitors returning within the specified session idle time will be reported as anonymous.

You should consider lowering the `max_age_session` setting when the Reporter system does not have enough memory and starts to swap. When this setting is lowered, and the monitored traffic contains mostly long sessions, user IDs can be lost. This setting should not be set lower than the `session_idle_time` setting.

Use the following commands to obtain a setting's current value:

```
execsql config_get_value processor session_idle_time
execsql config_get_value processor max_age_session
```

Use the following commands to modify a setting's value:

```
execsql config_set_value processor session_idle_time idle_time
execsql config_set_value processor max_age_session max_age
```

Where:

- *`idle_time`* specifies the number of seconds of visitor inactivity after which the session is considered terminated.
- *`max_age`* specifies the maximum number of hours after which session information is cleared from memory.

Improving Processing Concurrency

By default, 3 threads are used on the Reporter system for traffic processing. It is controlled by the `lookup_threads` setting. Performance improvement can be obtained (through additional concurrency in processing) by increasing the value of this setting. An indication that this setting is too low is the following internal error appearing in the Event log:

```
Processing backlog larger than %d minutes, restarting logr (the backlog will be
skipped).
```

It means that the Reporter system cannot keep up with the processing of the arriving data.

Use the following command to obtain the setting's current value:

```
execsql config_get_value processor lookup_threads
```

Use the following command to modify the setting's value:

```
execsql config_set_value processor lookup_threads threads
```

Where, *threads* specifies the number of threads available for use by the Reporter system. This setting must not be higher than the number of cores available on the Reporter system.

2
Configuring Collector Systems

This chapter provides information about the settings to configure your Collector systems to perform domain-based segmentation, and increase the memory available to Collector processes.

Increasing Memory Availability to Collectors

By default, the Collector process (`panther`) is assigned 30% of available system memory within a single-server installation. Within a remote Collector installation, the Collector process is assigned 70% of available memory.

To set the memory available to the Collector process, run the following command:

```
execsql config_set_profile_value profile config MaxMemoryUsage replace setting
```

Where:

- *`profile`* specifies the name of the Collector profile that needs to be updated.
- *`setting`* is the percentage of system memory available to the Collector process. The percentage sign must *not* be specified with the setting. It is recommended that you specify a percentage not higher than 90%. If the Collector process has to share resources with other software running on the system, a maximum setting of 80% is more appropriate.

To obtain the required Collector profile name on Reporter GUI, select **Configuration** > **Security**, and then **Collector profiles**, or run the following command:

```
execsql config_get_profiles
```

Configuring Domain-Based Segmentation

To configure RUEI to filter (segment) monitored traffic based on domain names, do the following:

1. Select **Configuration** >**Security** >**Network filters**, and select the required Collector profile. Ensure that the **Packet capture** menu specifies the **Specified domains** option for each of the required Collector profile.
2. Create, modify, or delete the required rows in the `c_domain_segments` database table. The table has the following format:

   ```
   ID        Priority   Domain    Profile_ID   Traffic_segment
   1000      10         *.nl           2            1|1
   1100      8          *.be           2            1|2
   1150      3          *.oracle.*     2            1|1
   1200      1          *.com          2            3|4
   ```

   Where:

   - The `ID` column represents a unique identifier for each row in the table.

- The `Priority` column represents the order in which the filters are applied. The filters with the highest priority numbers are applied first, and those with the lowest are resolved last. In the example, monitored traffic relating to the domain `myshop.oracle.com` would be filtered as `*.oracle.* 1|1`, and not the `*.com 3|4` filters. Also, all domain traffic with the country code `nl` is monitored, while only the first half of the data stream should be monitored for domains with the country code `be`.
- The `Domain` column contains the actual filter value where * can be used as a wildcard.
- The `Profile_ID` column relates to the ID of the Collector profile for which the filters should apply. This ID can be found in `c_cprofiles`.
- The `Traffic_segment` column contains the segment which should be used for the specified filter. You can specify up to 128 parts. For example, 34|128 will take the 34th segment out of 128.

3. To view the currently defined network filters, logon to the Reporter system as the *`RUEI_USER`* user, and run the following command:

```
sqlplus /@RUEI_DB_TNSNAME
select id, prio, domain, profile_id, traffic_segment from c_domain_segments order by prio;
```

4. To insert a row into the table, run the following command:

```
insert into c_domain_segments (id, prio, domain, profile_id,
traffic_segment) values (c_domain_segments_seq.nextval, 1, '*.nl', 2, '1|2');
```

5. To delete a row from the table, run the following command:

```
delete from c_domain_segments where id=1;
```

6. To alter a filter's priority, run the following command:

```
update c_domain_segments set prio=100 where id=2;
```

Configuring the Forms Socket Mode Timeout

By default, the Forms socket mode setting is set to 10 minutes. To view it, run the following command:

```
execsql config_get_profile_value System forms FormsSocketTimeout
```

To alter it, run the following command:

```
execsql config_set_profile_value System forms FormsSocketTimeout replace 600
```

Configuring Collector Performance Settings

The collector can use multi-threading to perform more traffic analysis tasks. Along with other scaling settings, this feature can be used to let the collector take advantage of the increasing amounts of CPU cores and memory installed in modern hardware configurations. This results in a more efficient usage of available hardware resources and increased performance.

The collector contains a pipeline of different thread types. Different thread types perform different functions, and it is important to increase the thread count for the specific thread type that is experiencing high load. Load per thread type can be viewed in the collector status, performance section.

Configuring the RX_RING Buffer

The RX_RING buffer is used to send network traffic from the kernel to the collector process. Configuring this buffer to contain more frames increases collector resilience against small fluctuations in the amount of incoming traffic. Configuring this buffer to contain fewer frames reduces memory usage, and may be required to configure *Jumbo* frames.

There are two collector configuration parameters:

- *CaptureLength*: The CaptureLength setting specifies the maximum size of a single packet to be captured. This parameter can be found in the UI at the following location: **Configuration**-->**Security**-->**Jumbo frames**
- *CaptureBufferMaxMemoryUsage*: The CaptureBufferMaxMemoryUsage setting limits the total size of the buffer.

The actual memory used is based on `CaptureBufferMaxMemoryUsage`, but may be adjusted down automatically. Kernels older than UEK v4.14.35-2025.400.1 or mainline v5.0 do not support 4GB or more of reserved memory.

Examples

```
execsql config_set_profile_value profile config CaptureLength replace 65536
execsql config_set_profile_value profile config CaptureBufferMaxMemoryUsage
replace 4095mb
```

Replace the word *profile* with the correct profile name. Instructions on how to obtain the profile name can be found at the bottom of this page.

NumATMThreads

The 'atm' threads handle HTTP stream parsing. By default, there is 1 ATM thread.

> **Note:**
>
> If your monitored traffic contains servlet-mode oracle forms traffic, do not increase the number of ATM threads.

HTTPPrestartApts

The 'apt' threads handle HTTP content parsing, including content scan and xpath scanning. By default, there is one APT thread per ATM, and additional threads start automatically. There can be a maximum of 64 threads per ATM.

Examples

If the collector system has a minimum of 12 cores and at least 32GB of RAM, run the following command as the RUEI_USER on the reporter system:

```
execsql config_set_profile_value profile http HTTPPrestartApts replace 4
execsql config_set_profile_value profile config NumATMThreads replace 2
execsql config_set_profile_value profile config CaptureBufferMaxMemoryUsage replace 3gb
```

Where, *profile* is the name of the Collector profile that needs to be updated.

If the collector system has a minimum of 24 cores and at least 32GB of RAM, enter the following command as the RUEI_USER on the reporter system:

```
execsql config_set_profile_value profile http HTTPPrestartApts replace 4
execsql config_set_profile_value profile config NumATMThreads replace 4
execsql config_set_profile_value profile config CaptureBufferMaxMemoryUsage
replace 3gb
```

Where, *profile* is the name of the Collector profile that needs to be updated.

Obtaining the Profile Name

To view the profile name on RUEI console, go to **Configuration**> **Security**, and then select **Collector Profiles** or run the following command:

```
execsql config_get_profiles
```

3
Maintaining the System

This chapter provides information about the settings to perform various maintenance tasks, such as backing up a RUEI deployment, and improving Reporter GUI performance.
In general, use the following procedure:

1. Stop processing by entering the following command as the *RUEI_USER* user:

   ```
   project -stop
   ```

2. Perform the maintenance as described in the relevant section.
3. Restart processing by entering the following command as the *RUEI_USER* user:

   ```
   project -start
   ```

Disabling Modification to Administrators' Properties

By default, users with Administrator permissions can change the properties of other Administrators, as well as create and delete Administrator user accounts. If this is not consistent with your security requirements, you can disable this functionality by running the following command:

```
execsql config_set_value wi_core user_mgmt_admin_edit_admins 0
```

Increasing the Linux Socket Memory Allocation Limit

The underlying Linux socket interface used by the Collector for monitoring traffic has a memory allocation limit of 20KB. This limit can be exceeded when a large number of network filters (or VLAN definitions) are configured. If limit exceeds, the following error is reported in the Event log:

```
linux.c, 326,cap_dev_set_filter()]: setsockopt(): Cannot allocate memory
```

To increase this limit, do the following:

1. Login to the required Collector system as the `root` user.
2. Run the following command to increase the underlying limit:

   ```
   /sbin/sysctl -w net.core.optmem_max=65535
   ```

3. To make this setting persistent across reboots, add the following line to the `/etc/sysctl.conf` file:

   ```
   net.core.optmem_max=65535
   ```

Backing up a RUEI Deployment

RUEI does not provide dedicated database backup and recovery functionality. Instead, it relies on standard Oracle database functionality. This is described in the *Oracle Database*

Backup and Recovery User's Guide, available at http://docs.oracle.com/cd/B28359_01/backup.111/b28270/toc.htm.

Important

Regardless of the backup method you use, it is recommended that you first stop RUEI data processing. Unless you do so, the integrity of the backed up data cannot be guaranteed. To stop RUEI data processing, run the following command as the *RUEI_USER* user:

```
project -stop
```

This procedure may take several minutes, and any data being processed at the time of the stop command will be lost. However, traffic monitoring continues, and is written to log files that will be committed to the database once processing is resumed.

After the backup is complete, restart the processing by running the following command:

```
project -start
```

Backing up RUEI Configuration Data

In addition to the database, RUEI configuration data should also be backed up. The following procedure helps to extract configuration data from both the database as well as the file system, and writes it to the file system where it can be picked up for further backup to a suitable storage device.

1. Login to the Reporter system as the *RUEI_USER* user, and run the following command:

   ```
   project -save
   ```

   By default, this stores the backup data at *RUEI_DATA*/processor/backup. An alternate location can be specified using the `-file` directive. For example, to store to the location `/tmp/backup`, run the following command:

   ```
   project -save --file=/tmp/backup/backup.tar.gz
   ```

2. To restore an earlier backup, run the following command:

   ```
   project -restore /tmp/backup/backup.tar.gz
   ```

Backing up Session Diagnostic Data

One of the major strengths of RUEI is its ability to diagnose individual user sessions for slow performance or problem pages. This functionality relies on log files that are stored outside of the RUEI database. In order to allow access to Session Diagnostics functionality, this data also needs to be available during a restore. Backup the contents of the *RUEI_DATA*/processor/data directory.

Replay content is the data required to replay error pages or the full content of a session. Backup of this data depends on your requirements. That is, if there is a need to replay session content on a regular basis. Replay content can be easily backed up from the file system. The relevant directories are *$APPSENSOR_HOME*/*/REPLAY. The default location is *RUEI_DATA*/collector/wg/REPLAY. The entire directory (and all sub-directories) should be backed up.

The directories indicated above must be backed for *each* required Collector system. In a distributed environment, the backup may have to be performed on multiple systems.

Restoring a RUEI Deployment Backup

To restore a RUEI deployment from scratch, do the following:

1. Install the RUEI software. For more information, see *Oracle Real User Experience Insight Installation Guide*.
2. Restore the database content for the selected backup approach. For information, see Oracle Database Backup and Recovery User's Guide.
3. To restore the RUEI configuration information, run the following command:

   ```
   project -restore --all backup-file-location
   ```

   Where, *backup-file-location* specifies the location of the backed-up data.
4. Restore the RUEI Session Diagnostics information by restoring the contents of the *RUEI_DATA*/processor/data directory.
5. Stop the Collector by running the following command as the RUEI_DATA user:

   ```
   appsensor stop wg
   ```
6. For each required Collector system, restore the replay content to the location *$APPSENSOR_HOME*/*/REPLAY.
7. Start the Collector by running the following command as the *RUEI_DATA* user:

   ```
   appsensor start wg
   ```

Moving RUEI Datafiles to a New Location

There might be a requirement to move the database data files to a new location. For example, because the current mount point or directory is running out of space. The following procedure assumes that the database is running on the Reporter system, and the default installation paths are being used. For more information, see Oracle Real User Experience Insight Installation Guide.

Complete the following steps:

1. Log in to the Reporter system as the *RUEI_USER* user.
2. Stop the database and processing by running the following command:

   ```
   project -stop/etc/init.d/oracledb stop
   ```
3. Prepare the new mount running the following commands:

   ```
   mkdir -p /oradata/ux/
   chown oracle:oinstall -R /oradata
   ```
4. Copy the datafiles as the oracle user by running the following commands:

   ```
   cd /u01/app/oracle/oradata
   mv ux/* /oradata/ux
   rm -f ux
   ln -s /oradata/ux ux
   ```
5. Restart the database and processing by running the following commands:

```
# /etc/init.d/oracledb start
# su - RUEI_USER$
project -start
```

Managing Users

You can create and manage user accounts by using Reporter interface in RUEI.

Creating Users

To create a new user account, run the following commands:

```
set serveroutput on
exec dbms_output.put_line (uxs_users.create_user('name', 'full-name', 'mail-
address', 'authentication', 'access-level', [ADM|SEC|EM_ACCESS => 1]));
```

Where:

- *name* specifies the user name by which the user will be known within the RUEI installation.
- *full-name* specifies the user's full name.
- *mail-address* specifies the user's E-mail address. This is the address to which reports and E-mail alerts will be sent. Ensure that this is correct.
- *authentication* specifies whether the user is authenticated against a configured LDAP (`ldap`) or Oracle SSO (`osso`) server.
- *access-level* specifies the Business and IT access-level permissions to be assigned to the user. This must be `0` (Full), `1` (Analytical), `2` (Inquiry), `3` (Overview), or `4` (None).
- Optionally, additional privileges can be assigned to the user. These are `ADM` (Administrator), SEC (Security Officer), or `EM_ACCESS` (Oracle Enterprise Manager access).

For example:

```
exec dbms_output.put_line(uxs_users.create_user('Jan', 'Jan Janssen',
'jan.janssen@test.com', 'ldap', '0', ADM => 1, SEC => 1));
```

The command will report an error message with the return code -1 if addition of the user account failed; 1 if successful.

Updating Users

To update a user account, issue the following commands:

```
set serveroutput on
exec
dbms_output.put_line(uxs_users.update_user('current_name','new_name','new_full_na
me', 'new_mail-address', 'new_authentication' ,'new_access-level',  [ADM|
SEC|EM_ACCESS => 1]));

exec dbms_output.put_line (uxs_users.create_user('name', 'full-name', 'mail-
address', 'authentication', 'access-level', [ADM|SEC|EM_ACCESS => 1]));
```

Where:

- *current_name* specifies the user name of the existing user that you want to update.
- *new_name* specifies the modified user name by which the user will be known within the RUEI installation.
- *new_full-name* specifies the user's full name.
- *new_mail-address* specifies the user's E-mail address. This is the address to which reports and E-mail alerts will be sent. Ensure that this is correct.
- *new_authentication* specifies whether the user is authenticated against a configured LDAP (`ldap`) or Oracle SSO (`osso`) server.
- *new_access-level* specifies the Business and IT access-level permissions to be assigned to the user. This must be `0` (Full), `1` (Analytical), `2` (Inquiry), `3` (Overview), or `4` (None).
- Optionally, additional privileges can be assigned to the user. These are `ADM` (Administrator), SEC (Security Officer), or `EM_ACCESS` (Oracle Enterprise Manager access).

The command will report an error message with the return code -1 if update of the user account failed; 1 if successful.

Deleting Users

To delete a user, run the following command:

```
exec dbms_output.put_line(uxs_users.delete_user('name'));
```

Where, *name* specifies the user name by which the user is known within the RUEI installation.

Modifying Browser JS Library API Keys

If you define Browser JS Library settings as described in the I*dentifying and Reporting Web Pages* chapter of the *RUEI User's Guide*, an API key is automatically created. To modify an API key, do the following:

> **Note:**
>
> Before attempting to modify the API key, make sure that the associated application is enabled.

1. List all RUEI applications and the association application ID by running the following command:

   ```
   execsql get_matches
   ```

2. Note the application ID for the application you want to modify.
3. Set the API key by running the following command:

   ```
   execsql config_set_api_key application_ID API_key
   ```

   Where, *application_ID* is the application ID you noted in step 2 and *API_key* is the new value for the API key.

Uploading Named Clients from file

You can now upload named clients from file through a new command line tool upload-named-clients.

Usage:

```
upload-named-clients [options] -f <named_clients_file>
```

Options:

```
-r clear existing entries
```

The format of the `named_clients_file` should be:

```
<ip_range>\t<client_group>\t<client_name> per line
```

Increasing SSHD MaxStartups on collector systems

When syncing collector log files to the reporter system on a busy collector system or if the collector system experiences a lower SSH throughput, the reporter displays a warning about the failure to sync data from that specific collector, or unexpected connection termination.

To increase SSHD MaxStartups on collector systems, do the following:

1. On the affected collector system as the `root` user, edit the file `/etc/ssh/sshd_config`.
2. Uncomment the line that reads `#MaxStartups 10:30:100`(remove the hash tag), and change the `MaxStartups` value to `30`.
3. Save the changes and restart the `SSH` daemon by running `service sshd restart` command as the `root` user.

4
Managing the Database

This chapter provides information about the settings required to perform database maintenance and facilitate backups.

Suspending Processing When Performing Database Maintenance

When performing maintenance on the database, it is recommended that you manually stop RUEI processing for the time that the database is down. This ensures that the error messages are not displayed during maintenance. To manually stop RUEI processing, do the following:

1. Use SSH to login to the Reporter system as the *RUEI_USER* user.
2. To stop processing, run the following command:

   ```
   project -stop
   ```

3. Ensure that the following processes are no longer running: `qjobd`, `logr`, and `rsynclogdird`. If necessary, run the `kill` command to stop them.
4. After completion of database maintenance, restart processing by running the following command:

   ```
   project -start
   ```

Enabling Online Tablespace Backups

As of version 12.1.0.3, the `USERS` and `UXCONF` tablespaces within new installations are set to `force logging` mode. Previously, the default mode was `nologging`. The upgrade procedure does not change your database's current setting. However, changing the tablespace mode to `force logging` can considerably increase disk I/O.

By default, the database does not support online backups. To enable online backups, the database's `noarchivelog` mode must be changed, and a number of operations changed from `nologging` mode to `force logging` mode. To support online backups, do the following:

1. Log in to the database system as the `oracle` user:
2. Stop all processing by runnning the following commands:

   ```
   source /etc/ruei.conf
   su - $RUEI_USER
   project -stop
   killall logmsgd
   killall qjobd
   killall rsynclogdird
   ```

3. Ensure that the *$RUEI_DB_INST* setting specifies the RUEI database.
4. Change the database to `archivelog` mode by running the following commands:

```
. oraenv
sqlplus / as sysdba
shutdown immediate
startup mount
alter database archivelog;
alter database open;
```

5. Set the required operations to `force` logging mode by running the following command:

```
alter tablespace USERS force logging;
alter tablespace UXCONF force logging;
```

6. Configure and schedule the online backup.
7. Restart processing by running the following command:

```
project -start
```

For more information, see the *Oracle Backup and Recovery User's Guide*.

Improving KPI Calculation Performance

By default, RUEI uses up to 8 parallel connections to run the KPI value queries. This is controlled by the `db_kpi_value_threads` setting. Increasing the number of parallel connections can improve KPI calculation performance. However, the value should not be set to a value higher than the amount of cores available from the database server, and there is no advantage of setting the value higher than the number of configured KPIs. It has no functional impact other than potentially making data processing run faster.

Run the following command to obtain the setting's current value:

```
execsql config_get_value processor db_kpi_value_threads
```

Run the following command to modify the setting's value:

```
execsql config_set_value processor db_kpi_value_threads *nthreads*
```

Where, *`*nthreads*`* specifies the number of parallel connections to use for KPI value queries.

Managing Subpartitions in RUEI Tables

RUEI tables have subpartitions for their primary partitions and during installation, they are set to a default value of 2.

To read the current value, run the following command:

```
execsql config_get_value processor num_subpartitions_kpi_id
```

If you need to change the number of subpartitions, use the following commands:

> **Note:**
>
> Changing the number of subpartitions may require an additional license.

- KPI tables:

  ```
  execsql config_set_value processor num_subpartitions_kpi_id 10
  ```

- User flow tables:

  ```
  execsql config_set_value processor num_subpartitions_user_flow_id 10
  ```

- All other tables:

  ```
  execsql config_set_value processor num_subpartitions_match_id 10
  ```

To apply the new settings, run the following command:

```
modr all --repartition
```

The new value will not take effect until a new primary interval partition has been created. Depending on the type of table, a new interval partition may be created only once a day or even once a month.

Enterprise Manager Repository Maintenance

When the Enterprise Manager repository is under maintenance, you can configure RUEI to ensure that no KPI status change information is sent to the non-functioning Enterprise Manager instance.

When the Enterprise Manager repository maintenance starts, run the following command on the RUEI system:

```
execsql emdb_set_status $host $sid maintenance
```

When the Enterprise Manager repository maintenance is complete, run the following command on the RUEI system:

```
execsql emdb_set_status $host $sid up
```

Removing Unused Columns in Fact Tables

Unused columns in fact tables may consume disk space and fact tables with fewer columns may increase performance.

If there are too many columns, a database error `ORA-01792: maximum number of columns in a table or view is 1000` is triggered when a new suite is added.

If a table has too many unused columns, you may see error messages similar to following message in project logs:

```
Fact table <FACT TABLE> has too many unused columns(unused=mmm, total=nnn)
```

Run the `ruei-upgrade.sh drop_unused_columns <FACT_TABLE>` command on table names shown in above message. For example, `F_RT_RTPAGE`.

To remove unused columns in the Fact tables, do the following:

1. Log in to the reporter machine as root.
2. Stop RUEI.
3. Run the following command:

   ```
   ruei-upgrade.sh drop_unused_columns <FACT_TABLE>, replace <FACT TABLE>
   ```

Where, `<FACT TABLE>` is fact table name displayed in warning message.

4. Start RUEI.

> **Note:**
>
> If the affected fact table contains large amount of data, then removing the unused columns might take time to process.

5
Troubleshooting

This chapter provides information about the settings for helping Customer Support to resolve problems encountered when using RUEI.

Enabling Core Dumps for Collector Processes

By default, in the event of a Collector instance crashing, no core dump is generated. This is for security reasons because the Collector may be monitoring encrypted (SSL) traffic. However, some customer issues can only be resolved by Customer Support if a core dump is made available. To enable creation of core dumps, do the following:

1. As a `RUEI_DATA` user, login to the system on which the Collector instance is running and run the following command:

   ```
   ulimit -c unlimited
   ```

2. Edit the *`APPSENSOR_HOME`*`/wg/config/config.cfg` file, and modify the value of `CoreSize` setting to -1.

3. Restart the Collector by running the following command:

   ```
   appsensor restart wg
   ```

When core dumps are enabled, stack trace extracts are stored in the *`APPSENSOR_HOME`*`/core_dir` directory. RUEI automatically cleans up any core dumps in the *`APPSENSOR_HOME`* directory, every night at 2:30 AM. In addition, if core dumps are regularly generated, the file system may fill up. Therefore, it is recommended that the default configuration is restored as soon as the required core dumps are generated.

Manually Creating Helpdesk Reports

When you contact Customer Support, it is recommended that a Helpdesk report file is created and uploaded to the Service Request (SR). The report file contains extended system information that is useful to Customer Support when handling the reported issues. To generate the report file, select **System**>**Maintenance**> **Helpdesk report**.

In the Reporter user interface, the Helpdesk report can be created manually by doing the following:

1. Log in to the Reporter system as the *`RUEI_USER`* user.

2. Run the following commands:

   ```
   source /etc/ruei.conf
   project -save --all
   ```

3. Retrieve the generated `.tgz` file from the location as indicated by the command output.

4. Upload the file to the appropriate SR.

A

Third-Party Licenses

This appendix contains licensing information about certain third-party products included with this release of RUEI. Unless otherwise specifically noted, all licenses herein are provided for notice purposes only.
The sections in this appendix describe the following third-party licenses:

Apache Software License, Version 2.0

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. **Definitions**. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or

other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. **Grant of Patent License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. **Redistribution**. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and
- You must cause any modified files to carry prominent notices stating that You changed the files; and
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places:

within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions**. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks**. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty**. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability**. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability**. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included

on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

OpenSSL

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org).

Copyright © 1998-2011 The OpenSSL Project. All rights reserved. It is distributed under the license available at the following location:

http://www.openssl.org/source/license.html

PHP

Copyright © 1999-2013 The PHP Group. All rights reserved.

This product includes PHP software, freely available from http://php.net/software/. It is distributed under the license available at the following location:

http://creativecommons.org/licenses/by/3.0/legalcode

Java Runtime Environment

ORACLE AMERICA, INC. ("ORACLE"), FOR AND ON BEHALF OF ITSELF AND ITS SUBSIDIARIES AND AFFILIATES UNDER COMMON CONTROL, IS WILLING TO LICENSE THE SOFTWARE TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY SELECTING THE "ACCEPT LICENSE AGREEMENT" (OR THE EQUIVALENT) BUTTON AND/OR BY USING THE SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THE TERMS AND AGREE TO THEM. IF YOU ARE AGREEING TO THESE TERMS ON BEHALF OF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE LEGAL AUTHORITY TO BIND THE LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO BE BOUND BY THE TERMS, THEN SELECT THE "DECLINE LICENSE AGREEMENT" (OR THE EQUIVALENT) BUTTON AND YOU MUST NOT USE THE SOFTWARE ON THIS SITE OR ANY OTHER MEDIA ON WHICH THE SOFTWARE IS CONTAINED.

1. DEFINITIONS. "Software" means the software identified above in binary form that you selected for download, install or use (in the version You selected for download, install or use) from Oracle or its authorized licensees, any other machine readable

materials (including, but not limited to, libraries, source files, header files, and data files), any updates or error corrections provided by Oracle, and any user manuals, programming guides and other documentation provided to you by Oracle under this Agreement. "General Purpose Desktop Computers and Servers" means computers, including desktop and laptop computers, or servers, used for general computing functions under end user control (such as but not specifically limited to email, general purpose Internet browsing, and office suite productivity tools). The use of Software in systems and solutions that provide dedicated functionality (other than as mentioned above) or designed for use in embedded or function-specific software applications, for example but not limited to: Software embedded in or bundled with industrial control systems, wireless mobile telephones, wireless handheld devices, netbooks, kiosks, TV/STB, Blu-ray Disc devices, telematics and network control switching equipment, printers and storage management systems, and other related systems are excluded from this definition and not licensed under this Agreement. "Programs" means: (a) Java technology applets and applications intended to run on the Java Platform, Standard Edition platform on Java-enabled General Purpose Desktop Computers and Servers, and (b) JavaFX technology applications intended to run on the JavaFX Runtime on JavaFX-enabled General Purpose Desktop Computers and Servers. “README File" means the README file for the Software set forth in the Software or otherwise available from Oracle at or through the following URL:

http://www.oracle.com/technetwork/java/javase/documentation/index.html

2. LICENSE TO USE. Subject to the terms and conditions of this Agreement including, but not limited to, the Java Technology Restrictions of the Supplemental License Terms, Oracle grants you a non-exclusive, non-transferable, limited license without license fees to reproduce and use internally the Software complete and unmodified for the sole purpose of running Programs.

3. RESTRICTIONS. Software is copyrighted. Title to Software and all associated intellectual property rights is retained by Oracle and/or its licensors. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer Software. You acknowledge that the Software is developed for general use in a variety of information management applications; it is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use the Software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Oracle or its licensors is granted under this Agreement. Additional restrictions for developers and/or publishers licenses are set forth in the Supplemental License Terms.

4. DISCLAIMER OF WARRANTY. THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. ORACLE FURTHER DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.

5. LIMITATION OF LIABILITY. IN NO EVENT SHALL ORACLE BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR DATA USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR TORT, EVEN IF ORACLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. ORACLE'S ENTIRE LIABILITY FOR DAMAGES HEREUNDER SHALL IN NO EVENT EXCEED ONE THOUSAND DOLLARS (U.S. $1,000).

6. TERMINATION. This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate

immediately without notice from Oracle if you fail to comply with any provision of this Agreement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right. Upon termination, you must destroy all copies of Software.

7. EXPORT REGULATIONS. You agree that U.S. export control laws and other applicable export and import laws govern your use of the Software, including technical data; additional information can be found on Oracle's Global Trade Compliance web site (Export Regulations). You agree that neither the Software nor any direct product thereof will be exported, directly, or indirectly, in violation of these laws, or will be used for any purpose prohibited by these laws including, without limitation, nuclear, chemical, or biological weapons proliferation.

8. TRADEMARKS AND LOGOS. You acknowledge and agree as between you and Oracle that Oracle owns the ORACLE and JAVA trademarks and all ORACLE- and JAVA-related trademarks, service marks, logos and other brand designations ("Oracle Marks"), and you agree to comply with the Third Party Usage Guidelines for Oracle Trademarks currently located at http://www.oracle.com/us/legal/third-party-trademarks/index.html. Any use you make of the Oracle Marks inures to Oracle's benefit.

9. U.S. GOVERNMENT LICENSE RIGHTS. If Software is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in Software and accompanying documentation shall be only those set forth in this Agreement.

10. GOVERNING LAW. This agreement is governed by the substantive and procedural laws of California. You and Oracle agree to submit to the exclusive jurisdiction of, and venue in, the courts of San Francisco, or Santa Clara counties in California in any dispute arising out of or relating to this agreement.

11. SEVERABILITY. If any provision of this Agreement is held to be unenforceable, this Agreement will remain in effect with the provision omitted, unless omission would frustrate the intent of the parties, in which case this Agreement will immediately terminate.

12. INTEGRATION. This Agreement is the entire agreement between you and Oracle relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

SUPPLEMENTAL LICENSE TERMS

These Supplemental License Terms add to or modify the terms of the Binary Code License Agreement. Capitalized terms not defined in these Supplemental Terms shall have the same meanings ascribed to them in the Binary Code License Agreement. These Supplemental Terms shall supersede any inconsistent or conflicting terms in the Binary Code License Agreement, or in any license contained within the Software.

A. SOFTWARE INTERNAL USE FOR DEVELOPMENT LICENSE GRANT. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the README File incorporated herein by reference, including, but not limited to the Java Technology Restrictions of these Supplemental Terms, Oracle grants you a non-exclusive, non-transferable, limited license without fees to reproduce internally and

use internally the Software complete and unmodified for the purpose of designing, developing, and testing your Programs.

B. LICENSE TO DISTRIBUTE SOFTWARE. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the README File, including, but not limited to the Java Technology Restrictions of these Supplemental Terms, Oracle grants you a non-exclusive, non-transferable, limited license without fees to reproduce and distribute the Software, provided that (i) you distribute the Software complete and unmodified and only bundled as part of, and for the sole purpose of running, your Programs, (ii) the Programs add significant and primary functionality to the Software, (iii) you do not distribute additional software intended to replace any component(s) of the Software, (iv) you do not remove or alter any proprietary legends or notices contained in the Software, (v) you only distribute the Software subject to a license agreement that protects Oracle's interests consistent with the terms contained in this Agreement, and (vi) you agree to defend and indemnify Oracle and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software. The license set forth in this Section B does not extend to the Software identified in Section D.

C. LICENSE TO DISTRIBUTE REDISTRIBUTABLES. Subject to the terms and conditions of this Agreement and restrictions and exceptions set forth in the README File, including but not limited to the Java Technology Restrictions of these Supplemental Terms, Oracle grants you a non-exclusive, non-transferable, limited license without fees to reproduce and distribute those files specifically identified as redistributable in the README File ("Redistributables") provided that: (i) you distribute the Redistributables complete and unmodified, and only bundled as part of Programs, (ii) the Programs add significant and primary functionality to the Redistributables, (iii) you do not distribute additional software intended to supersede any component(s) of the Redistributables (unless otherwise specified in the applicable README File), (iv) you do not remove or alter any proprietary legends or notices contained in or on the Redistributables, (v) you only distribute the Redistributables pursuant to a license agreement that protects Oracle's interests consistent with the terms contained in the Agreement, (vi) you agree to defend and indemnify Oracle and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software. The license set forth in this Section C does not extend to the Software identified in Section D.

D. JAVA TECHNOLOGY RESTRICTIONS. You may not create, modify, or change the behavior of, or authorize your licensees to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax", "javafx", "sun", “oracle" or similar convention as specified by Oracle in any naming convention designation. You shall not redistribute the Software listed on Schedule 1.

E. SOURCE CODE. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of this Agreement. Source code may not be redistributed unless expressly provided for in this Agreement.

F. THIRD PARTY CODE. Additional copyright notices and license terms applicable to portions of the Software are set forth in the THIRDPARTYLICENSEREADME file set forth in the Software or otherwise available from Oracle at or through the following URL: http://www.oracle.com/technetwork/java/javase/documentation/index.html. In addition to any terms and conditions of any third party opensource/freeware license identified in the THIRDPARTYLICENSEREADME file, the disclaimer of warranty and limitation of liability provisions in paragraphs 4 and 5 of the Binary Code License Agreement shall apply to all Software in this distribution.

G. TERMINATION FOR INFRINGEMENT. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right.

H. INSTALLATION AND AUTO-UPDATE. The Software's installation and auto-update processes transmit a limited amount of data to Oracle (or its service provider) about those specific processes to help Oracle understand and optimize them. Oracle does not associate the data with personally identifiable information. You can find more information about the data Oracle collects as a result of your Software download at http://www.oracle.com/technetwork/java/javase/documentation/index.html.

For inquiries please contact: Oracle America, Inc., 500 Oracle Parkway, Redwood Shores, California 94065, USA.

License for Archived Java SE Technologies; Last updated 13 March 2012.

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

B

Setting Up a Virtual Network TAP and L2TP Tunnel

This appendix provide information on capturing ethernet traffic on a web server instance and transporting it to the instance with the RUEI NPA Collector using a Virtual Ethernet Network TAP and a Layer 2 Tunneling Protocol (L2TP) tunnel. This is achieved using standard Linux kernel network functions and userspace tooling.

Before you begin

Real User Experience Insight (RUEI) performs advanced real-time analysis by its Network Protocol Analyzer (NPA) Collector on ethernet network traffic, to monitor communication between clients(browsers) and webservers.

Traditionally Switch Port Analyzer (SPAN) ports on physical network switches, or physical network Terminal Access Point (TAP) devices were deployed to, non-intrusively, mirror ethernet network traffic to the NPA Collector.

There is no requirement by the NPA Collector on how it receives a mirror of ethernet traffic, as long as the mirror contains all ethernet packets unchanged. However, in cloud environments, for example with heavily virtualized networks, capturing ethernet traffic using physical devices may not be feasible or possible, and therefore requires a different approach.

This appendix describes a best practice, developed over the years, to capture ethernet traffic on web application servers and transport the captured traffic over the network to the RUEI NPA Collector, using a Virtual Ethernet Network TAP and a Layer 2 Tunneling Protocol (L2TP) tunnel:

- **Virtual Ethernet Network TAP**: Traffic mirror using the Linux Traffic Control (TC) framework.
- **Ethernet Tunnel**: Transport of mirrored traffic over the network to the NPA Collector using L2TP

Before using the NPA Collector, remember that:

- The solution is shipped as a best practice.
- You are responsible for ensuring the NPA Collector has ethernet access.
- You must ensure that script usage complies with your company's security policies.

This best practice is achieved using standard Linux kernel network functions and userspace tooling. The following set of RPM packages provides an easy-to-use set of tools to manage Virtual TAPs and L2TP tunnels.

- *ux-tunnel-transmit*
- *ux-tunnel-receive*

> **Note:**
>
> Please note that similar restrictions in this context apply to the traditional deployment using physical TAPs, described in TAPs in the *Oracle Real User Experience Insight Installation Guide*, to the virtual TAPs. Which means that, for example, high network latencies or saturated networks can cause packet loss and/or heavy reordering, affecting the reporting accuracy.

Operation, System Impact, and Bandwidth Considerations

The Virtual TAP and L2TP tunnel solution is deployed in software, and handled by the Linux kernel, impacting:

- **Server network packet scheduling**
  - Egress: default queuing discipline on the mirrored interface is changed to ‘prio’
  - Ingress: default queuing discipline on the mirrored interface is enabled
- **Server performance**
  - Virtual TAP:
    - Various extra virtual interfaces (in promiscuous mode) are added and in use:
      - Virtual Ethernet Pair
      - Bridge
    - Linux Traffic Control (TC) filters:
      - Packet mirror (per configured port or all traffic)
      - Loop protection
  - L2TP Tunnel: encapsulation and decapsulation
  - IPv4/6 kernel parameters adjusted to prevent unneeded kernel packet handling (sysctl)
  - Additional routing table to prevent unneeded kernel packet handling
- **Security**
  - Firewall adjusted to allow L2TP protocol (115) traversal
  - Firewall adjusted to prevent mirrored traffic to (re)enter the kernel connection tracking system
- **Bandwidth**
  - Additional bandwidth is needed to transmit encapsulated traffic over a tunnel:
    - Mirrored **Ingress and Egress** traffic from the ethernet device is **aggregated** and encapsulated with L2TP on the egress of that **same** ethernet device!
    - L2TP protocol overhead
    - IP fragmentation overhead
- **Other**
  - Automatic loading and unloading of L2TP kernel modules

Prerequisites

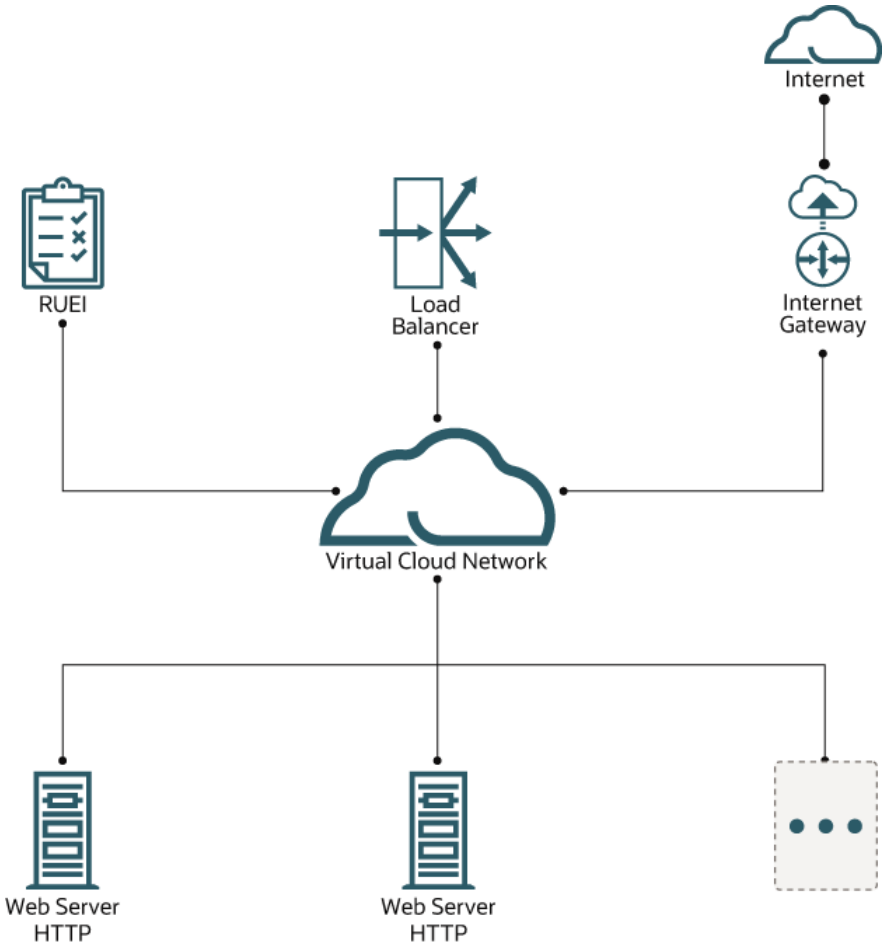
- System with RUEI installed (Which includes the RUEI NPA Collector)
- One or more running Web Application Servers with a deployed application (remote system)
- Network connectivity between both systems
- Remote Secure Shell (SSH) access to both systems
- Root / super user (sudo) access on both systems
- Sufficient network bandwidth to tunnel the captured network traffic
- Virtual TAP / Transmit side:
  - iproute
  - iproute-tc (when using iproute version 5)
  - kernel-modules-extra (starting with Oracle Linux 8)
  - veth kernel module
  - bridge kernel module
  - L2TP kernel module
  - UEK kernel version
    - UEK4 update 7
    - UEK5 update 1

It is possible to monitor SSL (HTTPS) protected web application servers. Please note that the NPA Collector is not capable of monitoring ephemeral based cipher suites. In that case, configure the SSL off-loader to use non-ephemeral cipher suites.

Architecture

A common deployment architecture contains a load balancer in front of several HTTP web servers. Browsers will connect via an Internet Gateway with a Load Balancer within a Virtual Cloud Network. The Load Balancer may perform SSL encryption and decryption. The Load Balancer forwards the HTTP requests, to one of the Web Servers in the backend set.

On each backend Web Server, using a Virtual Ethernet TAP, HTTP requests are mirrored and sent over the L2TP tunnel to the RUEI instance. The RUEI instance handles multiple tunnels concurrently.



Installation Choices

There are two installation methods:

- Step-by-Step Configuration
  1. Ethernet Tunnel on the RUEI System
  2. Virtual Ethernet TAP and Ethernet Tunnel on the Web Application Server System
- Automated
  1. Ethernet Tunnel on the RUEI system using the ‘Step-by-Step Configuration’
  2. Virtual Ethernet TAP and Ethernet Tunnel on the Web Application Server System using ‘Automated Install’

Step-by-Step Configuration

Repeat the instructions in this appendix for each web server instance or load balancer instance you want to monitor. The RUEI instance is capable of handling multiple tunnels concurrently.

> **Note:**
>
> Commands in the following configuration steps must be run as root user or using SUDO.

Step 1: Configure Firewalls

Network Firewalls

After the ethernet traffic is captured by the Virtual Network Ethernet TAP on the web application server system, it will be tunneled over L2TP to the RUEI system.

The network should allow L2TP traffic from the web application system to the RUEI system.

Please update all firewalls in the involved networks to allow L2TP traffic: IP protocol 115.

Warning: L2TP has IP protocol number: 115 Please note that L2TP is a protocol on top of IP and not on top of TCP(/IP). An often made mistake is that TCP (and/or UDP) *port number*115 is incorrectly enabled in security lists, while instead IP *protocol* number 115 should be enabled.

Linux Firewalls on Both RUEI and Web Application Server Systems

Firewall RUEI System

> **Note:**
>
> When using OCI Marketplace, the RUEI system firewall is already configured to allow L2TP.

To allow L2TP traffic flowing into the RUEI system, enable the following rules in the firewall. For example, when using native Oracle Linux `firewalld`:

```
firewall-cmd --zone=public --add-protocol=115
firewall-cmd --zone=public --add-protocol=115 --permanent
```

Firewall Web Application Server

In general the Oracle Linux firewall allows / trusts all traffic originating from the local machine, flowing out into the network. In case the system has restrictions on the output flows, use the following command on the web application server to allow L2TP traffic:

```
firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -p l2tp -j ACCEPT
firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0 -p l2tp -j
ACCEPT
```

Step 2: Set Up the RUEI Virtual Ethernet TAP and L2TP Tunnel

L2TP Tunnel on RUEI

Preparation

The L2TP tunnel configuration requires the proper tunnel source and destination endpoint IPs. All instructions in this chapter must be performed on the RUEI instance, using SSH, unless otherwise instructed.

Source IP (local, RUEI)

Determine the L2TP tunnel source IP using:

```
ip address show
```

Sample Output: `10.10.10.10`

Destination IP (remote, web server)

Determine the L2TP tunnel destination IP using:

```
host <dns-name-WEBSERVER>
```

Sample Output: `10.10.10.11`

Caution

When the WEBSERVER is behind Network Address Translation (NAT), not its locally configured IP (as can be determined using `ip address show` on the WEBSERVER instance) must be used, but the NAT address must be used. Using the host (DNS resolve) command, as above, provides the correct IP address in most circumstances.

Install L2TP Tunnel on RUEI

Run the following command as the root user:

```
cd /root/ruei
./ruei-install.sh rpms/ux-tunnel-receive-*.rpm
```

> **Note:**
>
> When using OCI Marketplace, the RPM is already installed.

Setup

Configure the L2TP Tunnel

1. As a root user, edit the configuration file `tunnels.conf`./`opt/ruei` should be replaced with the directory the customer configured in his installation for RUEI_HOME (can be seen in `/etc/ruei.conf`).

   > **Note:**
   >
   > The initial `tunnels.conf` file was created after installing the RPM file.

2. Configure the receive tunnel using tunnel source and destination IPs retrieved above. Add a line to the `tunnels.conf` file using the following format:

   ```
   receiving-side-ip transmit-side-ip - -
   ```

   For example:

   ```
   10.10.10.10 10.10.10.11 - -
   ```

   This configuration instructs the tunnel helper tooling to setup a L2TP tunnel (receive side) between 10.10.10.10 (RUEI) and 10.10.10.11 (WEBSERVER).

   > **Note:**
   >
   > The receive side can handle multiple tunnels. For each tunnel, add a line to the configuration file.

   > **Note:**
   >
   > See Receive Diagnostics (*Verify Incoming Tunnels*) for tunnels which are coming in and for the tunnel configuration. In case of NAT, the correct tunnel-ID will be displayed.

> **Note:**
>
> L2TP tunnels require a tunnel ID. By default, the transmit side, uses it's local tunnel source IP as tunnel ID. In case NAT is applied in the network, the Src-IP on the RUEI receive system may be different. If that's the case, make sure to configure the tunnel not just by Src-IP and Dst-IP in the `tunnels.conf` file, but include the tunnel-ID, for example:
>
> ```
> 10.10.10.10 192.168.1.11 - 10.10.10.11
> ```

> **Note:**
>
> The tunnel ID may be formatted as an IP-address or as a number. In case of IP clashes between multiple tunnels, use a unique number instead of an IP-address.

Start L2TP tunnel

Reload the tunnel service:

```
systemctl reload ux-tunnel-receive
```

When stopping the tunnel service (`systemctl stop ux-tunnel-receive`), instead of reloading the tunnel service, the RUEI collector may stop automatically as well. This is because the tunnel aggregation interface has been removed. The watchdog will automatically attempt to restart the collector. A restart will succeed when at least one suitable network interface is detected.

Verify the L2TP Tunnel

List the L2TP tunnel interfaces:

```
ip link
```

Sample Output:

```
10: ruei-mtun: <BROADCAST,NOARP,PROMISC,UP,LOWER_UP> mtu 64000 qdisc
noqueue state UNKNOWN link/ether [MAC] brd ff:ff:ff:ff:ff:ff
11: ruei-mtun-00000: <BROADCAST,NOARP,PROMISC,UP,LOWER_UP> mtu 64000
qdisc pfifo_fast state UNKNOWN qlen 1000 link/ether [MAC] brd
ff:ff:ff:ff:ff:ff
```

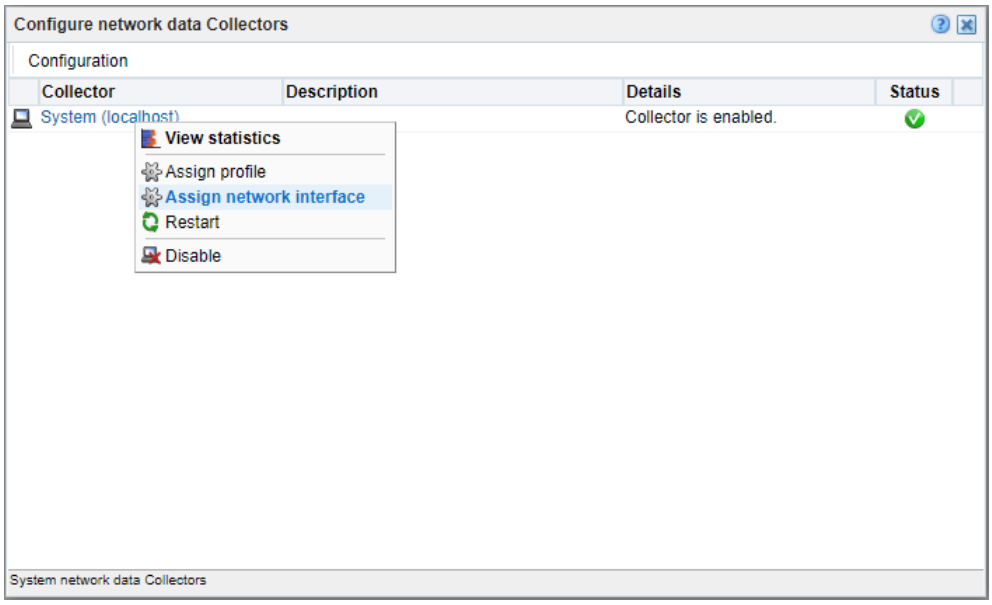
For more advanced tunnel setups, `ux-tunnel-receive config` can be used on the fly to add and remove tunnels, without manually editing the configuration file and without interrupting traffic tunneling and capture for existing on-line tunnels.

Set Up the RUEI NPA Collector

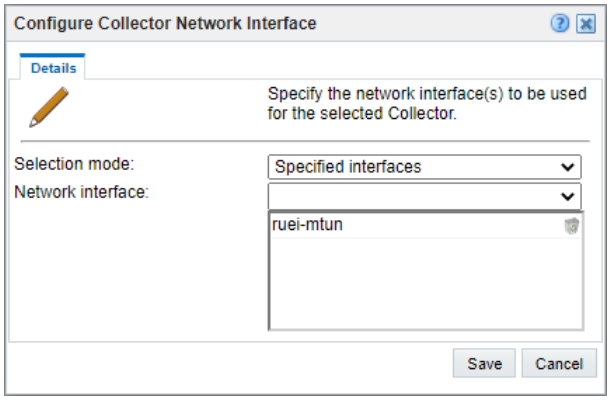
> **Note:**
>
> **This section can be skipped if your RUEI instance is deployed from OCI Marketplace.**

Configure RUEI NPA Collector to only use the L2TP tunnel aggregation device, and discard any other tunnel interfaces. Log in to the RUEI web user interface as admin, and perform the following steps:

1. Click on the **Configuration** tab.
2. Click on the **Collector Profiles** in the left pane.
3. Click on **Configure network data Collectors** in the main pane.
4. Right click on **System (localhost)** and click **Assign network interface**.
5. Set **Selection mode** to **Specified interfaces**.



6. Add the 'ruei-mtun' interface.



7. Click **Save**.

The RUEI NPA Collector will automatically be restarted after clicking **Save**. It takes a couple of minutes before the changes are activated.

Warning: Do not add the `ruei-mtun-<NUMBER>` interfaces to the list of interfaces of a collector. The tunnel tooling aggregates all traffic from the `ruei-mtun-<NUMBER>` interfaces to `ruei-mtun`.

Virtual Ethernet TAP and L2TP Tunnel on WEBSERVER

Preparation

The L2TP tunnel configuration requires the proper tunnel source and destination endpoint IPs. In addition, the Virtual Ethernet TAP requires the network interface name which will be mirrored and which TCP/IP ports will be mirrored.

All instructions in this chapter must be performed on the WEBSERVER instance, using SSH, unless otherwise instructed.

Source IP (local, web server)

You can determine the L2TP tunnel source IP using the following command:

```
ip address show
```

Sample Output: `10.10.10.11`

Destination IP (remote, RUEI)

You can determine the L2TP tunnel destination IP using the following command:

```
host <dns-name-RUEI>
```

Example Output: `10.10.10.10`

Caution: When the RUEI is behind Network Address Translation (NAT), not the locally configured IP (as can be determined with `ip address show` on the RUEI instance) must be used, but the NAT (outside) address must be used. Using the host (DNS resolve) command, as above, provides the correct IP address in most circumstances.

Web server port number

Please consult the configuration of your web server for the TCP/IP port number the server is running on. Alternatively, use the auto port/service detection tools as discussed in Transmit Diagnostics (*Verify Incoming Traffic to Local Services*).

Example: `port 80`

Mirror network interface

Please consult the configuration of your web server and your network setup to determine the name of the ethernet network interface the Virtual Ethernet Network TAP will be mirroring. Alternatively, or to validate, use `netstat -tpln`, `ip link`, and `tcdump` tools to locate on which network interface the clients to web server communication takes place.

For example, `ens3`

Installation

Copy the `ux-tunnel-transmit` RPM from the RUEI system to the application web server and install it as root user:

```
rpm -ivh ux-tunnel-transmit-*.rpm
```

> **Note:**
>
> Copy the `ux-tunnel-transmit` RPM from the RUEI system to the application web server and install it as root user: The RPM can be found in `/root/ruei/rpms`.

Setup

Configure Virtual Network TAP and L2TP tunnel

1. As root user, create the configuration file:

   ```
   cd /opt/ruei/tunnel/transmit/conf
   cp tunnel.conf.example tunnel.conf
   ```

2. Configure the transmit tunnel using tunnel source and destination IPs retrieved above. Add a line to the `tunnel.conf` using the following format:

   ```
   receiving-side-ip transmit-side-ip mirror-if-name mirror-port-list
   ```

   Example: `10.10.10.10 10.10.10.11 ens3 i80`

   This configuration instructs the helper tooling to setup a Virutal Network TAP and L2TP tunnel (transmit side) between 10.10.10.11 (WEBSERVER) and 10.10.10.10 (RUEI).

> **Note:**
>
> Notice the ‘i’ in front of the port number (80). This instructs the Virtual Tap to mirror ingress traffic. Alternatively, the ‘o’ prefix can be used to mirror egress traffic. It is possible to mirror multiple ports. Add each port (with its prefix) to a comma separated list (no spaces). In addition, it it possible to mirror all traffic by using the keyword ‘all’ in the mirror port list.

> **Note:**
>
> The transmit side can handle one tunnel.

Start Virtual Network TAP and L2TP tunnel

Start the tunnel service.

> **Note:**
>
> If the tunnel service is currently running, stop it first, and then restart the service.

```
systemctl stop ux-tunnel-transmit
systemctl start ux-tunnel-transmit
```

Verify Virtual Network TAP and L2TP tunnel

List the L2TP tunnel interfaces:

```
ip link
```

Example Output:

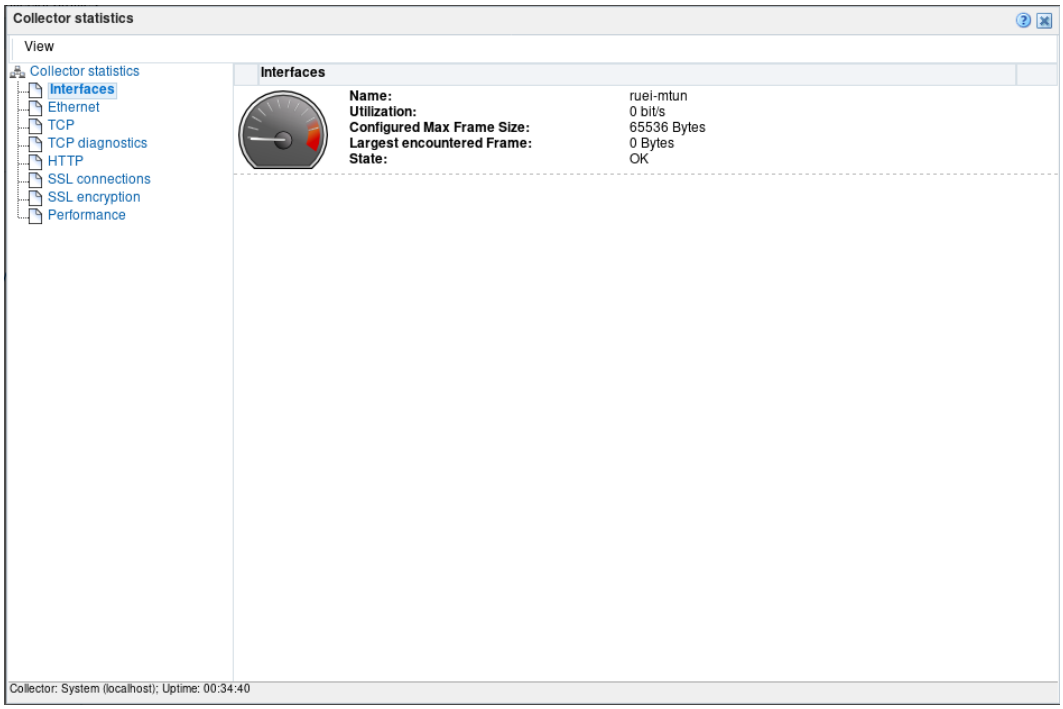
```
15: ruei-tun1: <BROADCAST,NOARP,PROMISC,UP,LOWER_UP> mtu 64000 qdisc
prio master ruei-br1 state UNKNOWN mode DEFAULT group default qlen
1000 link/ether [MAC] brd ff:ff:ff:ff:ff:ff
16: ruei-br1: <BROADCAST,NOARP,PROMISC,UP,LOWER_UP> mtu 64000 qdisc
noqueue state UP mode DEFAULT group default qlen 1000 link/ether [MAC]
brd ff:ff:ff:ff:ff:f
17: ruei-veth1in@ruei-veth1out:
<BROADCAST,MULTICAST,NOARP,PROMISC,UP,LOWER_UP> mtu 64000 qdisc
noqueue state UP mode DEFAULT group default qlen 1000 link/ether [MAC]
brd ff:ff:ff:ff:ff:ff
18: ruei-veth1out@ruei-veth1in:
<BROADCAST,MULTICAST,NOARP,PROMISC,UP,LOWER_UP> mtu 64000 qdisc
noqueue master ruei-br1 state UP mode DEFAULT group default qlen 1000
link/ether [MAC] brd ff:ff:ff:ff:ff:ff
```

Step 3: Verify traffic is received by the RUEI NPA Collector

Before continuing this chapter, make sure that the port numbers of your application(s) specified in the Virtual Network TAP and Tunnel configuration, are also added as protocol in RUEI. Protocol configuration can be found in the RUEI UI in the section **Configuration** > **Security** > **Protocols**.

To verify traffic is received by the RUEI NPA Collector, follow the steps below in the RUEI web user interface as an admin user:

1. Click the **Configuration** tab.
2. Click **Security** in the left pane.
3. Click **Protocols** in the left pane.
4. Click the protocol (**HTTP** or **HTTPS** depending on application configuration) in the main pane to open **Edit Profile Ports** window.
5. Enter the port number, and click **Add**.
6. Click **Save**.
7. Click **Configure network data Collectors** in the main pane.
8. Click **System (localhost)** to open the collector statistics window.



9. Click on the various statistics menu items in the left pane to verify traffic reception. The interface on which mirrored ethernet is coming in, is *ruei-mtun*.

Step 4: Configure an Application Instance/Suite in RUEI

Once traffic is flowing in correctly, it's time to configure a suite or application instance in RUEI. See Working With Suites and Web Services in the *Oracle® Real User Experience Insight User's Guide* for more information.

In case traffic is not flowing in as expected, see Diagnostics.

Automated Install

The installation and configuration of the Virtual Ethernet TAP and Ethernet Tunnel on the Web Application Server Systems has been automated to allow easier setup of multiple monitored systems.

Prerequisites

- Ethernet Tunnel **configured** and **activated** on the RUEI System
  Follow the steps related to the RUEI System in Step-by-Step Configuration.
- Remote SSH access to the target Application Server System [HOST]
  - Key based authentication identity file [IDENTITY-FILE]
  - Remote user name [USER]
  - Password-less sudo access for remote user
  - Sudo must allow access without TTY (no requiretty)
  - Remote user may execute BASH, standard command-line tools and Linux Kernel Network configuration command-line tools (ip, tc, bridge, etc)
- SSH firewall access from the RUEI System to the Application Server System

> **Note:**
>
> Detection and automation has been tested on standard server installations on recent Oracle Linux systems. It's expected that in most cases the detection and installation will succeed, however both detection and automated installation depend on various Linux kernel and Operating System components. This scripting therefore has been created based on a best effort approach and can't support every possible system configuration.

Overview

Each Application Server System is automatically installed from the RUEI System over a SSH connection. The connection with the remote system is achieved using key based authentication. The process is split into two steps:

1. Automatic detection of running HTTP and HTTPS web services on the remote system
2. Automatic installation, configuration and activation of the Virtual Ethernet TAP and Ethernet Tunnel on the remote system using the automatically detected configuration.

Step 1: Automatic detection of running HTTP and HTTPS web services

The automatic detection process tries to discover which web services are running on the remote Application Server System in order to be able to configure the Virtual Ethernet TAP. The Virtual Ethernet TAP is responsible for capturing the Ethernet traffic

(ingress and egress) of the web applications on the remote system. Captured traffic is transmitted to the RUEI System using the Ethernet Tunnel.

The detection process is designed as follows:

- Detect all open IPv4 based services / ports (listen sockets)
- Detect if the service / port implements plain HTTP
- Detect if the service / port implements HTTPS (SSL)
  - Detect if a SSL Ephemeral based cipher suite is used (incompatible with RUEI)

Perform the following steps:

```
cd $RUEI_HOME/tunnel/receive
./ux-tunnel-receive discover tunnel -c USER@HOST -i IDENTITY-FILE
>detect.info
cat detect.info
```

Step 2: Automatic installation, configuration and activation

After this step the new Virtual Ethernet TAP and Ethernet Tunnel is fully configured and operational. Traffic is captured on the remote Application Server System and transferred to, and received on the RUEI System.

The following steps are executed:

- Remote
  - Install the software (ux-tunnel-transmit RPM)
  - Configure the Virtual Ethernet TAP using detected configuration
  - Configure the Ethernet Tunnel using detected configuration
  - Activate the Virtual Ethernet TAP and Ethernet Tunnel
- Local
  - Configure the Ethernet Tunnel using detected configuration
  - Activate the new Ethernet Tunnel

Perform the following steps:

```
./ux-tunnel-receive discover tunnel-install -c USER@HOST -i IDENTITY-FILE -f
detect.info -a ux-tunnel-transmit-<version>.rpm
```

> **Note:**
>
> The ux-tunnel-transmit-<version>.rpm can be found on the RUEI host in the installation directory /root/ruei, or otherwise, when removed, in the shipped RUEI release ZIP.

> **Note:**
>
> To verify the Virtual Network TAP, Ethernet Tunnel and configure or verify RUEI processing of the traffic, see:
>
> - Step-by-step Configuration / Step 3: Verify traffic is received by the RUEI NPA Collector
> - Step-by-step Configuration / Step 4: Configure an Application Instance/ Suite in RUEI
> - Diagnostics

Diagnostics

The Virtual TAP and L2TP best practice contains various virtual network interfaces and the tunnel may need to pass various firewalls. In case the traffic is not received correctly, there are various steps to take, to verify the configuration. There are diagnostics steps for both the transmit and receive side.

Transmit Diagnostics

Verify Incoming Traffic to Local Services

The first step is to verify whether expected web traffic is flowing into the web application server.

Use the following tool, to automatically detect and list traffic (connections) to local services (including web application servers) aggregated by Server IPs and Server Port. By default the tool will look at packets on the tunnel mirror interface (eth0 in this example) previously configured. Supply a different interface to perform the same check on different interfaces.

```
cd /opt/ruei/tunnel/transmit
./ux-tunnel-transmit diag traffic servers
```

Example Output:

```
[info] Detecting server IPs and ports on interface 'eth0', capture
time: 30s, max packets: 1000
[info] Using filter: tcp[tcpflags] == tcp-syn|tcp-ack
[info] Server-IP Port Connections
[info] 10.10.10.01 80 38
[info] 10.10.10.02 7001 3
```

Verify Tunnel Status

Due to various local causes, packets may not be accepted by the L2TP tunnel and are silently dropped. Use the following tool to verify the tunnel status on the transmit side.

```
./ux-tunnel-transmit diag tunnel check
```

Example Output:

```
[info] Interface Local IP Remote IP Tx: packets Tx: errors Tunnel status
[info] ruei-tun1 10.10.10.10 10.10.10.11 0 0 OK
```

Verify L2TP Traffic Leaving the System

Use the following command to verify L2TP encapsulated packets are leaving the transmit system. Eth0 is used below as the interface over which the L2TP packets are leaving the system. Please replace with the actual interface over which L2TP packets are being routed.

```
/usr/sbin/tcpdump -n -nn -i eth0 proto l2tp
```

Example Output:

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
08:03:29.170192 IP 10.10.10.10 > 10.10.10.11: l2tp 106
08:03:29.380252 IP 10.10.10.10 > 10.10.10.11: l2tp 106
08:03:29.590259 IP 10.10.10.10 > 10.10.10.11: l2tp 106
```

Receive Diagnostics

Verify Incoming Tunnels

Use the following tool to automatically detect any incoming L2TP tunnel. Each tunnel configured and started with ux-tunnel-transmit will transmit a keep alive ping every 20 seconds. This keep alive ping is used to detect incoming tunnels.

```
sudo ./ux-tunnel-receive diag monitor detect
```

Example Output:

```
[info ] Detecting tunnels on interface 'any', capture time: 30
[info ] Type Dst-IP Src-IP Tunnel-ID NAT-Status
[info ] L2TP 10.10.10.10 192.168.10.11 10.10.10.11(168430091) yes
[info ] L2TP 10.10.10.10 192.168.10.12 10.10.10.12(168430092) yes
[info ] Detected tunnels: 2
```

L2TP tunnels require a tunnel ID. By default, the transmit side, uses it's local tunnel source IP as tunnel ID. In case NAT is applied in the network, the Src-IP on the RUEI receive system may be different. If that's the case, make sure to configure the tunnel not just by Src-IP and Dst-IP in the tunnels.conf file, but include the tunnel-ID, for example:

```
16859033
```

Verify Tunnel Status

Due to various local causes, packets may not be accepted by the L2TP tunnels and are silently dropped. Use the following tool to verify the status of each tunnel on the receive side.

```
./ux-tunnel-receive diag tunnels check
```

Example Output:

```
[info ] Interface Local IP Remote IP Rx: packets Rx: errors Tunnel
status
[info ] ruei-mtun-00000 10.10.10.10 10.10.10.11 180144698 0 OK
[info ] ruei-mtun-00001 10.10.10.10 10.10.10.12 0 0 OK
```

When a tunnel interface reports errors, its tunnel status will be reported as “FAIL”.

Verify Incoming Traffic on the Tunnel Aggregation Interface

All traffic coming in from all tunnels, are aggregated into ‘ruei-mtun’ on the receive side.

Use the following tool, to automatically detect and list traffic (connections) to mirrored services (including web application servers) aggregated by Server IPs and Server Port. By default, the tool will look at packets on the tunnel aggregation interface, ‘ruei-mtun’.

```
sudo ./ux-tunnel-receive diag traffic servers
```

Example Output:

```
[info] Detecting server IPs and ports on interface 'ruei-mtun',
capture time: 30s, max packets: 1000
[info] Using filter: tcp[tcpflags] == tcp-syn|tcp-ack
[info] Server-IP Port Connections
[info] 10.10.10.01 80 38
[info] 10.10.10.02 7001 3
```

Verify L2TP Traffic Entering the System

Use the following command to verify L2TP encapsulated packets are entering the receive system. Eth0 is used below as the interface over which the L2TP packets are entering the system. Please replace with the actual interface over which L2TP packets are being received.

```
./usr/sbin/tcpdump -n -nn -i eth0 proto l2tp
```

Example Output:

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
```

```
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
08:03:29.170192 IP 10.10.10.11 > 10.10.10.10: l2tp 106
08:03:29.380252 IP 10.10.10.11 > 10.10.10.10: l2tp 106
08:03:29.590259 IP 10.10.10.11 > 10.10.10.10: l2tp 106
```

Other Diagnostics

Interface Statistics

Use `ip -s -d link show` to list the available network interfaces, including interface statistics. Verify each of the virtual interfaces involved with the Virtual TAP / L2TP tunnel don't display errors.

Traffic Control Statistics

Low-level packet filtering and mirroring is done using the Linux Traffic Control(tc) system. Use `tc -s filter show` to display low-level statistics, including error counters.

C

Connecting a Collector to a GRE Tunnel

This appendix describes how to set up a GRE Ethernet (Layer 2) tunnel to a RUEI Collector Engine and how to use a tap with this configuration.
Before attempting this procedure, set up console access to the systems involved. This is required because running a wrong command can take a network offline, possibly severing any connection you have to the server. In this scenario, the console access is required to repair the network.

> **Note:**
>
> Oracle recommends that you set up a virtual network TAP and L2TP Tunnel as it's an easier and more user friendly way of configuring the tunnels. See Setting Up a Virtual Network TAP and L2TP Tunnel for more information.

Introduction and Features of GRE Tunnelling

The *RUEI User's Guide* describes how to locate your installation within a network. GRE Tunnelling allows you to locate the Collector Engine anywhere in your network as long as the tunnel endpoints can communicate with each other. While GRE tunnelling is efficient, the network throughput can decrease because of network throughput overhead caused by the additional headers added to the packets and the CPU time overhead caused by encapsulation and decapsulation of those packets.

GRE Tunnel Requirements

Linux supports GRE Ethernet tunneling from kernel version 2.6.28, and requires an up-to-date version of the `iproute` package containing the utilities (specifically the `IP` utility) to set up and configure GRE Ethernet tunnel (`gretap`) interfaces.

This procedure uses Oracle Linux 6.4 as a base for setting up the GRE Ethernet tunnels, as Oracle Linux version 6 provides a UEK kernel (Linux version 2.6.39-400.109.1.el6uek at time of writing) capable of setting up GRE Ethernet tunnels, as well as the correct version of the iproute package (iproute-2.6.32-23.el6.x86_64 for Oracle Linux 6.4) required to add, delete, or modify GRE Ethernet tunnels.

Oracle Linux 5 is not supported, even if it supports GRE Ethernet tunnels in its UEK kernel (Linux version 2.6.39-400.21.1.el5uek for OEL5.9). As it needs a newer version (2.6.28 or higher) of the `iproute` package capable of setting up the GRE Ethernet tunnels. The required version of the `iproute` package is not officially supported. Hence, is not covered in this procedure.

Overview of Procedure

While this appendix contains details on various aspects of taps and GRE tunnels, the following outlines the process that must be completed:

1. Perform either the manual or scripted GRE setup. See Manual Setup for Basic RUEI Tap and GRE Tunnel and Scripted Setup for Basic RUEI Tap and GRE Tunnel.
2. Ensure that the destination endpoint is only receiving traffic. See Making a Tunnel Unidirectional.
3. Add a tap. See Adding a Virtual Tap.
4. Configure the collector. See Configuring a Collector for GRE Tunnelling.
5. Test the setup. See Testing a GRE Tunnel.
6. Ensure that your configuration is applied even after a reboot. See Making GRE Tunnel Environment Changes Permanent.

Setting Up a Basic RUEI Tap and GRE Tunnel

This section describes creating a single GRE tunnel, and how to set up either endpoint (assuming they are both Oracle Linux version 6 machines) to be able to aggregate either tap (source) traffic or GRE tunnel output (destination) traffic. With this process you can add one or more taps on one machine to the GRE tunnel, have the collector listen to one or many incoming GRE tunnels. This section contains the following sections:

- Prerequisites for a Basic RUEI Tap and GRE Tunnel
- Manual Setup for Basic RUEI Tap and GRE Tunnel
- Scripted Setup for Basic RUEI Tap and GRE Tunnel
- Making a Tunnel Unidirectional
- Adding a Virtual Tap

Prerequisites for a Basic RUEI Tap and GRE Tunnel

The following components are required:

- Two Oracle Linux version 6 system endpoints, with one endpoint set up as a RUEI Collector
- On each system, the following packages must be installed:
  - `iproute2`
  - `tcpdump`
  - `bridge-utils`
- On each system, the following kernel modules must be present and loaded:
  - `ip_gre` - support for GRE tunneling
  - `bridge` - support for bridges
  - `veth` - support for virtual ethernet interfaces

Manual Setup for Basic RUEI Tap and GRE Tunnel

This section describes manually setting up two Oracle Linux version 6 systems, one as a source and the other as destination (with the RUEI Collector). It is an alternative process to Scripted Setup for Basic RUEI Tap and GRE Tunnel.

To prepare both systems, do the following:

1. Install a GRE tunnel between the source and destination systems by following the instructions in Configuring a GRE Tunnel Manually.
2. Install a bridge (*BRTUN*) on each of the source and destination systems by following the instructions in Creating and Setting Up a Linux Bridge.
3. On both the source and destination systems, add the local GRE tunnel endpoint interface (*GRETUN*) to the bridge (*BRTUN*) by following the instructions in Adding and Removing Bridge Interfaces .

By following the instructions in Testing a GRE Tunnel you should be able to view the generated test traffic from the source coming through the tunnel, both on the GRE tunnel interfaces as well as on the bridge interfaces on both ends.

Go to section Making a Tunnel Unidirectional.

Scripted Setup for Basic RUEI Tap and GRE Tunnel

This section describes setting up two Oracle Linux version 6 systems using a script, one as a source and the other as destination (with the RUEI Collector). It is an alternative process to Manual Setup for Basic RUEI Tap and GRE Tunnel

To prepare both the systems, use the `tunnelctl` script to create a bridged GRE tunnel as described in Configuring a GRE Tunnel Using the tunnelctl Script.

Making a Tunnel Unidirectional

Both the source and destination systems are set up, but no traffic flows through the bridges or the tunnel. Before we connect any taps to the tunnel source, we need to make sure that the destination GRE tunnel endpoint (*GRETUN* on the destination system) can only receive traffic, not send any over the tunnel.

Also, we need to ensure the GRE tunnel is unidirectional as we only want to monitor traffic, not take part in it. We will use linux traffic shaping to block outgoing traffic for the GRETUN interface on the destination endpoint system.

Do the following steps:

1. Select a handle (HANDLE) to be used for this `qdisc`. For example, reuse the GRE tunnel id (ID).
2. Replace the root qdisc of *GRETUN* with one (prio) that can filter the outgoing traffic, by running the following command as root:

   ```
   tc qdisc replace dev GRETUN parent root handle HANDLE: prio
   ```

3. Add a filter to pass all outgoing GRE traffic from the machine so that it does not get mirrored. Run the following commands:

```
tc filter add dev GRETUN parent HANDLE: \
protocol all prio 1 u32 \
match u32 0 0 flowid HANDLE:1 \
action drop
```

The GRE tunnel is now unidirectional. You can test this by generating traffic on one system by using the ping option and viewing it on the other system by using the `tcpdump` as described in Testing a GRE Tunnel, and then re-doing the test in the other direction. You should see traffic flowing from the source (tap) to destination (RUEI Collector), but no traffic from the destination (RUEI Collector) to source (tap).

Adding a Virtual Tap

We will now create a virtual tap for one of the local interfaces on the source system, by performing the following steps:

1. Choose an interface (*ETH*) on the source system whose traffic you want to monitor on the destination RUEI system.
2. Create a virtual tap for the interface chosen in step 1, by using the instructions in Creating a Virtual Tap.
3. On the source system, add the created tap interface (*TAP*) to the bridge (*BRTUN*), by using the instructions in Adding and Removing Bridge Interfaces .
4. On the destination system, test the incoming GRE tunnel traffic, by running the following command as `root`:

```
tcpdump -i GRETUN -c 100 -n
```

> **Note:**
>
> As described in the Testing the Tap, the traffic you see on the bridge interface on the destination system should now be the same as the traffic on the bridge interface on the source system.

Configuring a Collector for GRE Tunnelling

After we have a GRE tunnel set up and tested, the collector can be configured to listen to the traffic on the GRE Tunnel. To enable the RUEI Collector Engine to listen to the GRE Ethernet tunnel, do the following:

1. In the RUEI web interface, from **Configuration**, go to **Security**, click **Collector Profiles**, and select the appropriate profile.
2. Right-click the appropriate collector, and select **Assign Network Interface**.
3. Select **Specified Interfaces**.
4. Select the **GRE interface** from the dropdown to enable it for monitoring.
5. Click **Save** to confirm the changes.
6. Make sure there is not a firewall filtering any packets coming through the interface. It is outside the scope of this document to explain how to perform this task. Here are few tips:

- The firewall should be set up to totally ignore the interface, not set up to route everything to a single other interface in the GRE tunnel network. This is because any filtering causes CPU overhead, which can have a negative effect on throughput.
- Any generic firewall rules, that is rules covering all interfaces can also apply to the interface currently being configured, must be altered not to cover this interface. As a workaround, add new rules to ignore this interface.

7. Disable any network throttling that might affect the interface. It is outside the scope of this document to explain how to perform this task.

Configuring a GRE Tunnel Using the tunnelctl Script

This section describes how to create a GRE tunnel using the `tunnelctl` script provided with RUEI.

Requirements for tunnelctl Script

The following components are required:

- Two Oracle Linux version 6 system endpoints, with one endpoint set up as a RUEI Collector
- On each system, the following packages must be installed:
  - `iproute2`
  - `tcpdump`
  - `bridge-utils`
- On each system, the following kernel modules must be present and loaded:
  - `ip_gre - support for GRE tunneling`
  - `bridge - support for bridges`
  - `veth - support for virtual ethernet interfaces`
- The two endpoints are able to reach each other (for example, tested using ping). The relevant ports must have been opened in any firewalls, both on the endpoints as well as on any router in between.
- The two endpoints must have an executable copy of the `tunnelctl` script.
- `root` user access is available on both endpoints.

Setting Up a Tunnel Endpoint

Perform the following steps to set up the first endpoint:

1. Note the IP address of the local and remote endpoints.
2. Create a numeric Identifier (ID) to be used for both endpoints, for example `123`. This ID will be used to identify the tunnel on both sides.

3. Log in as root using `ssh` and run the following command:

   ```
   tunnelctl create gre Local_IP Remote_IP ID
   ```

   Where,

   - *Local_IP* is the address of the current server.
   - *Remote_IP* is the address of the remote server.
   - *ID* is the identifier you created in the previous step.

4. Check that the tunnel has been created:

   ```
   tunnelctl list
   ```

   An interface named gre*ID* should be listed.

Setting Up Other Endpoints

To set up a tunnel both endpoints must be configured. The 'other' endpoint can be a switch or router capable of duplicating streams and sending them out through a GRE Ethernet tunnel, or it may be another Linux server where any duplication/streaming can be set up.

If the other endpoint is a router or switch capable of duplicating streams and sending them out through a GRE Ethernet tunnel, refer to the product documentation for any steps that might be necessary.

If the other endpoint is a Linux server, repeat the steps in Setting Up a Tunnel Endpoint on the second endpoint (noting that you need to reverse the local and remote IP addresses when creating the tunnel).

Configuring a GRE Tunnel Manually

This section describes how to create a GRE tunnel manually.

Requirements for Configuring a GRE Tunnel Manually

The following components are required:

- Two Oracle Linux version 6 system endpoints, with one endpoint set up as a RUEI Collector
- On each system, the following packages must be installed:
  - `iproute2`
  - `tcpdump`
  - `bridge-utils`
- On each system, the following kernel modules must be present and loaded:

  ip_gre - support for GRE tunneling

- The two endpoints are able to reach each other (for example, tested using ping). The relevant ports must have been opened in any firewalls, both on the endpoints as well as on any router in between.
- The two endpoints must have an executable copy of the `tunnelctl` script.
- `root` user access is available on both endpoints.

Setting Up a Tunnel Endpoint Manually

Perform the following steps to set up the first endpoint:

> **Note:**
>
> Do not bring the interface up until completing this procedure.

1. Note the IP address of the local and remote endpoints.
2. Create a numeric Identifier (ID) to be used for both endpoints, for example `123`. This ID will be used to identify the tunnel on both sides.
3. Log in as root using `ssh` and enter the following command to load the GRE modules in the Linux kernel:

   ```
   modprobe ip_gre
   ```

4. Run the following command to check that the GRE modules are loaded in the Linux kernel:

   ```
   lsmod | grep gre
   ```

5. Log in as root using `ssh` and run the following command:

   ```
   ip link add ID type gretap local Local_IP remote Remote_IP
   ```

   Where,

   - *Local_IP* is the address of the current server.
   - *Remote_IP* is the address of the remote server.
   - *ID* is the identifier you created in the step 2.

6. Check that the tunnel has been created:

   ```
   ip link show
   ```

   An interface named gre*ID* should be listed.

7. Configure the kernel not to route anything coming from the tunnel interface by performing the steps in Configuring an Interface for Mirrored Traffic, taking care to swap *IFACE* with the interface name you are currently preparing (for example *greID*).

Setting Up Other Endpoints

To set up a tunnel both endpoints must be configured. The 'other' endpoint can be a switch or router capable of duplicating streams and sending them out through a GRE Ethernet tunnel, or it may be another Linux server where any duplication/streaming can be set up.

If the other endpoint is a router or switch capable of duplicating streams and sending them out through a GRE Ethernet tunnel, refer to the product documentation for any steps that might be necessary.

If the other endpoint is a Linux server, repeat the steps in Setting Up a Tunnel Endpoint Manually on the second endpoint (noting that you need to reverse the local and remote IP addresses when creating the tunnel).

Creating and Setting Up a Linux Bridge

This section describes how to create and set up a linux bridge which will act as a layer 2 *hub* for mirrored data. You can add virtual taps and GRE tunnels to the bridge to create the required configuration. Setting up multiple bridges is also possible, but such a configuration is beyond the scope of this document.

- Requirements for a Linux Bridge
- Creating a Linux Bridge
- Adding and Removing Bridge Interfaces

Requirements for a Linux Bridge

The following components are required:

- The following packages must be installed:
  - `iproute2`
  - `tcpdump`
  - `bridge-utils`
- The following kernel module must be present and loaded:

  `bridge`

Creating a Linux Bridge

Create a bridge by completing the following steps:

1. Log in as root using `ssh` and run the following command:

   `brctl addbr BRTUN`

   Where,

   - *BRTUN* is the name of the bridge.
2. Run the following command to check the bridge was created:

   `brctl show`
3. Run the following command to configure the bridge to act as a (dumb) hub instead of a switch:

   `brctl setfd BRTUN 0brctl setageing BRTUN 0`
4. Run the following command to configure the bridge to be silent:

   `brctl stp BRTUN off`

5. Configure the kernel not to route anything coming from the bridge interface by performing the steps in Configuring an Interface for Mirrored Traffic, taking care to swap *IFACE* with the interface name you are currently preparing (for example *BRTUN*>).
6. Run the following command to activate the bridge and set it to accept all traffic:

   ```
   ip link set BRTUN promisc on arp off up
   ```

Adding and Removing Bridge Interfaces

To add an interface (*IFACE*) to a bridge, run the following command:

```
brctl addif BRTUN IFACE
```

To remove an interface (*IFACE*) from a bridge, run the following command:

```
brctl delif BRTUN IFACE
```

To view the current configuration of the bridge, run the following command:

```
brctl show
```

Testing a GRE Tunnel

After we have a GRE tunnel set up between two endpoints, and an interface for mirrored traffic to ensure that no mirrored traffic is routed on the linux (virtual) machine, an unused GRE Ethernet tunnel can be tested by running `ping` on one end and `tcpdump` on the other to see the GRE tunnel traffic. In the following steps, the two endpoints are referred to as the source and the destination, where the source signifies the endpoint where `ping` is running, and the destination is where `tcpdump` is used to verify the traffic:

1. Make sure the GRE tunnel interface on either endpoint system is up, by running the following command as `root` on both systems:

   ```
   ip li set GRETUN up
   ```

2. Send ICMP packets through the tunnel, by running the following command as `root` on the source system:

   ```
   ping -I GRETUN 127.1.1.1
   ```

   The IP address has specifically been chosen so that it does not get inadvertently routed anywhere, as it is a local address. Using `ping -I` means that the ICMP packets only get sent over the GRE tunnel, restricting the visibility to the destination endpoint.

3. Check that the GRE encapsulated tunnel traffic was received, by running the following command as `root` on the destination system. Where, `ETH` is the interface the tunnel is routed over (the local endpoint, typically eth0), not the tunnel interface itself.

   ```
   tcpdump -i ETH -c 100 proto gre
   ```

   You should see ARP and/or ICMP requests for the above IP address wrapped in GRE packets (GREv0) similar to the following:

   ```
   … IP server_A > server_B: GREv0, length 46:
   ARP, Request who-has 127.1.1.1 tell server_A, length 28
   … IP server_A > server_B: GREv0, length 102:
   IP server_A > 127.1.1.1: ICMP echo request, id 62057,
   seq 1, length 64
   ```

Creating a Virtual Tap

This section describes a generic method of creating a “tap" network interface that will provide mirrored traffic from any other live interface on the Oracle Linux version 6 machine. This method uses linux traffic shaping to mirror incoming and outgoing data from an interface and copy that network traffic to a set of newly created virtual ethernet interfaces.

A set of two virtual ethernet interfaces are connected to each other in such a way that any data flowing into one will flow out of the other, in this sense they act as a virtual NIC cable. These virtual ethernet interfaces are commonly used in virtual networking.

Any local interface can be mirrored by using this method, including the interface the controlling `ssh` connection and the interface carrying GRE tunnel traffic. This is possible because GRE traffic will be filtered out of any mirrored traffic by one of the traffic shaping rules in this chapter.

Introduction to Virtual Taps

This procedure creates a pair of virtual interfaces, one called “*ETH*mirror" and the other called “*ETH*tap". For example, if you want to tap interface `eth0`, you first create a set of virtual interfaces called `eth0mirror` and `eth0tap`. The interfaces are named this way to help keep them apart from any other mirroring setups on the system, since this method allows us to mirror more than one local interface into the GRE tunnel. From now on we will reference them as *ETH*, *MIRROR* and *TAP*.

The *ETH* interface will have its traffic mirrored on the *MIRROR* interface. All traffic flowing through the *MIRROR* interface will also be seen on the *TAP* interface since they are a virtual ethernet pair, so that you can use that *TAP* interface in any network configuration (directly or in a bridge) that you want.

The following components are required:

- The tap is to be created on an Oracle Linux version 6 system
- The following packages must be installed:
  - `iproute2`
  - `tcpdump`
- The following kernel module must be present and loaded:

  `veth - support` for virtual ethernet interfaces
- A live interface to be mirrored exists, this interface will be referred to from now on as *ETH*.

Creating the Mirror and Tap Interfaces

Complete the following steps to create the mirror/tap virtual interfaces:

1. Create a pair of virtual interfaces by running the following command as `root`:

   `ip li ad` *`TAP`* `type veth peer name` *`MIRROR`*
2. Activate the interfaces, by running the following commands as `root`:

```
ip li set dev TAP up promisc on arp off
ip li set dev MIRROR up promisc on arp off
```

Configuring the Mirror

Traffic shaping enables the copying of all incoming and outgoing traffic for a given interface (*ETH*) to the newly created `MIRROR` virtual interface. How traffic shaping works is not explained in this document, though individual steps will be annotated.

The mirror setup is simple, though you do need to add an extra filter to prevent any GRE traffic (packet type GREv0, see Testing a GRE Tunnel) from being mirrored. This must be done to ensure that if you are mirroring the interface the GRE tunnel is transported over, you will not force the GRE tunnel to carry its own traffic (a loop) as that would cause the network to fail, and the server to fail.

To mirror the incoming traffic, do the following:

1. Add an ingress qdisc to *ETH* by running the following command as `root`:

   ```
   tc qdisc add dev ETH ingress
   ```

2. Add a filter to pass all incoming GRE traffic to the machine so that it is not mirrored. Run the following commands:

   ```
   tc filter add dev ETH parent ffff: \
   protocol all prio 1 u32 \
   match ip protocol 47 0xff flowid 1:1 \
   action pass
   ```

3. Add a filter to mirror all remaining traffic to our *MIRROR* interface. Run the following commands:

   ```
   tc filter add dev ETH parent ffff: \
   protocol all prio 2 u32 \
   match u32 0 0 flowid 1:2 \
   action mirred egress mirror dev MIRROR
   ```

To mirror all outgoing traffic, do the following:

1. Replace the root qdisc of *ETH* with one (prio) that can filter the outgoing traffic by running the following command as `root`:

   ```
   tc qdisc replace dev ETH parent root handle 10: prio
   ```

2. Add a filter to pass all outgoing GRE traffic to the machine so that it is not mirrored. Run the following commands:

   ```
   tc filter add dev ETH parent 10: \
   protocol all prio 1 u32 \
   match ip protocol 47 0xff flowid 10:1 \
   action pass
   ```

3. Add a filter to mirror all remaining traffic to our *MIRROR* interface. Run the following commands:

   ```
   tc filter add dev ETH parent 10: \
   protocol all prio 2 u32 \
   match u32 0 0 flowid 10:2 \
   action mirred egress mirror dev MIRROR
   ```

> **Note:**
>
> If you deactivate the *MIRROR* interface after completing this procedure, it will disrupt the *ETH* network traffic. Leave the *MIRROR* interface active, since you will only be using the *TAP* interface in the remaining setup, and that interface can be de-activated without any consequences

Testing the Tap

At this point you have two new interfaces, *MIRROR* and *TAP*. The *MIRROR* is used by the traffic shaping rules to mirror the network traffic from *ETH* to, and *TAP* is the virtual interface counterpart of *MIRROR*. Leave *MIRROR* active from now on, and use *TAP* in our networking setup, as long as you make sure it's data is not being routed by the system. To test this, look at the traffic on the *TAP* interface. That traffic should be the same as the traffic on *ETH*, minus the GRE traffic.

1. View the traffic on the *TAP* interface by running the following command as `root`:

   ```
   tcpdump -i TAP -c 100 -n
   ```

2. Compare the output of step 1 with the output of the following command, which is the traffic on *ETH* with the GRE traffic filtered out:

   ```
   tcpdump -i ETH -c 100 -n ! proto gre
   ```

> **Note:**
>
> To see the same output you should run both commands simultaneously. If you run the previous steps simultaneously you will probably see that the output does not line up, but after finding where they align you should see that they are the same.

Preparing an Interface for Mirrored Traffic

This section describes how to ensure that the Linux kernel does not route or filter any packets going through a specific interface.

- Configuring an Interface for Mirrored Traffic
- Adapting the Firewall
- Disabling Network Throttling

Configuring an Interface for Mirrored Traffic

In the following steps *IFACE* denotes the interface that is being set up to accept any packets without routing them.

1. Configure the interface to accept all traffic without responding to arp or multicast packets by running the following command as `root`:

   ```
   ip link set IFACE down promisc on arp off multicast off
   ```

This command also brings the interface down if it was not down already, so that you are not inadvertently routing any data. Do not bring the interface up again until all steps are completed.

2. To make sure the interface will not have an IPv6 address automatically assigned, run the following commands:

```
sysctl -w net.ipv6.conf.IFACE.autoconf=0
sysctl -w net.ipv6.conf.IFACE.accept_ra=0
```

3. Check if the interface has any IPv4 or IPv6 addresses already:

```
ip address show IFACE
```

4. Remove all addresses listed starting with “inet" (IPv4) or “inet6" (IPv6) in the output from the above command (where *IP* is the address you want to remove):

```
ip address delete IP dev  IFACE
```

5. Make sure the interface only respond to ARP requests for its own IP addresses (which it does not receive, so it will never respond):

```
sysctl -w net.ipv4.conf.IFACE.arp_ignore=1
```

6. Turn off reverse path filtering to ensure that the incoming packets are not dropped:

```
sysctl -w net.ipv4.conf.IFACE.rp_filter=0
```

7. Choose an empty routing table number so we can set up (no) routing specifically for the tunnel.

   In this example, we use table number 200. Ensure that the table is empty, by running the following command:

```
ip route show table 200
```

   Set up multiple interfaces on one system by using these steps they can all use the same table, as it will remain empty.

8. Create a routing table rule to have the kernel use the empty table to look up routing information for this interface:

```
ip rule add iif IFACE table 200
```

9. Check that the rule was added by running the following command:

```
ip rule show
```

   Following is an example output:

```
0: from all lookup local
32765: from all iif IFACE lookup 200
32766: from all lookup main
32767: from all lookup default
```

Adapting the Firewall

If a firewall is active on the system, ensure that it is not filtering any packets coming through the interface. It is outside the scope of this document to explain how to do this as there are too many different firewall applications to list here. Here is a short list of pitfalls to take into account:

- The firewall should be set up to totally ignore the interface, not set up to route everything to a single other interface in the GRE tunnel network. This is because any filtering causes CPU overhead, which can have a negative effect on throughput.
- Any generic firewall rules, i.e. rules covering all interfaces can also apply to the interface currently being configured. These rules must be altered not to cover this interface, or new rules should be added to ignore this interface.

Disabling Network Throttling

Some systems have network throttling enabled, this must be removed or turned off for the interface being configured, otherwise some packets of the copied or mirrored network may be dropped. How to change the configuration for network throttling fall outside the scope of this document. Though traffic shaping is used, you should be cautious with respect to the traffic shaping rules introduced in this document (also see,Configuring the Mirror, Configuring the Mirror, and Configuring a GRE Tunnel Using the tunnelctl Script).

Making GRE Tunnel Environment Changes Permanent

When the GRE tunnel configuration is working, create a boot script that executes the setup commands described in this appendix. The script should include items for:

- GRE Ethernet tunnel creation
- Firewall configuration
- Network throttling