Log Correlation Engine 4.2 High Availability Large Scale Deployment Guide

February 19, 2014

(Revision 2)

Table of Contents

Introduction

This document details various configuration methods, architecture examples, and hardware specifications for performance and high availability of large scale deployments of Tenable's Log Correlation Engine (LCE). Please email any comments and suggestions to support@tenable.com.

Some features in LCE 4 specifically address requirements for large deployments such as:

- Higher Events-Per-Second (EPS) processing
- Load-balancing
- Plugin (PRM) management and tuning
- Multiple CPU/core (SMP) utilization
- High Availability backup host (hot spare)

This guide is intended for organizations with logging requirements in the tens of thousands of EPS to assist with properly designing and configuring their LCE implementation to ensure optimal performance and availability.

Although this guide focuses on using SecurityCenter for event and LCE server management, the LCE Manager (included with LCE) offers many of the same options, only without the vulnerability management features.

Familiarity with system logging, network devices and applications, and a basic understanding of Linux is assumed. A working knowledge of network design and storage options is beneficial as well.

Because this is an advanced documentation of LCE capabilities, a strong understanding of LCE and its options is also assumed. Please refer to the documentation available on the Tenable Support Portal to gain the necessary background understanding of LCE installation, configuration, and usage.

Standards and Conventions

Throughout the documentation, filenames, daemons, and executables are indicated with a **courier bold** font such as **gunzip**, **httpd**, and **/etc/passwd**.

Command line options and keywords are also indicated with the **courier bold** font. Command line examples may or may not include the command line prompt and output text from the results of the command. Command line examples will display the command being run in **courier bold** to indicate what the user typed while the sample output generated by the system will be indicated in courier (not bold). Following is an example running of the Unix **pwd** command:

```
# pwd
/opt/local/lce
#
```

Important notes and considerations are highlighted with this symbol and grey text boxes.

Tips, examples, and best practices are highlighted with this symbol and white on blue text.

Overview

EPS (Events-Per-Second) is an industry standard term used to measure performance of Security Information and Event Management (SIEM) solutions.

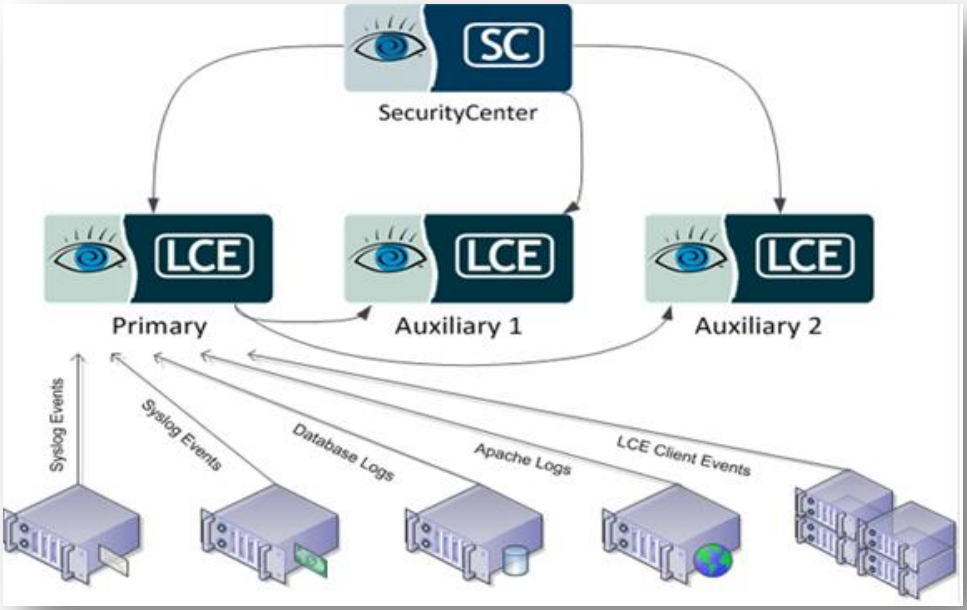
There are many factors that affect the ability of a SIEM solution such as LCE to achieve high EPS. Server hardware, network bandwidth and architecture, storage models, and load distribution all have a major impact on log collection performance.

Tuning the events to be collected and processed is another way to improve EPS numbers. This guide will provide details on these factors to help you configure and optimize your LCE deployment for high availability as well as scalability.

Optimizing LCE Performance

Multiple LCE servers can take advantage of a “smart” load balancing facility to further maximize throughput and reliability. A primary LCE server monitors workloads across multiple auxiliary servers, and automatically routes data for processing to the system with the lightest processing load. Aggressive queuing algorithms are used to take advantage of newly added LCE servers’ full processing capacity.

The following diagram shows a sample architecture using one primary and two auxiliary LCE Servers, managed by SecurityCenter.



There are multiple ways to configure LCE in a high-performance environment, which are described in the Tuning for Performance section of this document.

Hardware Recommendations

Storage

LCE has the ability to perform fast network and storage I/O operations, along with high compression ratios - 20:1 in most instances.

Data is collected and stored in an extremely high-speed proprietary storage mechanism known as “silos”. In the LCE configuration file, the number and size of silos define the total amount of storage space for event data. For example, an LCE instance could be configured to use 200 (Maximum number of silo’s is 1024) silos, each 4 GB (Maximum silo size is 10 GB) in size for a total of 800 GB of storage space. Each silo also has an index that allows for high-speed analysis of

the events stored by the LCE. As more and more events arrive, they are written to the current LCE silo. When the last silo is full, the first silo is either archived or deleted, and new data is added.

There are many factors to consider when choosing a storage solution. While onboard storage is recommended, a storage area network (SAN) using iSCSI or Fibre Channel can also provide the required speed and capacity necessary for a high performance installation. Network-attached storage (NAS), such as NFS or CIFS file servers, are generally not recommended based on performance-hindering shared access and network bandwidth limitations.

Hardware characteristics are also a consideration. Hard drive I/O performance is affected by the rotational speed (on mechanical drives), which directly impacts the read-write latency for data access. Most enterprise drives are either 10,000 or 15,000 RPM. 15,000 RPM is the best choice for I/O intensive tasks such as log collection and analysis.

Host interface technologies also play a major role in performance. Getting data to and from the drive is often the biggest bottleneck. Most servers utilize the latest SAS or SCSI protocols on a high-speed interface, hitting speeds of 6 Gb/s with SAS drives.

Solid-state drives (SSD) are starting to make a major footprint in enterprise storage. While almost every performance point is better with SSD, there are two downsides. First, SSD demand is higher than supply and pricing is, at this point, relatively high. The second, which is most important for EPS statistics, is write performance. While SSDs offer incredible read speeds, mechanical drives still perform better when writing large amounts of small data. This correlates directly with the need to write a lot of individual events at high speed. The plus side, of course, is retrieving and working with the data. This should be taken into consideration when designing your LCE architecture.

RAID is another technology to consider with such an arrangement. The absolute fastest method is RAID 0 (striped), and is recommended for high EPS systems. The downside to this RAID level is that there is no redundancy of data. Please refer to the section LCE High Availability Configuration for methods to protect LCE data.

High EPS will come with a need for large storage capacity. At a constant rate of 30,000 EPS, a full 24 hours of events collected (assuming no plugin tuning) would result in 2,592,000,000 events. At an average size of 200 bytes per event, this comes out to almost 500 gigabytes of uncompressed data per day.

Size can vary widely based on the type of events being collected. It is important to take into account the number of relevant events that are expected to be received since, by default, LCE will only save those events that match the criteria (plugins) you have configured. The ability to save all events is available, but disabled by default and will significantly impact the amount of disk space required.

Factoring in a top compression ratio of 20:1, as well as other data needed for normalization and analysis, it would be recommended to allocate at least 2 TB of storage given the numbers previously stated.

Memory, CPU, and Storage

LCE servers can use large amounts of memory and CPU cycles to perform high-speed analysis and reporting. Storage for these events will also require hard drive space. The following table lists general recommendations based on the number of events per second (EPS) processed per system. This also assumes a storage configuration of SAS 3 Gb/s, RAID 0 striping. With load balancing, the specifications can be slightly less for each system part of the implementation.

Events/sec	CPU	Memory	Disk*
< 3000	2 cores @ 3GhZ	4 GB	1.5x Licensed Storage Size
3000 - 5000	4 cores @ 3GhZ	8 GB	1.5x Licensed Storage Size
5000 - 20000	8 cores @ 3GhZ	16 GB	1.5x Licensed Storage Size
20000+	16 cores @ 3GhZ	32 GB	1.5x Licensed Storage Size

More CPU cycles will noticeably increase the performance of LCE, with more memory making the largest impact on search queries.

It should also be noted that LCE will automatically detect the number of CPU cores on the host system, and allocate threads in the most efficient manner possible. The **`log-processors`** option in **`lce.conf`** sets the upper-limit, as detailed later in this document.

Network

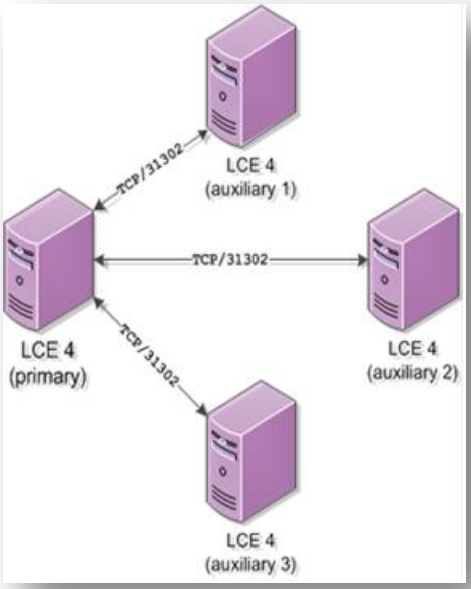
The primary LCE instance must have enough network bandwidth to receive data from each of its sources without affecting normal network operations.

This means if you have a 1 Gb/s network interface card (NIC), and you are saturating the link with events to the server, you will probably lose some events and will have latency issues when accessing the data from LCE Manager or SecurityCenter. Multiple NICs can help, and by default, LCE will listen on all IP addresses assigned to it.

Load Balancing

LCE 4 supports a primary/auxiliary configuration that intelligently balances workloads across multiple systems.

The following diagram shows the relationship between a primary instance of LCE, along with three auxiliary LCE systems for load balancing event processing.



The **`lce.conf`** file, located in the **`/opt/lce/daemons`** directory, maintains the configuration options necessary to implement this type of setup.

The basic implementation of load balancing with LCE requires a “primary” installation, along with one or many “auxiliary” installations configured to communicate with the primary. Nothing needs to be done for the primary LCE to listen for auxiliary connections. However, all auxiliary LCE systems need to be configured to communicate with the primary.

The following **`lce.conf`** entry tells the auxiliary LCE that it will be connecting to a primary:

```
primary-lce-address 192.168.1.20:31302
```

To enable this configuration option, uncomment this line and change the IP:port to point to the primary LCE. If you wish to use different port numbers, edit both the primary and auxiliary configurations.

LCE Manager or SecurityCenter must have every LCE system added in order to query all events. This is because each LCE instance stores only the events it processes, and LCE Manager and SecurityCenter need visibility into all of them to properly display and search all events across all LCE systems.

For extremely saturated systems, you can also have the primary system multi-homed, with one NIC receiving only event logs, and the second NIC for offloading work to the auxiliary LCEs. A third NIC could be added for management only (LCE Manager or SecurityCenter).

Once all systems are ready, each LCE instance (both primary and auxiliaries) must be added as resources within SecurityCenter or LCE Manager. This allows access to all data collected, with the added benefit of query load balancing.

Running primary and auxiliary LCE instances in a virtual environment on the same hypervisor is not recommended.

Tuning for Performance

Plugins

One of the most demanding features of LCE is the plugin libraries, and disabling unused plugins is a quick way to increase processing performance.

Since LCE supports logs from a vast array of products, there may be many libraries enabled for software and appliances that are not present on your network. Disabling the plugins for logs that will never be collected reduces the amount of work LCE has to perform, thereby allowing it to complete its normalization and correlation at a faster speed.

PRM files can be disabled by listing them in the **`/opt/lce/admin/disabled-prms.txt`** file. LCE 4 also includes a shell script that automates the process of determining which libraries are unneeded. This is accomplished by examining your LCE's database to determine which events have been encountered since the system was installed. These events are cross-referenced against the installed plugins to generate a list of plugin files for which no event has ever been encountered. The script reports on the number of such files, and provides the option of automatically adding them for disablement.

The **`plugin_manager.sh`** script is located in the **`/opt/lce/tools`** directory. When run, it will report on the number of installed plugin libraries that have never been used, and prompt you to disable the associated files. You may choose not to do so if you wish to review a full report prior to making any changes. In this case, the script will list the unused files.

An example of the plugin_manager.sh script output:

```
# sh plugin_manager.sh

171 plugin libraries have never been used. Would you like to disable these? [y/n] y

171 additional files were added to /opt/lce/admin/disabled-prms.txt. Please restart
      LCE for the changes to take effect.
```

The changes will take place when the LCE server restarts, or the next time a silo rolls if the **`auto-update-prms`** configuration option is enabled in **`lce.conf`**.

To observe the resulting increase in performance, you can view the LCE-Server_Statistics event, which can be found from drilling down into the "lce" type in LCE Manager or SecurityCenter. These events are generated by LCE once per hour by default, and report on a variety of performance statistics.

In general, it is recommended to wait about two weeks after the initial deployment of an LCE before disabling its unused plugin libraries. This will allow adequate time for all devices on your network to generate logs such that the LCE becomes aware of their presence.

When a large number of plugin libraries are disabled, it is a good idea to periodically check the **/opt/lce/db/notmatched.txt** file. Here, you should look for logs from any new device types that may have been added to the network since the plugins were disabled.

Since Tenable's research team adds support for new devices regularly, you may want to periodically re-run the script to disable any new libraries that are unneeded. The script will add only unused libraries that are not already present in the **disabled-prms.txt** file, so there are no extra steps necessary to avoid duplicating entries.

Rules

Another area of focus for fine tuning event normalization and storage is the **rules.conf** file, located in the **/opt/lce/daemons** directory.

On occasion, there may be a system for which you need only certain types of events collected. When dealing with plugins, you are enabling or disabling log collection for the entire device or application, such as Cisco VPN or Linux system logs. The **rules.conf** file allows a more granular approach, whereby you can add or remove specific types of events per device or application. Other criteria, such as IP addresses and named users, can also be added.

For example, you may not want to collect DHCP request events, with your focus being mainly on DHCP offers and other data collected. Here is the entry you would want to add to the **rules.conf** file:

```
+Types: dhcp
+Events: DHCP-Request
Ignore
```

There are quite a number of other tuning options that can be customized in this file. Please refer to the comments and to the LCE User Guide on the Tenable Support Portal. Filtering out unnecessary logs is probably the best way to process more events with LCE.

Excluding TASL Files

TASLs may be disabled selectively by adding the TASL script file name (e.g., **program_accounting.tasl**) to the **/opt/lce/admin/disabled-tasls.txt** file and then restarting the LCE daemon. This is useful for cases where a particular TASL script is not needed by an organization or where the TASL might be causing performance issues and needs to be disabled either temporarily or permanently.

Any disabled TASLs can be re-enabled by removing them from the **disabled-tasls.txt** file and restarting the LCE daemon.

Tuning lce.conf Parameters

There are a number of tunable parameters in the **lce.conf** file that directly impact the performance of LCE. Here are some sections to review, with recommendations for each. Keep in mind that when using multiple LCEs (as with load balancing primary and auxiliary systems) these settings should be based on the individual system. For example, it is recommended that an 8 CPU / 64 GB RAM primary have higher settings than a 2 CPU / 8 GB RAM auxiliary.

```
# The Log Correlation Engine is capable of leveraging multiple
# processor cores in order to process a number of incoming logs
# in parallel, yielding higher throughput. The following setting
# limits how many threads will be dedicated to log processing.
# For maximum perfomance, this should generally be set to or above the total
# number of CPU cores available on your system. For systems utilizing
# hyper-threading technology, the value can be scaled accordingly.
log-processors 8
```

This option leverages multicore processors and determines how many threads will be dedicated to log processing.

General guidelines are to specify a setting equal to the number of CPU cores in the LCE host system. This is an upper-limit, and should not be changed unless you have greater than 8 total cores (for example, a dual quad-core CPU system).

For systems with hyper-threading technology, the value may be scaled accordingly.

```
multiple-matches
```

This value can be used in deployments with custom plugin files that match logs that are already matched by factory plugins. Normally, if LCE encounters a matching plugin, the log will be normalized by that plugin and the match algorithm will stop. This configures LCE to continue evaluating plugins even after the first match is identified for each log. This can be detrimental to performance, because the matching algorithm must evaluate all plugins for each log received. Disabling the factory plugins that would match a custom log first may be a better alternative to using this value and evaluating all plugins for each log. See the section on “Plugins” above for details on disabling a particular factory plugin file.

```
# save-nonmatched      20000
# save-all             /opt/lce/db/lce.log
```

Uncommenting these values, or the explicitly marked debug values, may be detrimental to performance because the system will be forced to make a number of additional writes to disk. These values are intended for debugging purposes only.

```
# In order to maximize performance on multi-processor and multi-core systems,
# correlated TASL events are processed in parallel to the receiving of
# regular incoming events. Since some tasl scripts can run for an extended
# period of time, the primary event processor can potentially receive many
# tasl-triggering events while a tasl script is still being executed. In
# this case, the tasl job is stored in a queue for later processing. The
# following option defines the maximum size of this queue. On systems with
# extremely large volumes of data, setting the maximum queue size higher
# results in increased performance. If a tasl script designated as being
# sampleable is triggered while the queue is full, its callback functions
# will not be executed.

max-tasl-queue-memory 25M
```

This default of 25M is acceptable for most systems. Raising this limit will give systems that experience bursts of traffic (for example, batched syslog events received on an hourly basis) the ability to queue more work for processing when traffic slows down.

```
# The following file contains the filenames of TASL scripts that are
# designated as being sampleable, which results in the behavior
# described above (with max-tasl-queue-memory).
sampleable-tasls-file /opt/lce/admin/sampleable_tasls.txt
```

This file lists TASL scripts that are considered “sampleable”. Sampleable scripts will be skipped when the job queue is full. This can help temporarily alleviate load and recover from bursts of traffic faster; however, some events will not be evaluated by TASL scripts if this occurs.

```
# The lce_queryd process is responsible for high-performance processing of
# all query requests. By default, up to about 1 gigabyte of memory is used.
# For systems with large amounts of available memory, the following option
# can be used to allocate additional memory for the query daemon. This will
# increase the number of query results that can be cached, improving response
```

```
# time during event analysis in SecurityCenter. The option can be specified in
# kilobytes, megabytes, or gigabytes by appending a 'K', 'M', or 'G' to the value.
# Note that this value should never be set above 1500M on 32-bit systems
#
additional-query-memory 1G
```

Uncomment this parameter and increase the value by 1 GB for every 8 GB of RAM on your system. For example, a system with 32 GB of memory should have this set at 4 GB.

LCE High Availability Configuration

The goal of the LCE HA Configuration is to provide the capability for customers to deploy a real-time, fully functional backup on a separate physical host for a critical LCE deployment. The backup LCE shall have the following characteristics:

- Maintain replication of the primary database contents to within 3 seconds
- Provide a fully indexed database, instantly searchable at full speed
- In the event of the hard failure of the primary:
  - Assume a virtual server IP address within 3 seconds
  - Alert the user with a database event
  - Allow data sources to reconnect and resume sending events in the same state
- In the event of the recovery of the primary:
  - Provide a tool to re-synchronize the primary database to that of the backup while minimizing recovery time

Terminology

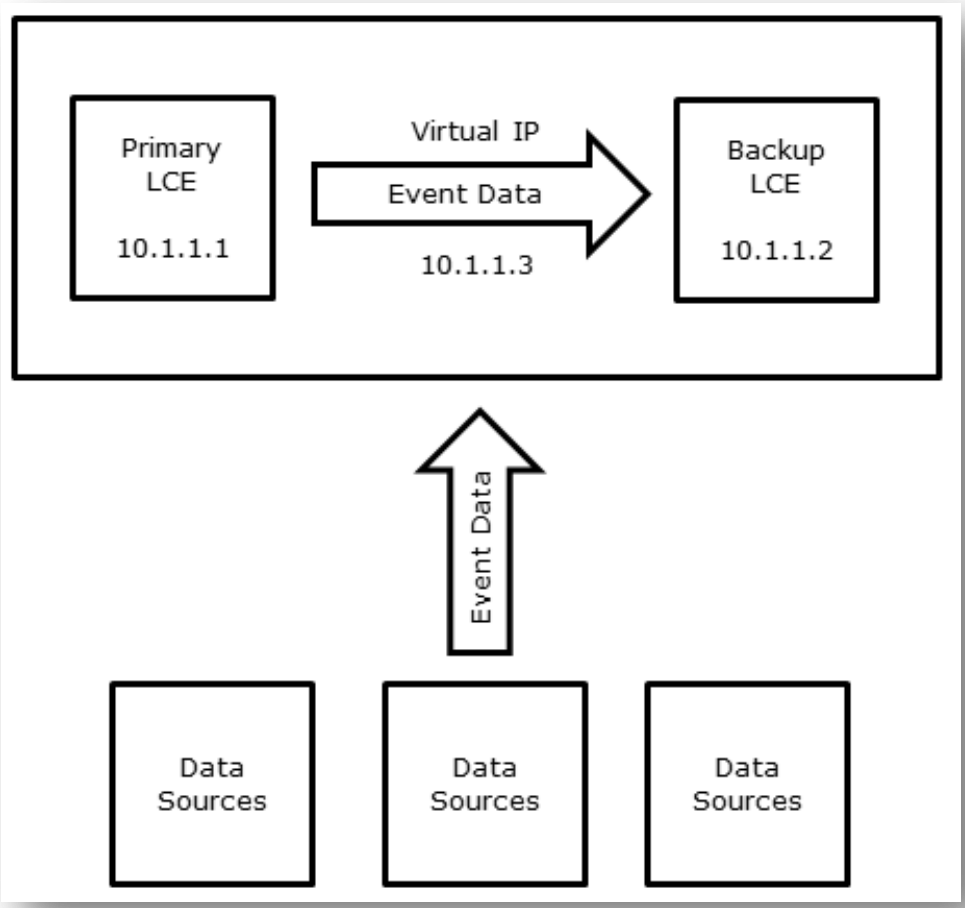
The following terms are used to describe the configuration:

Primary LCE: This is the main LCE deployment. It may be a fresh installation, or it may already contain data and a customized configuration. The Primary LCE may already have data sources already assigned to it (IDS devices, syslog devices, LCE clients, etc.). For this example, it will have a real IP address of 10.1.1.1.

Backup LCE (or Mirror LCE): This is a new LCE deployment intended only to act as a real-time backup of the Primary LCE. It will not have its own unique data sources; it will only receive data from the Primary LCE via the Load Balancing LCE interface. The Backup LCE is sometimes referred to as the “Mirror” because the database contents mirror those of the Primary LCE. For this example, it will have a real IP address of 10.1.1.2.

Virtual IP: This is a reserved IP address that doesn’t belong to any other host on your network. The Primary LCE will advertise that it owns this IP address, so data sources will be re-assigned to this virtual IP. In the event of a failure, the Backup LCE will advertise that it now owns this IP address, so that data sources will reconnect to the Backup LCE and data is not lost.

Data Sources: These are devices or programs sending data to LCE, whether switches, routers, firewalls, IDS devices, web servers, workstations, desktops, LCE clients, etc. Data sources may send data via syslog, reliable syslog, or the LCE Client API.



Pre-deployment Considerations

Deployment Requirements

- All LCE servers must be version 4.2.0 or higher, and have the same license size (1 TB, 5 TB, or 10 TB).
- We assume the Primary LCE is already installed, configured, collecting data, and can be searched via the SecurityCenter or LCE Manager. Primary LCE setup will not be covered in this document; please consult the Log Correlation Engine Administration and User Guide for more information.
- There must be an unused IP address on the network that can be allocated as a Virtual IP address for the LCE group.
- The Primary LCE and Backup LCE must exist on the same subnet.
- The user performing the installation and configuration must have root credentials on both the Primary LCE and Backup LCE.
- The user performing the installation and configuration must have Admin and Organizational Head / Manager credentials on the Organizations within SecurityCenter or LCE Manager to connect them to the Backup LCE and Virtual IP, and set up alerts.

Deployment Recommendations

- It is recommended that the Primary LCE and Backup LCE be on different physical hosts. The intent is to provide the best redundancy possible. If the Backup LCE is on the same physical host (virtualized, for example), it is likely that a hard failure disabling the Primary LCE will also disable the Backup LCE.

- It is recommended that the Primary LCE and Backup LCE have similar configurations to minimize deployment differences (for instance, either both or neither will update plugins – this will minimize processing differences).

To minimize processing differences as well as database recovery differences between the Primary LCE and Backup LCE, make sure that the underlying OS makes use of the Network Time Protocol (NTP) as described in:

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/sect-Date_and_Time_Configuration-Command_Line_Configuration-Network_Time_Protocol.html

Caveats

- Internal LCE events are not mirrored from the Primary LCE to the Backup LCE. These events are usually LCE-specific, such as processing statistics and License Expiration warnings. The Backup LCE will generate similar events for itself.

- TASL alerts are processed separately on each LCE. In order to maintain an instantly-ready, in-memory copy of the Primary LCE, the Backup LCE needs to feed its own TASL virtual machines to populate their internal state, much like the Primary LCE does.

Setup of the Primary and Backup LCE Servers

It is assumed that the Primary LCE is already installed and running on 10.1.1.1, and the Backup LCE is already installed on 10.1.1.2. There will be some setup involved on each host, but this example will start with setup Primary LCE.

It is imperative that the Primary LCE is completely set up before proceeding to the Backup LCE setup, otherwise the Backup LCE will falsely detect that the Primary LCE has failed because it is not yet correctly configured.

Primary LCE

1. Locate and uncomment, or add the following keywords and values to the file **`/opt/lce/daemons/lce.conf`**:

   `virtual-ip-address 10.1.1.3`

   This tells the Primary LCE the Virtual IP address that it will use.

   `virtual-ip-interface eth0`

   This defaults to `eth0` – it tells the Primary LCE which physical interface to use for the Virtual IP address.

   `lce-load-balance-local-addr 10.1.1.1`

   This is optional if there is only one IP address on the Primary LCE, but if there are multiple IP addresses, this option will tell the Primary LCE which one to use to communicate with the Backup LCE.

   `virtual-router-id 55`

   This is optional if you are not running the Virtual Router Redundancy Protocol (VRRP) elsewhere on your network. If you are running VRRP elsewhere, this option allows two instances of VRRP to interoperate. If this option is used, it must also match the value in the Backup LCE **`lce.conf`** file.

2. Run the following tool, giving it the IP address of the Backup LCE:

   `#` **`/opt/lce/tools/lce-setup-sync.sh 10.1.1.2`**

This will require root credentials on the remote LCE server. It performs a key exchange as user “lce” so that the Primary LCE may pull configuration items from the Backup LCE.

3. Restart the LCE services:

```
# service lce_server restart
```

4. Copy the following tool to the host running SecurityCenter or LCE Manager:

```
# scp /opt/lce/ha/add_virtual_ip_to_sc.sh root@10.1.1.5:~/
```

Backup LCE

1. Locate and uncomment, or add the following keywords and values, to the file **`/opt/lce/daemons/lce.conf`**:

   `primary-lce-address 10.1.1.1`

   This tells the Backup LCE the real IP address of the Primary LCE – it will send status to this address and receive logs from this address. If you have changed the **`lce-load-balance-local-addr`** or **`lce-load-balance-status-port`** on the Primary LCE, this setting will need to match those settings. The port is delimited from the IP with a colon (e.g., `10.1.1.1:31302`).

   `mirror-mode`

   This tells the Backup LCE to request copies of all data logs received on the Primary LCE, instead of normally receiving a subset of data logs in Load Balancing mode.

   `virtual-ip-address 10.1.1.3`

   This tells the Backup LCE the Virtual IP address that it will use if the Primary LCE experiences a failure.

   `virtual-ip-interface eth0`

   This defaults to eth0 – it tells the Backup LCE which physical interface to use for the Virtual IP address if the Primary LCE experiences a failure.

   `lce-load-balance-local-addr 10.1.1.2`

   This is optional if there is only one IP address on the Backup LCE, but if there are multiple IP addresses, this option will tell the Backup LCE which one to use to communicate with the Primary LCE.

   `virtual-router-id 55`

   This is optional if you are not running the Virtual Router Redundancy Protocol (VRRP) elsewhere on your network. If you are running VRRP elsewhere, this option allows two instances of VRRP to interoperate. If this option is used, it must also match the value in the Primary LCE **`lce.conf`** file.

2. Run the following tool, giving it the IP address of the Primary LCE:

```
# /opt/lce/tools/lce-setup-sync.sh 10.1.1.1
```

   This will require root credentials on the remote LCE server. It performs a key exchange as user “lce” so that the Backup LCE may pull configuration items from the Primary LCE.

3. [Optional] If the Primary LCE has been installed and collecting data for some time, you can optionally synchronize the database on the Backup LCE so that it is instantly synchronized with the event data from the Primary LCE. To do so, run the following tool and follow the prompts:

```
# /opt/lce/tools/lce-sync-to-remote-db.sh --init
```

4. Restart the LCE services if you did ***not*** perform the previous step to synchronize the databases (if you did, the tool will have already restarted this LCE):

   ```
   # service lce_server restart
   ```

Verification

There are a number of ways to verify that your setup is now working correctly. On the Primary LCE, verify that the Virtual IP is now listed with the "ip" utility. This indicates that the Primary LCE "owns" the Virtual IP.

```
[root@LCE_Primary ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:10:11:10:01:01 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.1/22 brd 10.1.1.255 scope global eth0
    inet 10.1.1.3/32 scope global eth0
```

The Primary LCE spawns another set of processes to maintain this Virtual IP. Verify that they are running as shown:

```
[root@LCE_Primary ~]# ps -C keepalived
  PID TTY          TIME CMD
21184 ?        00:00:00 keepalived
21186 ?        00:00:00 keepalived
21188 ?        00:00:00 keepalived
```

The Backup LCE should also show these processes, but ***not*** show the Virtual IP:

```
[root@LCE_Backup ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:20:22:20:02:22 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.2/32 scope global eth0
```

The Backup LCE will also start duplicating configuration items in a folder named after the Primary LCE IP address:

```
[root@LCE_Backup ~]# ls /opt/lce/ha/10.1.1.1/
lce.conf  pm.db  policies  syslog_sensors.txt
```

If you notice that both hosts have the Virtual IP, they may not be on the same subnet, and are unable to communicate with one another to signal that only one of them should own the Virtual IP.

If you notice that the **`keepalived`** processes are not running, you may have omitted the "**`mirror-mode`**" option, or you may have entered the "**`primary-lce-address`**" incorrectly in the Backup LCE configuration file.

Performing Server Maintenance

The Primary LCE and Backup LCE are now in constant communication, so stopping the Primary LCE service or host for any amount of time will trigger an immediate failover condition in which the Backup LCE assumes responsibility for all event processing and storage. To avoid false failovers, follow these guidelines to performing server maintenance.

Reconfiguring the Primary or Backup LCE Server

Reconfiguring the Primary or Backup LCE server can be done without stopping the **`lce`** or **`lce_server`** service. The **`lce_indexer`**, **`lce_query`**, **`lce_report_proxy`**, and **`stats`** services may be safely restarted individually without the risk of a false failover after completing these steps.

1. Make the required changes to **/opt/lce/daemons/lce.conf**, or other related configuration files.
2. Run the following utility to notify the LCE server daemon:

   # **/opt/lce/tools/lce-reload-conf.sh**

3. Restart ancillary LCE services that corresponds to the change in the **lce.conf** file:

```
If an item in the list below was edited in the lce.conf file a restart of the
       lce_indexer service would be required:
database-directory
# service lce_indexer restart

If an item in the list below was edited in the lce.conf file a restart of the
       lce_query daemon would be required:
additional-query-memory
database-directory
location
server-address
You can restart the daemon by doing the following:
# service lce_query restart
If an item in the list below was edited in the lce.conf file a restart of the
       lce_report_proxy daemon would be required:
log-directory
server-address
report-directory
reporter-user
reporter-password
reporter-port
ca_file
cert_file
key_file
client_dn
You can restart the daemon by doing the following:
# service lce_report_proxy restart
If an item in the list below was edited in the lce.conf file a restart of the stats
       daemon is required.
database-directory
plugins-directory
number-silos
report-directory
ids-database-directory
event-directory
log-directory
stddev-threshold
stddev-unit-threshold
nhits-frequency-threshold
iteration-threshold
syslog-alert
include-networks
exclude-networks
You can restart the daemon by doing the following:
# service stats restart
```

Restarting the Primary LCE Host

There are two recommended options to restart the Primary LCE host for other maintenance:

Option 1: Stop the Backup LCE Service

If the Backup LCE service is temporarily stopped, no failover will occur. This is the simplest option; however, no event processing will occur during the restart of the Primary LCE Host.

1. On the Backup LCE, stop the LCE service:

   ```
   # service lce stop
   ```

2. Restart the Primary LCE Host.

3. When the Primary LCE Host has restarted, and the Primary LCE Host owns the virtual IP (see the Verification section above for more details), restart the Backup LCE service:

   ```
   # service lce start
   ```

Option 2: Allow the Backup LCE to assume the Primary Role, then Re-synchronize Databases

If the Backup LCE is not stopped, it will assume the Primary LCE role until subsequently stopped. This will result in events being stored in the Backup LCE database that are not in the Primary LCE database. To recover, the same tool used after a failover will be employed after the restart is complete.

1. Restart the Primary LCE Host. The Backup LCE will assume data processing responsibilities.

2. Proceed to the "Recovery After Failover" section later in this document to re-synchronize the databases and ensure that the Backup LCE no longer has primary processing responsibilities.

Setup of Data Sources

Syslog Sources

To maintain redundancy in the event of a failure, devices sending data to the existing Primary LCE via syslog need to be updated to send their data to the Virtual IP address (i.e., 10.1.1.3 in lieu of 10.1.1.1). For example, if you are using Tenable's Passive Vulnerability Scanner, open **`/opt/pvs/etc/pvs.conf`** and add or modify the option to point to your Virtual IP address:

```
realtime-syslog 10.1.1.3;
```

LCE Clients

LCE Clients also need to be assigned to the Virtual IP address.

For LCE 3.x clients, you will need to modify their configuration file (a file in the same directory as the client ending in **`.conf`**) to tell them that the LCE server now lives at the Virtual IP address (10.1.1.3), for example:

```
[root@ LCE_Primary]# grep lce-server /opt/lce_client/lce_client.conf
/opt/lce_client/lce_client.conf:                lce-server 10.1.1.3 {
```

For LCE 4.x clients, you can use the LCE Client Manager to re-assign all clients to the new Virtual IP address. In this example, we first get a list of all client identifiers assigned to 10.1.1.1 (the Primary LCE), and re-assign them all in one command to 10.1.1.3 (the Virtual IP), port 31300. In this example, we have 7 clients numbered 1-7. The port assignment should match the "server-port" option in **`lce.conf`**. However, if you did not change it, the default is always 31300. The "…" will display a summary of all clients you successfully re-assigned, and the return value, if successful, should be 0.

```
[root@ LCE_Primary]# data=$(/opt/lce/daemons/lce_client_manager -D | grep "10.1.1.1" |
       awk '{ printf $5 "," }');
[root@ LCE_Primary]# echo $data
1,2,4,3,5,6,7,
```

```
[root@ LCE_Primary]# /opt/lce/daemons/lce_client_manager --assign-client --client-
      identifier $data --server-ip 10.1.1.3 --server-port 31300
…
[root@ LCE_Primary]# echo $?
0
```

Please read through the “Setup of SecurityCenter / LCE Manager” and “Setup of SecurityCenter / LCE Manager – Alternate Alert Method” before proceeding with your configuration. After you have reviewed the alternatives, choose one and proceed with your configuration.

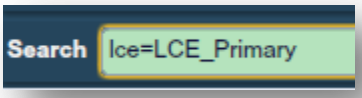
Setup of SecurityCenter / LCE Manager

It is assumed that the Primary LCE (10.1.1.1) is already configured and connected to the SecurityCenter (SC) or LCE Manager. With the addition of the Backup LCE and Virtual IP, we will now add the Backup LCE in order to set up an Alert to monitor for a failover condition. Please note that this addition will impact all existing queries that do not have an LCE filter. If this is undesirable, skip this section and see the next section (Setup on SecurityCenter / LCE Manager – Alternate Alert Method) to setup an alternate alert.

1. Log in to SecurityCenter or LCE Manager as the Admin user.
2. Click “Resources”, then “Log Correlation Engines”.
3. Click “Add” and populate the “Name” and “Description” of the Backup LCE. Enter the IP of the Backup LCE (i.e., 10.1.1.2), and select the same Organizations that the Primary LCE uses. Click “Submit”.
4. Your LCE list should look like this (your IP addresses may differ):

LCE_Primary	10.1.1.1	Working	4.2.0
LCE_Primary_Backup	10.1.1.2	Working	4.2.0

Once all LCEs are in “Working” status, keep in mind that “LCE_Primary_Backup” is simply a copy of “LCE Primary”, so you will see each event twice if you do not specify only to search “lce=LCE_Primary” in the search bar of your event queries:



This also allows users to manually load balance their queries if desired. “LCE_Primary” and “LCE_Primary_Backup” have the same data sets, so a user executing a full query on “LCE_Primary” may run perfectly parallel with another user executing a full query on “LCE_Primary_Backup”.

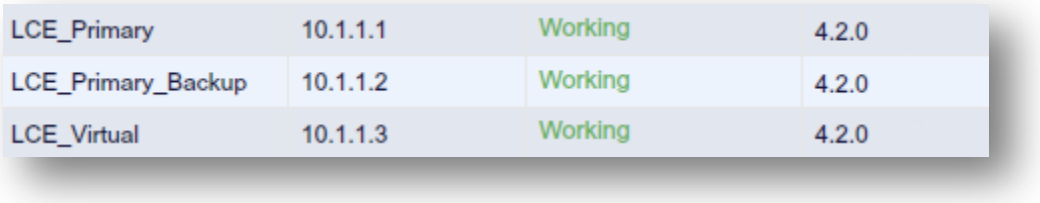
Optionally, the Virtual LCE may now be added. This allows the user to subsequently delete the LCE_Primary and LCE_Primary_Backup entries, so that no “lce=” query must be used. If a failover occurs, all user queries for LCE event data will also automatically resolve to the Backup LCE without additional configuration. To add the Virtual LCE:

1. Log in to the SecurityCenter or LCE Manager host via the terminal.

2. Invoke the tool previously copied from the Primary LCE installation, with the arguments <Virtual LCE IP> <Primary LCE IP> <Backup LCE IP>, in that order, for example:

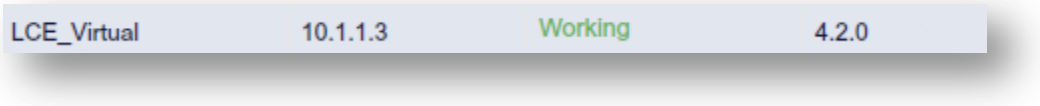
   ```
   # ./add_virtual_ip_to_sc.sh 10.1.1.3 10.1.1.1 10.1.1.2
   ```

3. Repeat steps (1) thru (4) in the section above using the Virtual LCE parameters (i.e., 10.1.1.3 and LCE_Virtual) in lieu of the Backup LCE parameters. Your LCE list should now look like this (your IP addresses may differ):



LCE_Primary	10.1.1.1	Working	4.2.0
LCE_Primary_Backup	10.1.1.2	Working	4.2.0
LCE_Virtual	10.1.1.3	Working	4.2.0

4. Click on “LCE_Primary”, then click “Delete”.
5. Click on “LCE_Backup”, then click “Delete”. Your LCE list should now look like the following screenshot, and no “lce=” filter is necessary for future queries because the Virtual LCE will automatically resolve to the LCE that owns the virtual IP address – the Primary or the Backup.



LCE_Virtual	10.1.1.3	Working	4.2.0

Next, to set up an Alert to be notified in the event of a failover, log out from the Admin user and log in as the Organizational Head or Manager.

1. Click “Workflow”, then “Alerts”.
2. Click “Add”, and populate the “Name” and “Description” as desired.
3. Set “Data Type” to “Event”.
4. Click “Click to Add Filters”:
   a. Under the “Event Filters” tab, set “Timeframe” to “Last 20 Minutes”.
   b. Under the “Event Filters” tab, set “Normalized Event” to = “LCE-Load_Balance_Primary_Failure”.
   c. Click “Apply Filters”.
5. Set “Trigger” to “Event Count >= 1”.
6. Set “Frequency” to “Every 15 Minutes”.
7. Set “Behavior” to “Perform Actions on Every Trigger”.
8. Add an appropriate action (e.g., Email the LCE Administrator and Security Officer, Assign a Ticket, etc.).
9. Click “Submit” to add the alert.

Timeframe and Frequency may be adjusted to lower values to avoid excessive queries to LCE. However, this will also increase the time it takes to be automatically alerted of the failure. It is recommended to have Timeframe as a slightly higher value than Frequency.

Setup of SecurityCenter / LCE Manager – Alternate Alert Method

If you skipped the previous section, it is recommended to complete this section. Choosing not to add the Backup LCE to SecurityCenter means that existing queries will not be impacted. However, if the Primary LCE fails, no events will be available and no Alert will be generated. The alternative is to add a different alert for the failure condition, and when it triggers, to add the Backup LCE to reinstate the ability to query the Backup database. The failure condition to watch for is when no events are received in a given timeframe. This is extraordinarily unlikely for most organizations, although you may need to adjust your Timeframe setting.

You must be logged into SecurityCenter or LCE Manager as an Organizational Head, or Manager to accomplish this.

1. Click “Workflow”, then “Alerts”.
2. Click “Add”, and populate the “Name” and “Description” as desired.
3. Set “Data Type” to “Event”.
4. Click “Click to Add Filters”:
    a. Under the “Event Filters” tab, set “Timeframe” to “Last 20 Minutes”.
    b. Under the “Advanced Filters” tab, set “LCEs” to “Primary_LCE”.
    c. Click “Apply Filters”
5. Set “Trigger” to “Event Count = 0”.
6. Set “Frequency” to “Every 15 Minutes”.
7. Set “Behavior” to “Perform Actions on Every Trigger”.
8. Add an appropriate action (e.g., Email the LCE Administrator and Security Officer, Assign a Ticket, etc.).
9. Click “Submit” to add the alert.

Timeframe and Frequency may be adjusted to lower values to avoid excessive queries to LCE and reduce the possibility of a false alarm. However, this will also increase the time it takes to be automatically alerted of the failure. It is recommended to have Timeframe as a slightly higher value than Frequency.

If the Alert triggers, it is recommended to immediately check the health and status of the Primary LCE, and in the event of an actual failure, to add the Backup LCE to SecurityCenter as described above and perform the recommended recovery.

Setup Complete

At this point, you have completed all necessary setup and your LCE data is now fully redundant across two physical hosts. The data on each host is fully indexed and instantly searchable at full speed. In the event of a failure on the Primary LCE, all clients and syslog devices will immediately be received by the Backup LCE. In the event of a failure on the Backup LCE, the Primary LCE will continue processing data normally.

Failover

What Happens when the Primary LCE Fails?

The LCE achieves high availability through the use of **`keepalived`**, a process that advertises a virtual IP address and communicates with another instance of **`keepalived`**, always ensuring that one of the two processes advertises the

virtual IP. In the event of a failure of the Primary LCE, the Backup LCE **keepalived** will begin advertising the Virtual IP address, and the address should even show up in the output by running the utility:

```
# /sbin/ip addr show
```

More importantly, the normalized event "LCE-Load_Balance_Primary_Failure" event (type: lce) is triggered on the Backup LCE to alert the user that it is taking over Primary LCE responsibilities. You should have set up an Alert specifically for this event during "Initial Setup – SecurityCenter / LCE Manager".

Clients and syslog sources will be briefly interrupted as the connections begin to switch over to the Backup LCE, then data continues to be processed normally, building on top of the data that had already been mirrored from the Primary LCE to the Backup LCE.

The Backup LCE will now begin to build up data that the Primary LCE does not have. At this point, it is imperative to restore the Primary LCE and proceed to the recovery step.

What Happens when the Backup LCE Fails?

There is no impact to LCE data if the Backup LCE fails.

Once the Backup LCE hardware is recovered you may go back to the "Setup – Backup LCE" section in this guide to re-synchronize the database on the Backup to that of the Primary (or re-install, re-setup, and then re-synchronize, if you are unable to recover the original file system).

Recovery after Failover

Running the Recovery Tool

The recovery tool can be run whether you were able to recover the Primary LCE file system or not. If recovery was successful, then you will have an out-of-date database on the Primary LCE, because the Backup LCE continued to process data while the Primary LCE was being restored. If you were unable to recover it, the tool can be used as a convenience mechanism to pull the database over in order to get back to a fully functional and redundant setup.

If the Primary LCE File System was not Recovered

1. Re-install the Primary LCE. The following important configuration items have been backed up on the Backup LCE in the following location, and can be retrieved to make your re-installation quicker (substitute your Primary LCE IP address in lieu of 10.1.1.1). Stop the LCE service on the Primary LCE before retrieving these from the Backup:

   /opt/lce/ha/10.1.1.1/

   lce.conf – place this into **/opt/lce/daemons/** on the Primary LCE

   pm.db – place this into **/opt/lce/daemons/** on the Primary LCE

   policies/ – place this entire folder into **/opt/lce/daemons/** on the Primary LCE

   syslog_sensors.txt – place this into **/opt/lce/admin/** on the Primary LCE

2. From the Primary LCE, re-run the following tool, giving it the address of the Backup LCE. This will require root credentials on the remote LCE server. It performs a key exchange as user "lce" so that the Primary LCE may pull configuration items and the database from the Backup LCE.

   ```
   # /opt/lce/tools/lce-setup-sync.sh 10.1.1.2
   ```

3. From the Primary LCE, run the recovery tool with the option **"--full**" to ensure that you pull all of the database, and follow the prompts carefully:

   ```
   # /opt/lce/tools/lce-sync-to-remote-db.sh --full
   ```

4. From the Backup LCE, re-run the following tool, giving it the address of the Primary LCE. This will require root credentials on the remote LCE server. It performs a key exchange as user “lce” so that the Backup LCE may resume pulling configuration items and the database from the Backup LCE.

   ```
   # /opt/lce/tools/lce-setup-sync.sh 10.1.1.1
   ```

5. Stop the **Backup** LCE:

   ```
   # service lce stop
   ```

6. Restart the **Primary** LCE:

   ```
   # service lce restart
   ```

7. Start the **Backup** LCE:

   ```
   # service lce start
   ```

If the Primary LCE File System was Recovered

From the Primary LCE, run the recovery tool and follow the prompts carefully:

```
# /opt/lce/tools/lce-sync-to-remote-db.sh
```

The tool is mostly automated, and will ask for verification before performing the primary synchronization. If any displayed information is incorrect, stop the tool immediately with “Control-C”. If the IP address of the Backup LCE was incorrect, remove all folders at the location **`/opt/lce/ha/`** on the Primary LCE that match IP addresses and create a single folder **`/opt/lce/ha/IP`**, where IP is the IP address of the Backup LCE, then rerun the tool. If the data list or approximate downtime is incorrect, rerun the tool with the argument “**`--full`**”.

When the primary synchronization is complete, you will be prompted for the remote LCE root password once to stop the Backup LCE, and then once again to restart it after the incremental synchronization is complete.

When the tool executes to completion, both LCE Servers will have been restarted in the original configuration. The Primary LCE will have processing responsibilities and own the Virtual IP, and the Backup LCE will once again serve as a backup.

For More Information

Tenable has produced a variety of additional documents detailing the LCE's deployment, configuration, user operation, and overall testing. These documents are listed here:

- Log Correlation Engine Architecture Guide – provides a high-level view of LCE architecture and supported platforms/environments.
- Log Correlation Engine Administrator and User Guide – describes installation, configuration, and operation of the LCE.
- Log Correlation Engine Quick Start Guide – provides basic instructions to quickly install and configure an LCE server. A more detailed description of configuration and management of an LCE server is provided in the "LCE Administration and User Guide" document.
- Log Correlation Engine Client Guide – how to configure, operate, and manage the various Linux, Unix, Windows, NetFlow, OPSEC, and other clients.
- LCE Best Practices – Learn how to best leverage the Log Correlation Engine in your enterprise.
- Tenable Event Correlation – outlines various methods of event correlation provided by Tenable products and describes the type of information leveraged by the correlation, and how this can be used to monitor security and compliance on enterprise networks.
- Tenable Products Plugin Families – provides a description and summary of the plugin families for Nessus, Log Correlation Engine, and the Passive Vulnerability Scanner.
- Log Correlation Engine Log Normalization Guide – explanation of the LCE's log parsing syntax with extensive examples of log parsing and manipulating the LCE's `.prm` libraries.
- TASL Reference Guide – explanation of the Tenable Application Scripting Language with extensive examples of a variety of correlation rules.
- Log Correlation Engine Statistics Daemon Guide – configuration, operation, and theory of the LCE's statistic daemon used to discover behavioral anomalies.
- Log Correlation Engine Large Disk Array Install Guide – configuration, operation, and theory for using the LCE in large disk array environments.
- Example Custom LCE Log Parsing - Minecraft Server Logs – describes how to create a custom log parser using Minecraft as an example.

Documentation is also available for Nessus, the Passive Vulnerability Scanner, and SecurityCenter through the Tenable Support Portal located at https://support.tenable.com/.

There are also some relevant postings at Tenable's blog located at http://blog.tenable.com/ and at the Tenable Discussion Forums located at https://discussions.nessus.org/community/lce.

For further information, please contact Tenable at support@tenable.com, sales@tenable.com, or visit our web site at http://www.tenable.com/.

About Tenable Network Security

Tenable Network Security is relied upon by more than 20,000 organizations, including the entire U.S. Department of Defense and many of the world's largest companies and governments, to stay ahead of emerging vulnerabilities, threats and compliance-related risks. Its Nessus and SecurityCenter solutions continue to set the standard to identify vulnerabilities, prevent attacks and comply with a multitude of regulatory requirements. For more information, please visit www.tenable.com.

GLOBAL HEADQUARTERS

Tenable Network Security
7021 Columbia Gateway Drive
Suite 500
Columbia, MD 21046
410.872.0555
www.tenable.com