Check Point
SOFTWARE TECHNOLOGIES LTD.

WE
SECURE
THE
FUTURE

Central Deployment Tool, Blink, and Zero Touch

Check Point Software

October 25, 2018

Technical Resources:

Tyler Roser, Security Engineer

Elie Klein, Security Engineer

EXECUTIVE SUMMARY

As the rate of consumption of applications grows exponentially in today's complex IT environment, traditional practices and upgrade methods need to evolve to the automation and orchestration levels of the dev-ops world. Cluster upgrades have been a tedious manual process until now. Check Point's **Central Deployment Tool (CDT)** is a utility that runs on Security Management Servers and Multi-Domain Security Management Servers running Gaia OS. This utility lets you manage deployment of software packages from your Management Server to multiple managed Security Gateways and cluster members at the same time. As soon as the command is executed, parallel instillations on multiple gateways and clusters is now possible even unattended. You can add some actions to be performed before and after the installations, monitor the progress, and you will get notifications by mail on any error and on completion. CDT allows for installation of software packages, take snapshots, run shell scripts, push/pull files, automate the RMA backup and restore process, and much more. CDT handles cluster upgrades automatically, including Connectivity Upgrade (CU).

Gaia Fast Deployment mechanism called "Blink" allows users fast and easy deployment of *cleanly installed* Check Point Security Gateways and Security Managements. Upon completion of the deployment process, user gets a cleanly installed machine (with completed First Time Configuration Wizard), desired Hotfixes, and updated signatures for Software Blade installed. Blink allows for deployment within ~5-7 minutes for Security Gateway and ~10-16 for Security Management of a cleanly installed Security Gateway or Security Management (Blink Image), including desired hotfix packages, and updated Software Blades signatures.

Zero Touch is a Check Point cloud deployment service. It enables the initial deployment of multiple Gaia gateways. The administrator applies the initial deployment configuration for the gateways in the Zero Touch portal. When a gateway is connected to the internet for the first time, it fetches the settings automatically. The settings from the Zero Touch server replace the First Time Configuration Wizard. The combination of CDT, Blink, and Zero Touch will transform and increase the speed of deployment for Check Point physical and virtual appliances.

INTRODUCTION

The time has come to update your Check Point machines – install the latest version, or the latest Jumbo hotfix, or even both. You have some available maintenance windows, in which you need to deploy the update to your gateways and clusters, but there are so many of them! Doing it manually one by one takes a lot of time, and can drag the update process for several weeks or even months.

CDT comes to the rescue. With CDT, you can install the update - upgrade and/or as many hotfixes as you would like - on all your gateways in parallel in one command. It also performs Connectivity Upgrade for your clusters. You can add some actions to be performed before and after the installations, monitor the progress, and you will get notifications by mail on any error and on completion.

CDT also introduces an exciting new capability – RMA backup & restore. You can use CDT to back up your gateway's configuration, saved as a small file on the management machine. When one of your gateways becomes damaged and needs to be replaced via RMA, you can use the backed-up information with CDT to restore the gateway to working capability – CDT will install the version and the hotfixes, which were installed on the gateway before, and will restore the configuration for you.

- **Basic Flow** –you can use the CDT to upgrade or install hotfixes on multiple gateways. Clusters upgrades are performed automatically and the management objects are upgraded automatically.
- **Advanced Flow** – you can now prepare a complete **deployment plan** that will be executed on all gateways and clusters by the CDT. The deployment plan is a set of actions such as: install a package, uninstall a package, download package from cloud, push/pull files, take snapshot, run script, etc. As with the basic flow, CDT automatically controls cluster upgrades, and upgrades the management objects as well.
- **RMA** – CDT now allows you to automate your RMA process. You can use the CDT to collect version and configuration information from all of your gateways, and use the CDT to automatically restore the GW on a new appliance after RMA. All you need to do is set the IP on the new appliance, and run CDT to restore the gateway.

Candidates List

The Candidates List lets you select the Security Gateways, on which to install the CPUSE packages. The Candidates List is a CSV (comma-separated values) file generated by CDT. This list contains the supported Security Gateways and Cluster Members in the Security Management

Server or Domain Management Server database. Note the number associated with the upgrade order column; 1 denotes eligible for upgrade/installation of hotfix, N/A articulates the gateway is not valid for this package.

```
Object Name , Cluster Name , IP Address ,   Version/FW build , Member State , Upgrade Order
=============================================================================================

 c01gw01 ,    cluster01 ,    172.23.1.112 ,   R75.46/102 ,       active ,           1
 c01gw02 ,    cluster01 ,    172.23.1.113 ,   R75.46/102 ,       standby ,          1

  gw-062 ,       N/A ,       172.23.1.62 ,    R77.30/101 ,         N/A ,        Installed
  gw-245 ,       N/A ,       172.23.1.245 ,   R77.10/243 ,         N/A ,          N/A
```

Example CDT Configuration File

These are basic examples of the primary configuration file CentralDeploymentTool.xml:

For the CDT Basic Mode:

```
<PackageToInstall   Path="/home/admin/CDT/Check_Point_geyser_T204_Install_and_Upgrade.tgz"
                    Type="MAJOR" Version="R77.30"
                    RequiresReboot="true"
                    ConnectivityUpgrade="true"/>

<Logging FileLevel="DEBUG" ScreenLevel="NORMAL" SyslogLevel="ERROR"/>

<CPUSE RPMPath="/home/admin/CDT/CPda-00-00.i386.rpm"/>

<PreInstallationScript Path="/home/admin/CDT/RemovePrivateJumbo.sh" IsBlocking="true" />

<PostInstallationScript Path="/home/admin/CDT/VerifyNetworkConfigurations.sh" IsBlocking="false" />

<MailNotification SendTo="admin@organization.com"/>
```

For the CDT Advanced Mode:

```
<?xml version="1.0" encoding="UTF-8"?>
 <CentralDeploymentTool>
  <Logging FileLevel="DEBUG" ScreenLevel="NORMAL" SyslogLevel="NONE"
Colors="false"/>
  <CPUSE RPMPath="/home/admin/CPda-00-00.i386.rpm"/>
  <Batch MaxMachinesCount="UNLIMITED" LatestAllowedDate="31/12/2099"
LatestAllowedTime="23:59"/>
  <MailNotification SendTo="aa@xyz.com"/>
 </CentralDeploymentTool>
```

For the CDT RMA Mode:

```
<?xml version="1.0" encoding="UTF-8"?>
 <CentralDeploymentTool>
  <Logging FileLevel="DEBUG" ScreenLevel="NORMAL" SyslogLevel="NONE" Colors="false"/>
  <CPUSE RPMPath="/home/admin/CPda-00-00.i386.rpm"/>
```



```
  <MailNotification SendTo="aa@xyz.com"/>
  <Repository path="/home/admin/"/>
 </CentralDeploymentTool>
```

Deployment Plan

In CDT Advanced Mode, you can define a sequence of actions for the remote Security Gateways. The supported actions include:

import_package – Sends a package to the remote Security Gateway (to the /var/log/upload/ directory) and imports it with CPUSE. If the package was already sent with the send_package, this only imports it on the remote Security Gateway.

send_package – Sends a package to the remote Security Gateway (to the /var/log/upload/ directory) without importing it with CPUSE.

install_package – Installs a package with CPUSE and validates that security policy is installed.

uninstall_cpuse_package – Uninstalls a package with CPUSE.

uninstall_legacy_package – Uninstalls a legacy package (a package that was installed with the Legacy Installation method in Expert mode CLI).

execute_command – Runs a command on the Security Gateway in Bash shell (Expert mode).
execute_script – Runs a user shell script on the Security Gateway.

pull_file – Downloads a file from the remote Security Gateway to the Management Server. Limitation: the size of the file must be less than 1 GB.

push_file – Uploads a file from the Management Server to the remote Security Gateway.

send_email – Send an email message.

Log – Generates a log message. The logging level of this message can be DEBUG, NORMAL, ERROR, ALWAYS.

Reboot – Reboots the remote Security Gateway.

download_from_cloud – download a package from the Check Point Cloud with CPUSE.

create_snapshot – Creates a Gaia snapshot.

This example Deployment Plan performs these actions on all applicable Security Gateways:

1. Backs up the file /opt/productname/conf.txt on the remote Security Gateway to the /opt/CPcdt/ConfigurationBackupFiles/ directory on the Management Server.
2. Sends a file /opt/CPcdt/conf.txt from the Management Server to the remote Security Gateway as the /opt/productname/conf.txt file.

Example XML file for this Deployment Plan:

```
<?xml version="1.0" encoding="UTF-8"?>
<CDT_Deployment_Plan>
 <plan_settings>
  <name value="Change configuration file" />
  <description value="Example deployment plan - replace a file" />
  <update_cpuse value="true" />
 </plan_settings>
```



```
 <!-- Backup the configuration file -->
 <pull_file remote_path="/opt/productname/conf.txt"
local_dir="/opt/CPcdt/ConfigurationBackupFiles/" />

 <!-- Push the new configuraion file -->
 <push_file local_path="/opt/CPcdt/conf.txt"
remote_path="/opt/productname/conf.txt" />
</CDT_Deployment_Plan>
```

RMA – Backup & Restore

The CDT RMA features allows you to back up your gateways information17, and when they need to be replaced via RMA, it allows you to easily recover the saved information and restore it on the replaced gateway. It saves and restores your gateway's version, installed hotfixes and configuration.

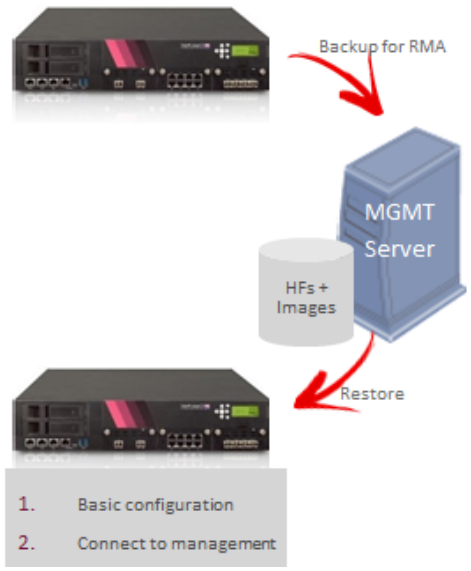
Backup connected gateways

- Version
- Installed Hotfixes
- OS configuration (SIC, licenses, basic files for policy installation process)

Restore a single (replaced) gateway from saved backup

- Connecting the gateway to the network
- Original networking settings (IP, Default GW, etc.)
- Relevant hotfixes and version image in the CDT repository

```
[Expert@gw-a3221d:0]# ./CentralDeploymentTool  -rma -backup -candidates=<file name> -server=<Domain Management Server IP>
```

Frequently Asked Questions

Q: What is the impact CDT has on CPU usage and traffic bandwidth of the Management Server?

A: The main bottleneck when using CDT for multiple Security Gateways in parallel is the delivery of the CPUSE Offline package to the Security Gateways.
CPU usage will be high on the Management Server during the first phases of Candidates List generation and validation, and during the delivery of CPUSE Offline package.
After delivering the CPUSE Offline package, the CDT will only monitor the process by querying the Security Gateways every few seconds - a process that does not consume a lot of traffic bandwidth.

Q: My maintenance window is short. How can CDT help me?

A: CDT can be executed in **Preparations mode**. In this mode, CDT delivers the CPUSE Agent RPM and CPUSE Offline package to the Security Gateways, without installing them. There will be no connectivity loss on the Security Gateways. If Preparations mode is executed *before* the maintenance window, the "installation" process will be much faster. Alternatively, CDT can be executed in **Extended preparations mode**. In this mode, in addition to everything that Preparations mode accomplishes, CDT will update the CPUSE Agent on the gateways, import and verify the CPUSE package without installing it. In addition you can use **Advance mode** for doing preparations / Extended preparations by creating a deployment plan without any installation actions.

Q: Is it possible to copy the package to the gateway manually to shorten CDT installation time?

A: Yes, It is possible to manually copy the package to the remote Security Gateway to the **/var/log/upload/** directory. When executing the installation via CDT, CDT will detect that the package is already on the remote Security Gateway, will verify the package MD5 signature, will skip the sending stage and will continue to importing and installing the package. This can be useful if the bandwidth between the Management Server and the remote Security Gateways is limited, and sending files via CDT installation/preparations is slower than placing the package file on the remote Security Gateway by pulling them from an FTP server or another location.

Please refer to the CDT SK article for more FAQs and troubleshooting questions.

References

Central Deployment Tool (CDT) v1.5.2 Administration Guide

SK111158 Central Deployment Tool

Blink

"…Ever been working on an issue and after an hour say to yourself, I can likely fix this issue by just reimaging the box rather than continuing to troubleshoot." While this sounds great, you also realize that means, needing to head onsite, format a USB with the proper image and patches, reformat the gateway, and then reconfigure it with its previous settings, re-establish SIC and push policy. At this point, you are looking at a couple hours downtime for this. What if there was another way? Welcome to Blink – Gaia Fast Deployment mechanism that allows users fast and easy deployment of cleanly installed Check Point Security Gateways and Security Managements. Blink allows for the deployment of gateways in 5-7 minutes and security management in 10-16 minutes and the result is a cleanly installed image with desired hotfix packages, and updated Software Blades signatures.

This white paper will go through the requirements and document examples of how to configure and deploy Blink. Readers will have a detailed understanding of how to best utilize the Blink mechanism once finished. The examples provided in the following text are an example of a reimage of a gateway.

Requirements:

Versions:	Gaia OS R77.30 Gaia OS R80.10
Product:	Security Gateway Security Management
Hardware:	**For Check Point Gateways:** 2000, 3000, 4000, 5000, 12000, 13000, 15000, 21000, and 23000 **For Security Management:** Smart-1 25B, Smart-1 150, Smart-1 205, Smart-1 225, Smart-1 3050, Smart-1 3150, Smart-1 410, Smart-1 425, Smart-1 525, Smart-1 5150 Running Blink on software RAID appliances is supported on Blink images created in 24 Jan 2018 and later

Before beginning, it is important to discuss a few ways Blink can be utilized. There are two forms of installation that can be leveraged: basic and advanced. Basic configuration allows for base functionality of the appliance: i.e. IP address on Mgmt interface, SIC configuration and Gaia OS admin password, administrator credentials, approval to upload/download data from Check Point Cloud, Image updates. Note; that image updates piece is optional and is not required.

Files Needed:

- Blink Utility
- Image
- Latest Blink Updates [optional]

Utilizing the Blink Utility:

- Create a new directory on the appliance you are installing Blink on
  [Expert@HostName:0]# mkdir -v /var/log/MyBlink
- Transfer the three files to this directory
- Go to the new created directory
  [Expert@HostName:0]# cd /var/log/MyBlink
- Unpack the Blink utility package
  [Expert@HostName:0]# tar -zxvf blink.tgz

[Expert@HostName:0]# ls –lha

```
total 2.2G
drwx------ 6 admin root 4.0K Sep 17 04:48 .
drwx------ 4 admin root 4.0K Sep 17 04:48 ..
-rwxr-xr-x 1   105   80 2.2M Sep 17 03:00 BlinkInstaller
-rw-r--r-- 1   105   80  758 Sep 17 03:00 BlinkInstaller.config
-rw-r--r-- 1   105   80  512 Sep 17 03:00 BlinkInstaller.sha256
-rwxr-xr-x 1   105   80 2.2G Sep 17 03:00 CheckPoint_Gaia_fd.tgz
-rw-r--r-- 1   105   80  512 Sep 17 03:00 CheckPoint_Gaia_fd.tgz.sha256
drwxr-xr-x 2   105   80 4.0K Sep 17 03:00 blades_updates
drwxr-xr-x 2   105   80 4.0K Sep 17 03:00 installation_logic
-rw-r--r-- 1   105   80 1.3K Sep 17 03:00 manifest.xml
-rw-r--r-- 1   105   80  512 Sep 17 03:00 manifest.xml.sha256
drwxr-xr-x 2   105   80 4.0K Sep 17 03:00 user_updates
```

- Assign the execute permission to the Blink utility:

[Expert@HostName:0]# chmod -v +x blink

- Execute the Blink utility by running the desired basic flow:
  - [Expert@HostName:0]# ./blink –i /var/log/MyBlink/blink_image_1.0_Check_Point_R80.10_T462_Jumbo_T103_Gateway.tgz –x –a /var/log/MyBlink/answers.xml –d /var/log/MyBlink

- [-x] Will extract the blink image [to the same folder] without running it, allowing you to configure the blink utility.

- Configuring the answers.xml file

```
<properties xmlVersion="1.1">
        <installation>
                <reboot_delay>10</reboot_delay>
        </installation>
        <machine_configuration>
                <perform>false</perform>
                <hostname>blinky</hostname>
                <password_hash>PASSWORD_HASH_FIELD</password_hash>
                <network>
                        <ipv4addr>1.2.3.4</ipv4addr>
                        <masklength>24</masklength>
                        <interface>Mgmt</interface>
                        <default_gw>1.2.3.1</default_gw>
                </network>
                <role_configuration>
                        <gateway>
                                <!--  activation_key must be in base64 encoding -->

        <activation_key>SIC_BASED64_FIELD</activation_key>
                                <cluster>false</cluster>
                        </gateway>
        <send_data_to_usercenter>true</send_data_to_usercenter>
        <enable_download_from_checkpoint>true</enable_download_from_checkpoin
t>
        </machine_configuration>
        <user_updates>
                <entry_point>install_content.sh</entry_point>
        </user_updates>
        <!--
        logging - Used in order to filter the logs saved to files, displayed on the
screen or sent to the syslog.
        Supported logging levels: DEBUG, NORMAL, ERROR, ALWAYS, NEVER
        Colors - Should be set to true for displaying log messages in color on the
screen.
        -->
```



```
        <logging>
            <file_level>DEBUG</file_level>
            <screen_level>NORMAL</screen_level>
            <sys_log_level>NEVER</sys_log_level>
            <colors>true</colors>
        </logging>
    </properties>
```

A quick and easy way to find the base64 of your gateways SIC password is to perform the following from CLI:

[Expert@HostName:0]# echo –n <input> | base64

To run the Blink utility; run the same command you did prior to editing the xml file but take out the "-x".

- [Expert@HostName:0]# ./blink –i /var/log/MyBlink/blink_image_1.0_Check_Point_R80.10_T462_Jumbo_T103_Gateway.tgz –x –a /var/log/MyBlink/answers.xml –d /var/log/MyBlink --reimage

Argument	Description
-i <path to Blink Image>	Specifies the path to the *Blink image*. If this path is not specified explicitly, then Blink will search in the current working directory for a file with prefix *blink_image*.
-b <path to Blink Image updates package>	Specifies the path to the *Blink Image updates* package (*blink_updates_<OSVERSION>.tgz*). If this path is not specified explicitly, then Blink will search in the current working directory for a file *blades_updates_<OSVERSION>.tgz*.
-u <path to user TGZ file>	Specifies the path to the user TGZ file that contains user shell scripts and binary files that should be executed and installed during the main installation process. If this path is not specified explicitly, then Blink will search in the current working directory for a file *blink_custom_content.tgz*. Note: The package *blink_custom_content.tgz* must contain the main shell script as specified in the *answers.xml* configuration file (by default, Blink will search for the script *install_content.sh* - refer to section "(7-A) How to configure the Blink mechanism - The *answers.xml* file").
-a <path to answer.xml file>	Specifies the path to the user's configuration file for unattended installation (if needed). Refer to section "(7-A) How to configure the Blink mechanism - The *answers.xml* file".
-d <output directory>	Specifies the output directory, into which the Blink image and all the other packages should be extracted. If this path is not specified explicitly, then the Blink image and all the other packages will be extracted into the */var/log/blink/launcher/files* directory.
-x	Specifies that Blink image should be *only* extracted, skipping the installation. This option is for advanced users that wish to configure an unattended installation - refer to ***Step 9*** below.
--reimage	Using this flag will allow installation on machines that are already configured (performed First Time Wizard). By default, a snapshot of the old partition is saved, unless *--delete-old-partition* flag is supplied.
--delete-old-partition	Removes the old partition. Does not override the *--keep-old-partition* flag.
--keep-old-partition	A snapshot of the old partition is saved if this flag is on,

The above steps describe how to deploy a basic Blink deployment to a gateway that deploys a base OS and establishes SIC with the management server. The following is an example of an advanced deployment that is essentially a reimage of an existing gateway. In this scenario, Blink deploys the base OS, installs a JHF, performs an entire clish configuration and deletes the old partition.

```
[Expert@HostName:0]# ./blink -i
/var/log/MyBlink/blink_image_1.0_Check_Point_R80.10_T462_Jumbo_T103_Gateway.t
gz –x –a /var/log/MyBlink/answers.xml –d /var/log/MyBlink --reimage
```

[-x] Will extract the blink image [to the same folder] without running it, allowing you to configure the blink utility.

After extracting the Blink utility, you will find a directory named “installation_logic”. Within this directory, you will need to create two files. The first is: install_content.txt, the second is called clish_commands.txt.

install_content.sh:

```
#!/bin/bash
Log_File="/var/log/user_main_script.log"
echo "Configuring Mgmt interface..." >> $Log_File
clish -i -s -f "clish_commands.txt" >> $Log_File
```

clish_content.txt: [context of this file will be any additionaly clish configurations you would like to add, ie. Additional interfaces, dns, ntp etc.]

```
lock database override
set interface eth1 ipv4-address 9.8.7.6 masf-length 24
set dns tertiary 8.8.8.8
save config
```

Section	XML element	Description
<directories>	<working_directory>	Specifies the main working directory of the main executable file *BlinkInstaller*.
	<user_scripts_dir>	Specifies the directory (inside the main the working directory) that contains pre-stages / post-stages shell script(s) that should be executed during the installation: • scripts that should be executed *before* the specified stage should be called *pre_stage_<STAGE_NAME>.sh* • scripts that should be executed *after* the specified stage should be called *post_stage_<STAGE_NAME>.sh* where <*STAGE_NAME*> is the name of the relevant installation stage: • *blades_configurations* • *create_partition* • *extract_fcd* • *fdwizard_gateway* • *finalize_action* • *merge_var_log* • *run_post_script* • *update_blades* • *user_updates*
	<user_updates_dir>	Specifies the *directory* (inside the main the working directory), to which the *user custom shell scripts and binary files* will be copied during the

		installation. The default directory is: *user_updates*
	<blades_updates_dir>	Specifies the *directory* (inside the main the working directory), to which the *Blink Image updates* package (*blades_updates_<OSVERSION>.tgz*) will be copied during the installation. This package will be installed after the reboot. The default directory is: *blades_updates*
<status>	<status_file_name>	Specifies the name of the *status monitoring file* (inside the main the working directory). This file is used to monitor the *BlinkInstaller* process. This file contains the last stages performed by *BlinkInstaller*. The default file is: *status.txt*
	<status_brief_file_name>	Specifies the name of the *brief status monitoring file* (inside the main the working directory). This file is used to monitor the *BlinkInstaller* process. This file contains a brief description about the last stage performed by *BlinkInstaller*. The default file is: *status_brief.txt*
	<status_object>	Specifies the name of the *status object file* (inside the main the working directory). This is an XML file that stores the last status information object. The default file is: *status_obj.xml*
<collector>	<file_path>	Specifies the path and the name of the *CPdiag collector file* (inside the main the working directory). The file will be used by the CPdiag utility to collect statistics about the installation. The default is: */var/log/fd_collector.xml*
<configuration_files>	<manifest_file>	Specifies the name of the package manifest file that represents the structure of the Blink package. The default file is: *manifest.xml*
	<manifest_file_sha256>	Specifies the name of the Check Point Signature file for the package manifest file (used for integrity validation). The default file is: *manifest.xml.sha256*

Monitoring the Blink Installation:

[Expert@HostName:0]# ./BlinkInstaller -status <json | full | id>

where:

Option	Description

./BlinkInstaller -status json	Returns the last recorded status in JSON format.
	
	Example:
	
	`[Expert@HostName:0]# ./BlinkInstaller -status json`
	`{`
	`"isCompleted" : "true",`
	`"stageEndTime" : "5:0:4",`
	`"stageID" : "finish_message",`
	`"stageName" : "BlinkInstaller Installation",`
	`"stageStartTime" : "4:56:39",`
	`"state" : "Success",`
	`"statusDescription" : "The installation has finished successfully and is pending reboot!"`
	`}`

./BlinkInstaller -status full	Returns the last recorded status in a single-string representation.
	
	Example:
	
	`[Expert@HostName:0]# ./BlinkInstaller -status full`
	`BlinkInstaller Installation - The installation has finished successfully and is pending reboot! - S`
	`uccess [Started at: 4:56:39] [Ended at:5:0:4]`

./BlinkInstaller -status id	Returns the last status recorded identifier as a string.
	
	Example:
	
	`[Expert@HostName:0]# ./BlinkInstaller -status id`
	`finish_message`

Check the output files:

File	>Path
Log file	*/var/log/blink/logs_<DATE>/Main_log.elg*
Status file	*/var/log/blink/status.txt*

References:
Link to SK, downloads and documentation located here:
[Blink SK120193]

