MonBOX Remote Monitoring Appliance
User Guide

info@monbox.com

November 23, 2013

Contents

1 Introduction to the MonBOX Remote Monitoring Appliance

The MonBOX Remote Monitoring Appliance (MBR) is intended to make it easy to monitor systems and networks in remote or restricted access locations.

The MBR is a small, self-contained computer that can be installed where needed, and runs system and network checks locally, reporting the results back to a central monitoring server. The checks can be invoked by the MBR itself (through "tasks") or can be invoked by the central monitoring server, via SSH or NRDP connections (if direct access to the device is available).

This means that you can monitor remote locations without requiring inbound access through your firewalls, and without needing to set up VPN connections.

An MBR can be completely managed through the centralized MonBOX Management Service (MMS), and requires no direct access to the device itself —the device "calls home" (via HTTPS) periodically to check-in and receive instructions. If direct access (HTTPS) is available to the MBR, it can also be managed through a web management interface on the device itself. Basic configuration is also availble through the device console, or via SSH.

The MonBOX Management Service is a centralized service run "in the cloud" by SYONEX (the developer of the MBR). The MMS is accessed over HTTPS, through either the interactive web interface, or through a simple API. The API provides methods to manipulate an MBR, or to query or update the tasks to be performed by the MBR.

The MonBOX Remote Monitoring Appliance uses industry-standard Nagios plugins and protocols (NRDP, NRPE, NSCA) but can be used with almost any central monitoring server.

We hope that you find the MBR and MMS useful in your environment. But, we likely haven't thought of every possible situation, and we may have overlooked some things. We would very much welcome any feedback or suggestions that you may have.

2 MonBOX Remote Monitoring Appliance Use Cases

This section provides a few potential use case for the MBR and related tools, as a means of helping you become familiar with the possibilities.

The MBR is intended to be a tool that you can apply according to your needs and according to your environment.

Organizational Boundaries

You may need to monitor devices on networks that are controlled by other organizations. For example, perhaps you're the global network group charged with keeping all the divisions and offices in your company connected, but have nothing to do with the LANs behind the gateways that you manage. An MBR behind the firewall can provide you with a way to check that network connectivity to the outside world, or central corporate resources, is working just fine.

Or perhaps you're a service provider that manages the office computers for a variety of customers. An MBR in each location provides an easy way for you to keep an eye on your customers' networks, without having to worry about settings up VPNs or inbound connectivity through a variety of low end routers/firewalls.

Unreachable Locations

Perhaps you need to monitor the point of sale networks in a chain of stores, which are all connected to the internet with consumer grade DSL connections, with dynamic external IP addresses. An MBR in each store provides you with a way to monitor the internal networks, and a way to determine (and track) the current external IP address for each location.

Maybe you sell or maintain high-end network printing systems or imaging systems that are installed inside your customer premises. If you deploy an MBR with each of your systems, you have a tool for monitoring and reporting on your primary systems, even though you have no control over or access to your customers' networks.

Unreliable Connections

You're the network group in a large company, with many small locations, where the network connectivity is often very busy, or is provided over not-very-reliable links. You want to use SNMP to monitor traffic levels inside each network, but sending unencrypted UDP SNMP queries across the public internet is unreliable and unacceptable. Put an MBR in each location, and use `check_by_ssh` to "tunnel" UDP SNMP queries into each location.

Local-Only Services

Perhaps each of your remote locations has a file server for local use, which provides services only to the local network. An MBR in each location provides a way to check local-only services.

The MonBOX Remote Monitoring Appliance provides a tool for working around limitations or restrictions in your network. Think of the MBR as another level of indirection that's solving a problem in computer science.

Section <direct-checks-mbr> outlines some of the benefits of using the MBR, even when direct checks of services are possible.

3 Centralized Information Collection, Reporting and Control

Important
By default, the MonBOX Remote Monitoring Appliance is configured to automatically call out to the MMS, provide information about your network and the device to the MMS, and accept instructions from the MMS.

One of the primary features of the MonBOX Remote Monitoring Appliance is that it can be centrally managed and controlled, through the MonBOX Management Service. To make this possible:

- The MBR periodically calls the MMS and reports its current state, including network information, processes, software currently installed, etc.
- The MMS sends management commands (such as update, reboot) and tasks (network checks to run and report, network scan requests) to the MBR which are executed within your network(s).

The MBR and MMS are built and configured to only run commands on your network that you authorize, and to not run any commands or perform any activities that are not authorized by you, or that are likely to cause any disruption to your network or devices.

By default, the software installed on the MBR can be updated in response to a remote command, and the MBR can run commands allowed by the software. A software update could allow more commands to be run in response to remote commands.

This means that the MonBOX Management Service (i.e. the web-based service run and controlled by the developers of the MonBOX Remote Monitoring Appliance) has information about your network(s), and could (as a result of software updates and configuration changes) cause (essentially) arbitrary commands to run on the MBR, which is, of course, a computer attached to your network.

It is possible for you to change the configuration of the MBR so that it will not “call home”, interact with, or accept any commands from the MMS. This will disable the “remote control” abilities of the MBR/MMS system, but may be appropriate in some more restricted environments. Instructions on how to disable these features are in the subsection Disabling Remote Control and Reporting on the MBR, below.

We want you to understand what goes on “under the hood” so that you can decide whether or not the MBR/MMS combination is appropriate for your environment. Plesae contact us if you need additional information or details on how the MBR/MMS interactions work, to assist you in your evaluation of the MBR for your environment.

4 Not Suitable for All Situations

We think we've built a pretty useful tool for the system and network administrator's toolbox. And we hope you think so too.

But, the MonBOX Remote Monitoring Appliance may not be a good match for every situation. In particular, we feel we should mention:

- High security environments. Because the MBR calls out to the MMS, and because the MMS can remotely control the MBR, it may not be a good choice for environments where high levels of security are required.
- Life safety situations. Don't even think of using an MBR in situations that may affect someone's health or safety.
- Harsh environments. The MBR is likely not suitable for unprotected use in harsh environments, such as locations with extreme temperatures, or dirty or wet environments.
- Highly Critical Environments. The MBR is built on non-redundant hardware, using software that is updated dynamically. It is not designed to deliver high-reliability or high-availability in the face of hardware or software failures. If your environment is such that a failure of the MBR will cause significant disruption or loss, the MBR may not be appropriate for you.

We recommend that you read the Limited Warranty, Disclaimer of Liability appendix.

We've tried to build a tool that will work in a variety of environments, and which will provide a good and useful service that will serve you well for a long time to come. But, unfortunately, a cost effective device like the MonBOX Remote Monitoring Appliance simply isn't suitable for every need.

If you do have need of a remote monitoring tool for more challenging environments, please let us know, and we'll see what we can do.

5 MonBOX Remote Monitoring Appliance Quick Start

1. Connect the MBR to your network with a Cat5 cable
2. Connect the USB power supply to your MBR and plug it it
   - LEDs on your MBR will light up — you should see a red power indicator, and soon a green link indicator (and usually more)

3. Wait a few minutes for your MBR to finish booting, and for it to call home to the MMS
4. Connect to the MMS at https://secure.monbox.com/ using your email address and password
5. You should be able to see (in the Device List) that your new device has “called home”
   - And you should also see the IP address and DNS hostname for your MBR

And you’re ready to go!

Next, you can use the MMS interface to change settings, and give it some tasks to do.

6 Support and Services for the MBR

Each MBR comes with one year of support, access to any software updates that are released, and one year of access to the MonBOX Management Service.

If you run into any problems, or have any questions or suggestions for the MBR and/or its related services, please drop us a line and we will do our best to help you out.

7 Finding an MBR on the Network

Each MBR has a unique name — the name is on the label on the MBR, and is shown in the MBR’s administrative menu, web management interface, and in the MMS.

When initially connected to a network, an MBR will attampt to use DHCP to acquire an IP address. If desired (or required) a static address can be configured through the MBR’s web management interface or administrative menu. The current IP address is visible in the MBR’s web management interface and administrative menu.

When an MBR calls home to the MMS, the MMS will set up a dynamic DNS entry for the MBR (which will typically be available on the public network within 15 minutes). The dynamic DNS names are in the `mbr.monbox.com` domain, so an MBR named “bob” will show up in DNS as `bob.mbr.monbox.com`. (This DNS name is also included on the label on the MBR.)

Please note that in most cases, the IP address will be an “internal” network address (behind a firewall) so will typically be of limited use from other outside networks.

8 Security and Access Control

There are several security mechanisms used by the MonBOX Remote Monitoring Appliance

- An MBR has an administrative password, a relay token, and may be configured to allow NRPE or SSH access to the "nagios" userid for running checks.
- In MMS, an account has an API token, and each user associated with that account has a password associated with their email address.

8.1 MBR Access

By default, an MBR is controlled by the MMS — it calls out periodically to the MMS.

The MBR's web management interface is always available (if the MBR is on a network you have access to). Use the userid "admin" and the adminstrative password (provided with the MBR, or set via the administrative menu or the MMS) for web access.

You can access an administrative menu on the console by connecting an HDMI display and a USB keyboard. No userid or password is required on the console.

If SSH has been enabled on the MBR (and you have access to the network the MBR is connected to), you can SSH to the MBR with userid "admin" and the administrative password. This will connect you to the administrative menu on the MBR.

If SSH access has been enabled, root SSH access can also be enabled (though it will be automatically disabled each time the MBR is rebooted). The root password is the same as the normal administrative password.

SSH keys can be used for access to the nagios user (for use by `check_by_ssh` for example), but there is not currently any SSH key support for the admin or root users.

8.2 MMS Access

Access to the MMS web interface is over HTTPS using a user's email address and a per-user password.

The MMS groups users and devices into "accounts" e.g. your company would have one MMS account, and may allow multiple users access to that account.

API access to the MMS is also over HTTPS, and uses a per-account API token. The API token is available in and can be changed using the MMS web interface.

8.3 Lost or Forgotten Passwords

For the MMS, you can request a new password from the login page.

For an MBR:

- You can connect an HDMI display and a USB keyboard and change the MBR's password from the administrative menu.
- If the MBR is still talking to the MMS, you can use the MMS to change the MBR's password.

For MMS API access, the current API token is available in MMS, under Account / API.

For MonBOX Relay Service access, the MBR's relay token is available in MMS, under Devices / Relay / Configure.

8.4 Disabling Remote Control and Reporting on the MBR

Note

We really recommend that you do not disable the remote control features of the MBR, because we think that you'll be limiting the utility of the device, and missing out on some great features. But we would be remiss if we didn't make it possible for you to decide for yourself what's most appropriate for your environment.

By default, the MonBOX Remote Monitoring Appliance calls the MonBOX Management Service periodically, and accepts commands, configuration, and other information from the MMS. We think this is one of the great features of the MBR, and hope you do too.

However, if you wish to disable these features, this is how to do it. Access to the MBR web management interface is required to disable or re-enable these features.

- Tasks, such as update, reboot, cron and Nagios object configuration. To stop the MBR from accepting task requests from the MMS, choose Config / Tasks from the web management interface menu, and on the “Tasks Configuration” page, set “Enable Tasks” to “No”, and save the configuration.
- CallHome, the periodic reporting of current operational state to the MMS. To stop the MBR from “calling home” to report on its state and well-being, choose System / Config / CallHome from the web management interface menu and on the “CallHome Configuration” page, set “Enable CallHome” to “No”, and save the configuration.

Again, we hope that you will choose to keep these features enabled. If you have concerns or questions about the operation of these features, please feel free to contact us and we will do our best to help you out.

9 The MBR Hardware Platform

The MonBOX Remote Monitoring Appliance is built on the Raspberry Pi single board computer.

We chose to use the Raspberry Pi because of its combination of no moving parts, high function and low cost, and its compact size.

The underlying OS is Raspbian, a Debian derivative optimized for the Raspberry Pi, with a variety of open source software packages on top.

We have stripped down the base OS installation, written all sorts of tools to implement the MBR functionality, and wrapped it all up with tools to run with a read only filesystem.

We use a read only filesystem for added system reliability. It greatly reduces the possibility of a corrupted filesystem (in cases of system or power failure), and reduces the number of writes to the SD card (which is reputed to make the card last longer). Plus, writing SD card on a Raspberry Pi is not blindingly fast, so if we avoid SD card writes, and use a memory filesystem instead, we should have better overall performance.

But because it's a read only filesystem, things aren't quite as they would be on a "normal" computer. So we ask you to resist the urge to use the shell access that is available on your MBR to attempt to make your own modifications.

10 MBR Installation and Maintenance

The MBR is intended to be easy (dare we say trivial?) to install and to require little or no maintenance.

Installation is typically just as simple as described in the Quick Start section.

- Use the provided power supply to ensure sufficient current is available to power the MBR.
- As with any electronic device, avoid locations that are hot or cold, or dusty or dirty. And especially avoid wet and damp locations.

As is common, there are no user serviceable parts (or software) inside. The administrative menu provides a shell access menu item, primarily as a tool for trouble shooting.

No ongoing hardware maintenance is required. You should check for software updates periodically, and apply any recommended updates, either through the administrative menu, or through the MMS.

11 MBR Functional Overview

The MonBOX Remote Monitoring Appliance is intended to be a reliable and effective tool for system and network monitoring in remote or less accessible environments.

The web management interface and administrative menu on the MBR, coupled with the MonBOX Management Service, are intended to eliminate the need (or perceived need) to customize the operating system and environment on the MBR. And, combined with the read only filesystem, and other tools and mechanisms on the MBR, the intention is to make it difficult to manually (or otherwise) modify the MBR. This is intended to improve the overall reliability and security of the device.

Note

Because we expect that many or most MBRs will be installed in remote locations, or at customer premises, we recognize that a software, system, or configuration error that renders an MBR inaccessible or ineffective could be very inconvenient to fix. Avoiding these kinds of problems is the primary reason for using a read only filesystem, and for limiting uncontrolled administrative access to the MBR.

Functionally, the MBR performs a small number of activities:

- Once an hour, the MBR does periodic tasks, which includes “calling home” to the MMS, reporting on its current state — processes, resources, network settings, installed software, etc. (“Call-Home” can be disabled, but that will limit some of the functionality.)
- Every 15 minutes, the MBR contacts the MMS to check for “work” tasks to perform — new settings for cron checks, Nagios object definitions, or requests to do network scans, software updates, reboot, etc. (This can also be disabled, but will prevent the use of these functions.)
- The MBR runs cron tasks at the specified intervals, which typically include reporting the results to a monitoring server of some form.
- If Nagios objects are defined, the MBR runs a Nagios process, which will perform the checks specified in the object definitions, and will (presumably) report the check results to a monitoring server.
- If SSH access is enabled, tools like the `check_by_ssh` plugin can be used (from a central monitoring server) to run active checks on the MBR and return the results.
- If NRPE is enabled, tools like `check_nrpe` can be used in a similar fashion.

12 MMS — the MonBOX Management Service

The MonBOX Management Service (MMS) provides a centralized management interface for control of and reporting on your MBR devices. The MMS is a centralized service provided by SYONEX for use in managing MBR devices.

The MMS is used for management, and is not required for ongoing monitoring. If the MMS is unavailable or unreacahble for a short period of time, your monitoring processes will (in most cases) be unaffected.

Each MBR is recorded in MMS, and is associated with an account, and each account (for a company or organization) has one or more userids associated with it. Userids are in the form of email addresses, and each userid associated with an account can control the account and the devices associated with the account.

The primary menu items in the MMS are Home, Devices, Account, Password, Help and Logout. The Account submenu provides information on the account (name, address, etc.), management of the users associated with the account, and account API access control and token generation.

Some of the pages in the MMS have a ContextHelp menu item, which links to the appropriate section in the MonBOX Remote Monitoring Appliance User Guide.

12.1 MMS Device Management

The primary purpose of the MMS is, not surprisingly, the management and control of MBRs. There are five sections in the Devices submenu: List, Manage, Relay, Tasks and Discovery.

12.1.1 Devices / List

The Devices / List page shows a summary of all the MBR devices associated with the account. The name of the currently managed MBR is shown in bold type. To manage a different MBR, click the MBR's name link.

Each MBR has a fixed name, which is usually an English word (on the assumption that words might be easier to remember than serial numbers). An MBRs name is shown on the label on the device, and is also shown in the device's web interface and administrative menu.

Dynamic DNS is used automatically to associate a standard FQDN with the device, in the domain `mbr.monbox.com` so the IP address of an MBR named `banana` would be available as a DNS A record for `banana.mbr.monbox.com`.

The information shown for each device includes such things as hostname, MAC address, serial number, and description, as well as the most recent operational information for the device that has been reported to the MMS:

- Tasks? Is the MBR configured to accept tasks from the MMS?
- Updates? Are software package updates known to be available for the MBR?
- Last Contact: the date/time the MBR last connected to the MMS.
- Last Int IP: the local IP address last reported by the MBR to the MMS (i.e. the address reported by `ifconfig eth0`). This is the address used in the dynamic DNS record for the MBR.
- Last Ext IP: the IP address last used to connect to the MMS — this is typically the external public address of the firewall or router that connects the MBR to the public internet.

Depending on network configurations (firewall rules, NAT, etc.) the reported IP addresses may or may not be useful, but they may give you some hints on where the MBR was last seen.

If you wish to use your own naming scheme for your MBRs, you can usually set up the desired name in your own DNS data as a CNAME record referring to the `mbr.monbox.com` A record.

12.1.2 Devices / Manage

The Devices / Manage submenu gives access to more detail for the currently managed MBR, and provides mechanisms to modify the MBR's configuration.

The Summary page shows the summary information for the device, and provides buttons to request that the MBR perform a software update or reboot itself. The requests will be acted on the next time the MBR contacts the MMS for tasks to perform (typically every 15 minutes).

The CallHome Report page shows the most recently received CallHome Report received from the MBR. It provides a variety of information gathered from the MBR (processes, mounts, installed software, etc.), primarily to aid in device tracking and trouble shooting.

The Device Password page lets you set the administrative password for the MBR (via a task). This is the password used in the device's web management interface or when using SSH to connect to the device as the `admin` user.

More remote management functions are being considered — we would welcome your feedback and suggestions for additional useful functionality.

12.1.3 Devices / Relay

The Devices / Relay submenu allows you to enable/disable access from the MBR to the MonBOX Relay Service, and to (re-)generate the device's relay token.

If the relay token is (re-)generated, a request is submitted to update the stored relay token on the MBR, which will be acted on the next time the MBR contacts the MMS for work tasks.

You can also view any results recorded in the relay service from a device.

12.1.4 Devices / Tasks

The Devices / Tasks submenu allows you to manage the Cron check and Nagios object definition fragments for the current device.

The two third level menu selections, Cron and Nagios, provide the same functions, for Cron check and Nagios object definitions, respectively.

Each type of fragment can have multiple fragments that can be edited and managed independently. The fragments are displayed in alphabetical order, and passed to the MBR in the same order.

At the top of the page is a button menu with the following selections:

- Show: Expand all fragments on the page, so that the contents of the fragments are visible.
- Hide: Collapse all fragments on the page, so that on the fragment buttons and the name of the fragment are visible.
- Download: Concatenate all fragments together (along with comments indicating the start and end of each fragment, etc.) and download the resulting file through the browser.
- Add: Create a new fragment and edit it in the fragment editor.
- Activate: Verify the fragments and, if verification is successful, request that the MBR download and activate the fragments.
- Verify: Check the syntax of the fragments and report any errors that were found.

Each fragment is displayed in a separate sections, headed by a button menu and the name of the fragment. The button menu has the following selections:

- Show or Hide: Expands or collapses the fragment.
- Download: Downloads a copy of the fragment through the browser.
- Edit: Opens the fragment editor for this fragment.
- Delete: Deletes this fragment — confirmation is requested before deletion.

More information on Tasks and their syntax is in the Tasks and Fragments, and Performing Checks section, below.

12.1.5 Devices / Discovery

The MBR can run a scan on its local network, to aid in discovering what devices and services are visible on that network. It uses the “Nmap” (“Network Mapper”) tool to do the scan. More information on Nmap is available on the Nmap web site at http://nmap.org/.

The two Discovery submenus are Request and Reports.

A Discovery / Request can be submitted to request that the MBR run an nmap scan on its local network. The report from the scan will typically be retured to the MMS within half an hour.

The Discovery / Reports submenu provides access to recent nmap Discovery reports that have been returned from the MBR. You can view the report online, or download the XML report for further analysis.

13 MRS — the MonBOX Relay Service

The MonBOX Relay Service provides a method for a device running monitoring checks to transfer the results back to a monitoring server, without requiring a direct connection between the systems (in either direction). The check results are passed from the MBR to your monitoring server through the MRS intermediary, which means that “inbound” network connections to the MBR or the monitoring server are not required. i.e. You don’t need to open your firewall on either end in order to monitor your networks.

The MRS is a centralized service provided by SYONEX for use in managing MBR devices. If you are using the MRS as part of your monitoring, and the MRS is temporarily unavailable or unreacahble, your monitoring results will be unavailable during that time. The MBR is built with reliability in mind — we want to avoid service disruptions even more than you do. If you have concerns about MBR reliability or availability, please contact us, and we will try to answer your questions, and offer suggestions about how to configure your monitoring environment and checks to help ensure that your needs are met.

The checking device runs a service check (e.g. checks that an HTTP server is serving up pages), reports the results of the check to the MRS, and the monitoring server periodically polls the MRS for the most recent check result. Communication is via NRDP.

The MRS was inspired by the Nagios Reflector service.

Authentication for the MRS is by way of a relay token, using a distinct token for each MBR. When a relay token is (re-)generated, the MMS creates a task to transfer the new token to the MBR. This means that MRS authentication for use from an MBR is automatic. You will, however, have to copy the relay token to your monitoring server.

To collect check results from the MRS, use the `check_reflector.py` plugin (which is available at http://assets.nagios.com/downloads/exchange/reflector/check_reflector.py) on your monitoring server to poll the MRS.

On the MBR, use the fragments mechanism provided through the MMS (see "Tasks and Fragments" below) to run checks and report the results to MRS.

You can, of course, submit an MBR's check results to any NRDP or NSCA server, using the `nrdp_check` and `nsca_check` tools. We've tried to provide "mechanism not policy", so that the MBR can be used in the way that works best for your environment and your network.

14 MBR Management

The MBR itself allows local management of its settings and configuration. And, if your MBR is directly accessible on your network from your central monitoring server, there are several ways that your server can connect to the MBR and cause monitoring checks to run.

If you're not sure where your MBR is on your network, see the section Finding an MBR on the Network, above. And you may wish to review the subsection on MBR Access for security and access control information.

14.1 Administrative Menu

The administrative menu is a text-based menu that can be used for basic management of the MBR. It runs on the MBR's console, so it is available if you connect a display and keyboard to the MBR. The administrative menu is also used as the shell for the "admin" user, so if SSH is enabled to the MBR and you SSH in as the user "admin", you will get the administrative menu.

The menu has a small number of selections that should allow you to perform the necessary steps to get the MBR running on your network.

- Show status — Displays basic system status: hardware, uptime, file systems, network, processes, etc., which could be useful in identifying the MBR or basic troubleshooting.
- Basic configuration — Provides a menu driven interface for setting the hostname, network settings, etc. In most cases this will not be necessary, as DHCP will be used for network configuration. If you have a WiFi network interface (see WiFi Wireless Networking), you can set wireless network parameters here.
- Configuration save — Save any configuraiton changes to the SD card memory.
- Change password — Changes the admin password for the web management interface and for ssh access. (This also changes the password for the "root" userid, which is available if root SSH is enabled.)
- Update software — Download and install available software updates.

- Reboot system
- Halt system
- Shell — Gives you access to a command shell running as the root user. Note that the SD card is mounted read only, so some things will not work as they do on a "normal" computer.
- Exit menu — This will exit the SSH session, or start a new menu instance on the console.

14.2 Web Management Interface

Each MBR runs a web server to provide the web management interface, which is accessed over HTTPS. The MBR has a self-signed SSL certificate.

The MBR also allows a small number of "quick links" that allow quick, unauthenticated, plain text access to certain status information, such as the current data from temperature sensors. See the Quick Links on the MBR appendix for more details on quick links.

The primary menu items in the MBR's web management interface are Home, Status, Config, System, Password, and Logout.

14.2.1 Status

The Status menu provides submenu choices to examine the current status of the NRPE and SSH services, and information about Tasks and available monitoring Plugins.

Currently, the information provided for Tasks and Plugins is very minimal.

14.2.2 Config

The Config menu provides submenu choices for configuring the monitoring services on the MBR

- NRPE — Enable/disable the NRPE service, set listen port, timeouts and access rules for which hosts are allowed to connect.
- SSH — Controls the SSH service and access via SSH.
  - Main — Enable/disable the SSH service, set listen port, allow root SSH access
  - Authorized — Manage SSH authorized keys for the nagios user. These are the keys that would be used to run checks via, for example, the `check_by_ssh` plugin on your central monitoring server.

  - Keys — Displays the passphrase-less keys that would be used by the local nagios user to connect to other hosts, if, for example, you're using `check_by_ssh` on the MBR to run local checks on systems from the MBR. You can also force a deletion and regeneration of the keys.
  - KnownHosts — Displays the current known hosts list for the local nagios user. The known hosts list is populated as needed. Provides the option to empty the known hosts list, which could be useful if a host being monitored changes its host key.

- Tasks — Enable or disable accepting tasks from the MMS. See the Disabling Remote Control and Reporting on the MBR section for reasons why you should not decide to disable tasks without due consideration.
- Plugins — Configure monitoring plugins. No functions currently available for plugin configuration.
- Save — Save any configuration changes to the SD card and make them active.

14.2.3 System

The System menu provides submenu choices for configuring and managing the operation of the MBR system and software.

- Config — System configuration options.
  - CallHome — Enable or disable calling home to the MMS. See the Disabling Remote Control and Reporting on the MBR section for reasons why you should not decide to disable CallHome without due consideration.
  - Hostname — Set the hostname of the MBR.
  - Network — Enable/disable DHCP and/or set a static IP address for the MBR.
  - Syslog — Configure a syslog server for logging from the MBR.
  - Save — Save any configuration changes to the SD card and make them active.
- Operations — Halt or reboot the MBR, force restart system services such as crond and syslog.
- Status — Display running system status, contents of the local syslog log file.
- Updates — Check for and/or apply available software updates.

15 Direct Checks via the MBR

If the MBR is reachable on your network from your central monitoring server, you can send direct monitoring checks through the MBR by two methods: SSH and NRPE.

If the MBR is accessible on your network, isn't it possible that the other devices on the same subnet are also accessible? And if so, what are the advantages of checking via the MBR rather than checking the devices directly?

- The MBR could be in a DMZ network, with permission to access internal machines, but with the internal machines not directly accessible from outside the network.
- Proxying checks through the MBR may mean that only a very limited set of ports needs to be allowed through an external firewall — SSH or NRPE to the MBR, rather than ports that can access all devices of interest.
- SNMP or other UDP-based checks may be more reliable when performed from a device on a local network. UDP packets may not be allowed in through the firewall, or UDP (sometimes referred to as Unreliable Datagram Protocol) packets may not be reliable over a WAN or public internet link.
- Some protocol checks (SNMP V1, HTTP) are of necessity, un-encrypted and un-authenticated. Proxying those checks through the MBR means that the traffic from the central monitoring server to the MBR is encrypted.

Monitoring checks can be run on (or through) the MBR using the standard `check_by_ssh` and `check_nrpe` plugins.

For ease of integration into an existing monitoring system, you may find the MBdivert plugin wrapper useful. MBdivert was written to make "diverting" checks through an MBR easy. The `mbdivert.cfg` configuration file provides rules to transparently divert checks through an MBR (or any other system running SSH or NRPE). More information on MBdivert (and the MBdivert software distribution itself) is available at http://www.syonex.com/resources/software/

16 Tasks and Fragments, and Performing Checks

The MMS provides the ability to create and manage configuration "fragments" that are transferred to the MBR and used to run monitoring checks and (presumably) report the results to a central monitoring server. Fragments can be modified both through the MMS web interface and through the MMS API.

There are two types of fragments:

- Cron — These fragments get converted into a Linux crontab file, and so run the specified check commands at a fixed interval.

- NagiosObjects — These fragments are combined into a Nagios object definition file, and so provide more flexible scheduling and the ability to make use of some of the features (dependencies, etc.) available in Nagios.

Fragment files are processed in alphabetical order, and multiple files are effectively equivalent to a single file containing the same content. The ability to have multiple fragment files is intended to make your life easier, and allow you to organize your settings in the way that works best for you. For example, you could use a separate fragment file for each host that you are monitoring.

From the point of view of a central monitoring server, checks run via Cron or NagiosObjects fragments are likely "passive" checks (using the Nagios terminology). That is, these are checks that are not directly invoked by the central monitoring server, but instead are invoked by another system and the results of checks are reported back to the central monitoring server. (Use of the MonBOX Relay Service blurs this distinction somewhat.)

16.1 Custom Reporting Plugins

The MBR includes some custom reporting plugins, which invoke a standard plugin, and report the results to another server (via NSCA or NRDP). (Another example of another level of indirection solving a problem.)

If calling these plugins from within Nagios, you can use Nagios custom object variables to provide some defaults to these plugins. Additionally, these plugins will make use of standard Nagios macros (e.g. $NAGIOS_HOSTNAME) as the defaults for some options.

16.1.1 `nsca_check`

The `nsca_check` plugin runs a standard plugin and reports the results to an NSCA server, optionally returning the plugin output to the caller.

`nsca_check` accepts the following options:

-H hostname
Sets the hostname for the check being performed, used when reporting the results to the NSCA server.
Default: `$NAGIOS_HOSTNAME`

-S service
Sets the servicename for the service check being performed, used when reporting the results to the NSCA server.
Default: `$NAGIOS_SERVICEDESC`

-c configfile

The config file to use with the `send_nsca` client.
Default: `$NAGIOS__SERVICENSCACONFIG`, or `$NAGIOS__HOSTNSCACONFIG`

-h nscahost

The host running the NSCA server to report results to.
Default: `$NAGIOS_NSCAHOST`, `$NAGIOS__SERVICENSCAHOST`,
or `$NAGIOS__HOSTNSCAHOST`

-p nscaport

The port to connect to on the NSCA server.
Default: `$NAGIOS_NSCAPORT`, `$NAGIOS__SERVICENSCAPORT`,
or `$NAGIOS__HOSTNSCAPORT`

-t timeout

The connect timeout for connecting to the NSCA server.
Default: `$NAGIOS_NSCATIMEOUT`, `$NAGIOS__SERVICENSCATIMEOUT`,
or `$NAGIOS__HOSTNSCATIMEOUT`

-q

Quiet — don't provide the plugin output or the exit code of the plugin to the caller. This is useful if invoking `nsca_check` outside of Nagios (from cron for example).

Sample nsca_check usage:

```
nsca_check -q -h nsca.example.com \
    -S www -H google check_http -H google.com
```

16.1.2 `nrdp_check`

The `nrdp_check` plugin runs a standard plugin and reports the results to an NRDP server, optionally returning the plugin output to the caller.

`nrdp_check` accepts the following options:

-H hostname

Sets the hostname for the check being performed, used when reporting the results to the NRDP server.
Default: `$NAGIOS_HOSTNAME`

-S service

Sets the servicename for the service check being performed, used when reporting the results to the NRDP server.
Default: `$NAGIOS_SERVICEDESC`

-t nrdptoken

The token to use for authenticating to the NRDP server.
Default: `$NAGIOS_NRDPTOKEN`, `$NAGIOS__SERVICENRDPTOKEN`,
or `$NAGIOS__HOSTNRDPTOKEN`

-u nrdpurl

The URL for the NDRP server to send the results to.
Default: `$NAGIOS_NRDPURL`, `$NAGIOS__SERVICENRDPURL`,
or `$NAGIOS__HOSTNRDPURL`

-q

Quiet — don't provide the plugin output or the exit code of the plugin to the caller. This is useful if invoking `nrdp_check` outside of Nagios (from cron for example).

Sample nrdp_check usage:

```
nrdp_check -q -t token -u https://www.example.com/nrdp/ \
    -S www -H google check_http -H google.com
```

16.1.3 `relay_check`

The `relay_check` plugin is simply a wrapper around `nrdp_check` that reports to the MonBOX Relay Service. `relay_check` accepts the same options as `nrdp_check` (in fact, it just passes the options over to `nrdp_check`). Don't use the `-u` or `-t` options with `relay_check` because that will override the MRS settings in `relay_check`.

Sample relay_check usage:

```
relay_check -q -S www -H google check_http -H google.com
```

16.2 Cron Fragment Syntax and Usage

The cron fragments use a special syntax, intended to make it easy to define monitoring checks with a minimum of repetition.

Cron fragment files are processed in alphabetical order (of the name of the fragment). Blank lines, or lines where the first non-blank is "#" are ignored.

Cron fragment files consist of a number of commands, which define servers to report to, set defaults, or define monitoring checks to perform. Defaults take effect from that point on, and persist across file boundaries e.g. a fragment file named “0defaults” would set the defaults for the remaining files, unless redefined in a following file.

When activated, the cron fragment files are processed and used to create a standard “crontab” file, and a request is created to transfer the resultant file to the MBR. The first time for a check to run each hour is randomized (within the desired interval) so that every check doesn’t happen at the top of the hour — the checks are spread out to even out the load.

The following lines are understood in cron fragment files:

`interval n`

How often (in minutes) to schedule checks. Default: 5.

`server servername nrdp token url`

Defines a server to report check results to using NRDP. The `servername` is remembered and becomes the current default, and can be later referred to in `check` or `relay` lines.

`server servername nsca host [port [timeout]]`

Defines a server to report check results to using NSCA. The `servername` is remembered and becomes the current default, and can be later referred to in `check` or `relay` lines. The `port` and `timeout` parameters are optional.

`host hostname`

Sets `hostname` to be the current default host for `check` and `relay` lines.

`check service[:host[:servername]] plugin pluginargs`

Defines a monitoring check named `service`. The `service` and `host` are passed to `servername` (using NRDP or NSCA) and are used to tell `servername` what host and service the check result is for. If `servername` or `host` is not provided, the current defaults are used. If `servername` or `host` is provided, the value becomes the current default. The `plugin` parameter is the name of the monitoring plugin to run for the check (e.g. `check_http`) and the `pluginargs` are the arguments to be passed to the plugin.

`relay service[:host] plugin pluginargs`

Defines a monitoring check named `service`, to be reported to the MonBOX Relay Service. The `host`, `plugin` and `pluginargs` parameters are as in the `check` fragment line.

Sample Cron Fragment

```
# This is a comment

# We want the checks to run every 3 minutes
interval 3

# Define an NSCA server named "mynsca" on the host nsca.example.com
# using the defined port
server mynsca nsca nsca.example.com
```

```
# Define an NRDP server named "mynrdp" on the host nrdp.example.com
# using HTTPS, and mytoken for authentication.
server mynrdp nrdp mytoken https://nrdp.example.com/nrdp/

# Check HTTP to google.com, and report it to the mynrdp server (the
# most recently referred to server), and label the result with host
# "google" and service "www"
check www:google check_http -H google.com
# Check google.ca and report and tag the service "wwwca"
check wwwca check_http -H google.ca

# Check HTTP to monbox.com, and report the result to the NSCA server
# "mynsca", tagged as host "monbox" and service "www"
check www:monbox:mynsca check_http -H monbox.com
# Check HTTPS to monbox.com, and report to the same server.
check secure check_http -S -H secure.monbox.com

# We want the following checks to be run every 5 minutes
interval 5

# Check SMTP to smtp.syonex.com and use the MRS to report it
relay smtp:syonex check_smtp -H smtp.syonex.com
# Check HTTP to www.syonex.com
relay smtp check_http -H www.syonex.com
```

16.3 Nagios Object Definition Fragments and Nagios on the MBR

The object definition fragments are standard Nagios object definition files. The defined fragment files are combined with a small default set of Nagios objects (see Default Nagios Objects below). These defaults allow you (if you wish) to define a single service check, without having to define a full set of hosts, hostgroups, contacts, timeperiods, etc. You can, of course, define any valid Nagios objects that you wish in object definition fragments, and you can ignore the default Nagios objects. (But be aware that the default Nagios objects exist, and will be included, so don't choose any object names that conflict with the default objects.)

There is not currently any mechanism to customize any options in the standard `nagios.cfg` or `resource.cfg` files. (Please let us know if this functionality would be useful for you.)

If any object definition fragments are defined and activated, the MBR will run the Nagios daemon using those object definitions.

Default Nagios Objects

```
# This is intended to be the minumum set of objects that do nothing
# that will stop any pre-flight errors and warnings.

define contact{
    contact_name                        nocontact
    alias                               nocontact
    service_notification_period         never
    host_notification_period            never
    service_notification_commands       nocommand
    host_notification_commands          nocommand
}
define contactgroup{
    contactgroup_name                   nocontactgroup
}
define timeperiod{
    timeperiod_name                     never
    alias                               never
}
define command {
    command_name                        nocommand
    command_line                        /bin/true
}
define host {
    host_name                           nohost
    alias                               nohost
    hostgroups                          nohostgroup
    check_command                       nocommand
    max_check_attempts                  1
    contact_groups                      nocontactgroup
    notification_period                 never
    check_period                        never

}
define hostgroup {
    hostgroup_name                      nohostgroup
    alias                               nohostgroup
}
define service {
    service_description                 noservice
    host_name                           nohost
    check_command                       nocommand
    max_check_attempts                  1
    notification_period                 never
    check_period                        never
}
```

17 API Access to the MMS

There is a simple API for accessing the MonBOX Management Service. The API is intended to make it possible to integrate your MBR into your existing systems without too much trouble.

The API is built on HTTPS, using a simple token based authentication mechanism, combined with your Account Code used in the API URL. There is one API access token per account, which is (re-)generated through the MMS. When authenticating, use the API token as the username, and use an arbitrary filler string for the password (only the token is used for authentication).

Here is a simple example, which gets a list of devices in your account (assuming your Account Code is `abcdco` and your token is `123456`):

```
% curl -u 123456:x https://secure.monbox.com/api/v1/abcdco/devlist
liststart
listlegend device name tasks updates lastintip lastextip lastcontact
device apple Yes No 192.168.1.111 1.2.3.4 1363229589
device banana Yes No 192.168.1.222 1.2.3.4 1363229761
listend
apiend
```

A few notes:

- API call URLs start with `https://secure.monbox.com/api/v1/` followed by the Account Code, a slash, and the API request string.
- Many API calls return a list of things. Lists are surrounded by `liststart`, `listlegend`, and `listend`.
- Almost all API calls end with an `apiend` line. This lets your API tools confirm that they have received a complete response from the server.
- API calls that “get” a file do not end with an `apiend` line, and return only the content of the file (which can be confirmed with the checksum provided in the file list API call).

17.1 API Requests

In the following, `devname` means an MBR device name, and `fragtype` is either `cron` (for cron task fragments) or `objects` (for Nagios object definition file fragments).

`devlist`
List the devices associated with the account.

`devlist/devname`
List only `devname`.

`reboot/devname`
Request a reboot task for `devname`.

`update/devname`
Request a software update task for `devname`.

`fragtype/list`
List all `fragtype` fragments for all devices.

`fragtype/list/devname`
List for just one device.

`fragtype/verify`
Verify (check syntax, consistency, etc.) `fragtype` fragments for all devices.

`fragtype/verify/devname`
Verify for just one device.

`fragtype/activate`
Activate (verify, then request download to devices) `fragtype` fragments for all devices.

`fragtype/activate/devname`
Activate for just one device.

`fragtype/delete/devname/fragname`
Delete the `fragtype` fragment for device `devname` named `fragname`.

`fragtype/get/devname/fragname`
Get (download) the `fragtype` fragment for device `devname` named `fragname`.

`fragtype/put/devname/fragname`
Put (upload) a new `fragtype` fragment for device `devname` named `fragname`, replacing any existing fragment of the same type, device and name.

18 Add-Ons and Additional Functionality

18.1 Temperature and Environmental Sensors

The MBR has support for the TEMPer USB temperature sensors from PCsensor.[1]

Temperature sensor data is available from the status reports, the call home report, or through * the `check_tempers` plugin (see Available Monitoring Plugins), or * the `temper` quick link (see Quick Links on the MBR)

Supported sensors are available directly from MonBOX.com

18.2 WiFi Wireless Networking

The MBR has built-in wired ethernet, and provides basic support for WiFi wireless networking, using supported USB WiFi interfaces (supported by the Raspbian distribution, and not requiring more power than is available).

WiFi support assume WPA or WPA2 authentication, and DHCP for address assignment. Up to three SSID/password combinations can be defined for each MBR, through the Administrative Menu, the Web Management Interface, or through MMS.

Supported WiFi adapters are available directly from MonBOX.com

A Trouble Shooting

We hope nothing goes wrong, but if your MBR does not seem to be working properly.

1. Disconnect any unnecessary USB devices: The Raspberry Pi (and power supply) has limited power available for powering USB devices.

2. Check the power supply: The provided power supply should be sufficient, but other similar-looking power supplies might not provide enough current to reliably power a Raspberry Pi.

3. Check the LEDs: Red on LED #4 means power, green on #5 is SD card activity, and the other 3 are wired network connectivity.

4. Power cycle: Does a reboot sort things out?

[1] The MBR uses the TEMPered software package by EdorFaus, and so probably supports any PCsensor USB device supported by that package. https://github.com/edorfaus/TEMPered

5. Console messages: If you can hook up a monitor and keyboard, the administrative menu (or boot details) may provide some clues.

   - For example, the device might not be properly acquiring an IP address.

6. Check the SD card: If the SD card is not properly seated in the Raspberry Pi, it will have trouble booting.

B Quick Links on the MBR

Most connections to the MBR are authenticated and encrypted. There are a small number of web-based "quick links" that require neither authentication nor encryption.

For example, the current temperature sensor readings can be retrieved from the MBR example.mbr.monbox.com with

```
% curl http://little.mbr.monbox.com/quick/temper
temper /dev/hidraw1     TEMPerV1.2      28.38   XX
datetime 20131123181647
```

Quick links are, of course, only useful if you have direct network access to the MBR.

The available quick links are:

temper
Returns the output from the `temper` command, with each line prefixed with `temper`, followed by a `datetime` line containing the date and time in the format `yyyymmddhhmmss`. The temper output is the device path, device type, termperature, and humidity. If the device does not provide a humidity reading, the value `XX` is returned. The temper output fields are tab separated, the prefix labels are followed by a space character.

temperc
Like `temper` but forces temperatures to be in degrees Celsius (this is the default).

temperf
Like `temper` but forces temperatures to be in degrees Fahrenheit.

C Available Monitoring Plugins

The MBR has a variety of Nagios-compatible monitoring plugins installed and available for use.

The MBR has the standard Nagios Plugins distribution installed.

A list of available plugins and (likely) summary help information will be made available. The list of plugins currently installed on an MBR is available through the web management interface, under Status / Plugins.

If you are interested in having additional monitoring plugins made available, please let us know. We can't (well, won't) install absolutely every possible plugin, but we want to make sure that a comprehensive and functional set of plugins is available on the MBR.

D Languages and Internationalization

At this point, the MBR and MMS are available in the English language only. Initially we are aiming at North American users, for ease of documentation, support, and electrical and regulatory compliance.

We hope to become less North American English centric as time and resources allow, and hope for your understanding.

E Limited Warranty, Disclaimer of Liability

We warrant that the MonBOX Remote Monitoring Appliance and related tools generally operate as described in this document.

But, as is typical with software products,

- We do not warrant that the MBR and related tools are appropriate for any particular purpose. You must judge whether the MBR is suitable for use in your environment.
- We disclaim any liability for any damages, direct or indirect, that may arise in conjunction with the use of the MBR and related tools, regardless of how caused.

We hope that you will be happy with your MBR, and that it will be useful in your environment. If you are unhappy with your MBR, we want to know.

F Trademarks, Licenses, Copyrights, etc.

The MonBOX Remote Monitoring Appliance is built on top of the Raspbian Linux distribution, and many other open source tools. The copyright to all that software remains (or course) with the original copyright holders. That software is licensed under various open source licenses, much of it under the GNU General Public License.

It is our intention to fully abide by the letter and the spirit of all applicable software or other licenses. If you believe that we may not be living up to that intention, please let us know, and we will do our best to make things right.

F.1 MonBOX Custom Software and Tools

We have written many programs and other tools to implement the MonBOX Remote Monitoring Appliance, the MonBOX Management Service, and the MonBOX Relay Service. We retain the copyright on the software and tools that we have created, and reserve all rights to them.

The purchase of a MonBOX Remote Monitoring Appliance gives you the right to use the MonBOX software and tools installed on the device. An annual subscription to the MonBOX services provides you with access to the MonBOX Management Service, the MonBOX Relay Service (in conjunction with the MBR), and access to any MonBOX software updates made available during the term of your subscription.

F.2 Trademarks

This document, and the MonBOX software, refer to various software packages, products and trademarks that are the property of others. It is our intention to comply with any guidelines or restictions on the use of trademarks and other trade names belonging to others.

Nagios is a trademark of Nagios Enterprises, LLC.

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

G Glossary

It will be helpful if you are familiar with the Nagios monitoring system, and the terminology established surrounding it.

Account Code
This is the name used for your account in the MMS, and is available under Account / Summary.

API Token
Secret “password” used with your Account Code to access the API, (re-)generated through the MMS. API Tokens are (currently) 16 alphanumeric characters.

Central Monitoring Server
We use this phrase to refer to a server running your choice of monitoring software (such as Nagios). This will typically be where you define all the hosts and services you wish to check on your network. Your monitoring server will use one or more of the available mechanisms to collect and act on monitoring check results generated on your MBR.

MBR
MonBOX Remote Monitoring Appliance

MMS
MonBOX Management Service

MRS
MonBOX Relay Service — an NRDP-based service that allows an MBR to send a check result to the relay server, and a central monitoring server to poll the relay server for the most recent results.

NRDP
Nagios Remote Data Processor — a data transport mechanism (protocol) for transferring monitoring check results between systems. Implemented using XML over HTTPS.

NSCA
Nagios Service Check Acceptor — an older data transport mechanism for submitting monitoring check results to a monitoring server. Uses a non-standard network port and protocol.

plugin
A program or script that performs a monitoring check, and provides the results to the calling program, through a message and exit code. The plugin style and definition comes from Nagios, and is a de-facto industry standard.

Relay Token
Secret “password” used by an MBR and a cooperating monitoring server, to pass check results through the MRS.