RAYMANAGESOFT® UNIFIED ENDPOINT MANAGER

Unified Endpoint Management

Technical Documentation

RayManageSoft Unified Endpoint Manager
is part of RaySuite UEM

raynet

Content

Modern Endpoint Management

Companies are increasingly reaching their limits when it comes to managing software or operating systems. As times change, modern endpoints are becoming constant companions for employees in companies. According to the motto Bring Your Own Device, private devices are increasingly being used in companies. This presents companies with major challenges in terms of data security.

To protect company data from unauthorized access, data traffic from the company to the device, as well as data located directly on the device, must be protected.

RayManageSoft UEM provides a centralized platform for the installation of the company's individual applications on various endpoints. New apps can be installed on all devices in real time with just one click. Through our management console, enterprise apps are installed in no time or unwanted apps are blocked via whitelisting or blacklisting.

Scope of functions

The following management solutions are now united under the name **RayManageSoft Unified Endpoint Manager**:

Architecture

- Cloud based or on-premises model (operated in your infrastructure)
- Also available as virtual appliance
- Encrypted communication between client and server
- Bidirectional communication between client and server (Push and Pull)
- LDAP integration for selection of the current structure
- Own certificate server

Console

- Policy change per drag / drop
- Multi-tenant capable architecture
- Role and responsibility dependent administration console
- Self-service portal for end users
- Intuitive dashboard for first level support
- Revision secure recording
- Meaningful reports and analyses
- Ticket system/Workflow for desired changes
- Group/user/device-specific permission
- Web based management console

Hardware & Software inventory and Asset Management

- Inventory of the devices in the company
- Decentral and central scanning approach as well as various scanning methods (agent-based & agentless: zero touch, remote, portable, etc.)

Expense Management

- Management of contract information for the devices

Software/Application Management

- Software deployment
- Deployment based on desired policy with built in self-healing and network optimizations
- Installing and deleting of Apps – "over the air"
- Distributing of company applications via Push
- Enterprise App Store
- Black and whitelisting of applications
- App Wrapper
- Kiosk Mode

Connection Management

- Configuration of guidelines for WiFi, VPN, and APN

PIM Management

- Configuration of guidelines for email, Exchange Active Sync, CalDav, Subscribed Calendars, and WebClips
- Secure PIM Container
- Secure email Gateway for Exchange

Vulnerability Management

- Software catalog to identify and optimize software portfolio. The existing data can be augmented with functional and technical information including compatibility and software vulnerabilities
- Daily updated vulnerability information to eliminate security gaps

Security Management

- Central enforcement of complex passwords
- Encrypted messaging container for email, calendar and contacts (BYOD)
- Secure web browser
- URL-whitelisting
- Dual persona – Full separation of work and personal data
- Samsung SAFE Integration
- Encryption of memory card and device memory
- Encryption of communication between server and client
- Blocking of camera function in the company
- Prohibition of performing certain applications
- Central certificate management for mobile devices
- Blocking of mobile devices from the central console
- Central deletion of complex data or subareas
- Location of mobile devices in case of theft
- Sending of a message to the finder
- Establishing guidelines for limiting device and application settings
- Blocking of data synchronization to Apple iCloud
- Blocking of jailbreak or rooted devices
- Blocking of devices after a pre-determined inactive time period
- Encrypted communication of devices

Content Management

- Provisioning of corporate data – Secure content box

Data Management

- All data in one central solution: Collection data from different sources to create key figures and actions. Integrated ETL processes with easy-to-use web-based editors
- Managed connectors to relevant ecosystems (SAM, CMDB, support, enduser request management, identity, security, remote access, …)
- Integration of different data sources through standard and SaaS collectors (e.g. Active Directory, VMware, RayVentory, SCCM, Office 365, AWS, Azure, …)
- Flexible custom collectors (e.g. inhouse software, …)
- Out of the box reports for complete transparency, e.g. devices, deployment status, users, groups, assignments, …

Package Editor

- Integrated Package Store: Access to pre-configured and quality-assured software packages for immediate use
- Package editor: for easy editing of software applications

Prerequisites

RayManageSoft Unified Endpoint Manager requires a cloud storage to store all uploaded package files and to make them available to all devices. In this version cloud storage backed by Azure infrastructure is supported.

Be aware:
While is possible to install the product in both cloud- and on-premises environments, both approaches still require a valid Azure Cloud Storage for the files that get distributed to managed devices.

- Docker Images for RayManageSoft Unified Endpoint Manager. These must be provided by Raynet, as they are not contained within publicly available Docker registries.
- Docker for Windows (onsite installation)
- Microsoft SQL Server
    - The server must be reachable from the Docker environment.
- A network storage solution (Azure, MinIO, Amazon Web Services)

System requirements

Network Storage

The network storage is used by RayManageSoft Unified Endpoint Manager to store package files and distribute them to the client.

Currently the following storage options are supported:

- Azure
- Amazon Web Services (AWS) storage
- MinIO

With MinIO the files can be stored on a local system.

Azure

1. Create a new Azure Storage account.
    - Basics:
        - Set account kind to BlobStorage
        - Set blob access tier to Hot
    - Networking:
        - Set connectivity method to Public Endpoint
    - Advanced:
        - Disable Blob public access
2. Wait for the storage account to be set up
3. Open the details of your Storage account
4. Go to the settings/cors Section
5. Add a new CORS Entry and ensure the following configuration is used:
    - Allowed origins: *

(For POC and test, putting asterisk is OK. In production, make sure that the origin is set to the URL under which your RayManageSoft Unified Endpoint Manager will be hosted).

- Allowed methods:
  DELETE, GET, HEAD, MERGE, POST, OPTIONS, PUT, PATCH
- Allowed headers: *
- Exposed headers: *
- Max age: 0

6. The Cloud Storage should be ready to use

AWS

To use the AWS storage, create a new user in the AWS subscription. The user needs full access to the Amazon S3 Resource. After the creation of the user, an AccessKey and a Secret Key are displayed. Save those values as they are required during the setup of RayManageSoft Unified Endpoint Manager.

Best practice:
Raynet recommends turning on the Block Public Access settings for account for the S3 account.

For further instructions regarding these settings please refer to https://docs.aws.amazon.com/AmazonS3/latest/userguide/configuring-block-public-access-account.html.

MinIo

MinIO is an open-source object storage which supports storing files in the cloud or on a local file system. MinIO can be hosted on multiple platforms.

Best practice:
For easy installation, Raynet recommends the usage of the MinIO Quickstart Guide.

The configured username and password will later be required by RayManageSoft Unified Endpoint Manager to connect to the MinIO server.

Device Management

Supported device operating systems:

- Apple iOS 11.0 or higher*
- Apple macOS 10.11 or higher
- Google Android 7.0 or higher**
- MS Windows 10 or higher (Desktop-Computer, Notebook, and Tablet)
- Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2 Core, Windows Server 2008 R2, Windows Server 2008 Core, Windows Server 2008, Windows Server 2008 Core x64, Windows Server 2008 x64, Windows Server 2003, Windows Server 2003 x64
- Windows 10, Windows 10 x64, Windows 8.1, Windows 8.1 x64, Windows 8, Windows 8 x64, Windows 7, Windows 7 x64, Windows Vista, Windows Vista x64, Windows XP, Windows XP x64, Windows 2003 R2, Windows 2003 R2 x64,

- RedHat Linux 8 and 9, RedHat Enterprise Linux 3, 4, 5, 6,6.1
- SuSE Professional/OpenSuSE 9, 10,11
- SuSE Enterprise Server (SLES) 9, 10,11
- Mac OS X 10.8, 10.9, 10.10
- Solaris 9, 10, 11 (Intel)2, Solaris 8, 9, 10, 11 (SPARC)2
- CentOS 6.x, 7.x2
- Fedora 212AIX 5.2, 5.3, 6.1, 7.12
- HP-UX 11.00, 11i, 11i v2, 11iv32

**Please note that devices with iOS 10 or earlier cannot be enrolled due to drastic changes made by apple in the enrollment process.*
***Please note that older devices can still get enrolled but will only be supported partially if you encounter a problem.*

We highly recommend using an OS version which is still supported by the manufacturer. Not only for compatibility but also for security reasons. Therefore, we recommend iOS 12 or higher and Android 9 or higher.

Supported LDAP Directories:

- Microsoft Active Directory
- Open LDAP

Installation

Installing on Azure

1. Create an SQL Server
   Create a simple MS SQL Server using the Azure Portal or use an existing MS SQL Server which is accessible over the internet.

2. Create the database
   Depending on your choice, either create a new database using the Azure portal or create a new Database on existing MS SQL Server (for example using Microsoft SQL Server Management Studio).

3. Prepare and validate connection strings Copy connection string to your SQL Server, either from the Azure portal or use the correct format of connection string for local SQL databases. Ensure that the connection string is valid.

4. Install RayManageSoft Unified Endpoint Manager Backend
   Create a new container instance, using the following suggested parameters:

Basics:

- Image source: Docker Hub or another registry
- Image Type: Private
- Image: `raynetnightly.azurecr.io/raynet/raymanagesoftcloud/`
- `rmsc_frontend:insider`

- The image name was provided to you with the docker credentials and should be adjusted accordingly
- Image registry login server: `raynetnightly.azurecr.io`
- Image registry username: `yourUser`
- Image registry password: `yourPassword`
- OS type: Windows

Networking:

- Networking type: Public
- DNS name label: `yourDnsName`
- Ports: `80 TCP`

Advanced

- Restart Policy: Always
- Environment variables
  - o SystemDb: `yourConnectionString`
  - o ResultDb: `yourConnectionString`

Once the container is up and running, make sure to note the DNS name of the instance. You will need this value in the next step.

5. Install RayManageSoft Unified Endpoint Manager Web UI
   Create a new container instance, using the following suggested parameters:

Basics:

- Image source: Docker Hub or another registry
- Image Type: Private
- Image: `raynetnightly.azurecr.io/raynet/raymanagesoftcloud/`
- `rmsc_frontend:insider`
- The image name was provided to you with the docker credentials and should be adjusted accordingly
- Image registry login server: `raynetnightly.azurecr.io`
- Image registry username: `yourUser`
- Image registry password: `yourPassword`
- OS type: Windows

Networking:

- Networking type: Public
- DNS name label: `yourDnsName`
- Ports: `80 TCP`

Advanced:

- Restart Policy: Always
- Environment variables
  - `SystemDb`: `yourConnectionString`
  - `ResultDb`: `yourConnectionString`
- `BackendEndpoint`: `yourBackendDnsName`

This should be the DNS name of your backend component.
- `BackendPort`: `80`
- For a production environment a more advanced setup using the 443 Port and HTTPS is highly recommended.
- `BackendProtocol`: `http`

The following parameter depend on the chosen storage hoster:

Azure
- `DefaultHoster`: Azure
- `AzureStorageEndpoint`: `yourStorageEndpoint`
  This is the connection string property of Azure Storage. It can be found in the Azure Porta l > Stora ge Accounts > Account Deta ils > Settings > Access Keys > Connection String (key1 or key2).
- `AzureEndpointUrl`: `yourStorageEndpointUrl`
  This is the primary endpoint property of your Azure Storage. It can be found in the Azure Porta l > Stora ge Accounts > Account Deta ils > Settings > Access Keys > Primary Endpoint.
- `AzureTokenTimeout`: `60`

AWS
- `DefaultHoster`: AWS
- `AwsAccessKey`=T1SS4CC322KEYYO83C3V
  This is the access key received during the setup of the AWS IAM user.
- `AwsSecretKey`=exa+tfekKDsresRuBJ65forasecr3TK3ythATYIU
  This is the secret key received during the setup of the AWS IAM user.
- `AwsRegion`=eu-central-1
  This is the region which should be used to host the storage. A full list of the regions can be found here: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.RegionsAndAvailabilityZones.html.

MinIO
- `DefaultHoster`: MinIO
- `MinIOEndpoint`=yourMinIOEndpoint (e.g. play.min.io:80)
  The endpoint of the used MinIO instance (ip:port) or (fqdn:port)
- `MinIOAccessKey`=yourMinIOAccessKey
  The access key/user that has been configured during the MinIO setup.
- `MinIOSecretKey`=yourMinIOSecretKey
  The secret key/password that has been configured during the MinIO setup.
  MinIOSSL=true
  A boolean value indictating whether the MinIO server requires/uses a https connection or not (the usage of an https connection is recommended).

Installing on-premises

1. Create a SQL Server

Set-up a new MS SQL Server or use an existing MS SQL Server which is accessible from the hosting environment.

2. Create the database
   Create a new database on existing MS SQL Server (for example using Microsoft SQL Server Management Studio).

3. Prepare and validate connection strings
   Copy connection string to your SQL Server. Ensure that the connection string is valid.

4. Install container images
   The installation on on-premises environment is straightforward with the usage of compose file, which requires only minimal adjustment.

   docker-compose.yml
   The file has the following content:

```
version: "3.7"
services:
frontend:
image:
raynetnightly.azurecr.io/raynet/raymanagesoftcloud/rmsc_fr
ontend:ports:
- "80:80"
restart: always
env_file:
- env_frontend.list
backend:
image:
raynetnightly.azurecr.io/raynet/raymanagesoftcloud/rmsc_ba
ckend:depends_on:
- frontend
ports:
- "8080:80"
restart: always
env_file:
- env_backend.list
```

   Additionally, you need two text files with environment variables:

   env_frontend.list

   The file has the following content:

```
SystemDb="SQLConnectionString"
ResultDb="SQLReportConnectionString"
BackendEndpoint="publiclyReachableDNS"
BackendPort="8080"
BackendProtocol="http"
```

The following parameters depend on the chosen storage hoster:

Azure

- `DefaultHoster`: Azure
- `AzureStorageEndpoint`: `yourStorageEndpoint`
  This is the connection string property of Azure Storage. It can be found in the Azure Porta l > Stora ge Accounts > Account Deta ils > Settings > Access Keys > Connection String (key1 or key2).
- `AzureEndpointUrl`: `yourStorageEndpointUrl`
  This is the primary endpoint property of your Azure Storage. It can be found in the Azure Porta l > Stora ge Accounts > Account Deta ils > Settings > Access Keys > Prima ry Endpoint.
- `AzureTokenTimeout`: `60`

AWS

- `DefaultHoster`: AWS
- `AwsAccessKey`=T1SS4CC322KEYYO83C3V
  This is the access key received during the setup of the AWS IAM user.
- `AwsSecretKey`=exa+tfekKDsresRuBJ65forasecr3TK3ythATYIU
  This is the secret key received during the setup of the AWS IAM user.
- `AwsRegion`=eu-central-1
  This is the region which should be used to host the storage.
- A full list of the regions can be found here:
  https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.RegionsAndAvailabilityZones.html.

MinIO

- `DefaultHoster`: MinIO
- `MinIOEndpoint`=yourMinIOEndpoint (e.g. play.min.io:80)
  The endpoint of the used MinIO instance (ip:port) or (fqdn:port)
- `MinIOAccessKey`=yourMinIOAccessKey
  The access key/user that has been configured during the MinIO setup.
- `MinIOSecretKey`=yourMinIOSecretKey
  The secret key/password that has been configured during the MinIO setup.
- MinIOSSL=true
  A boolean value indictating whether the MinIO server requires/uses a https connection or not (the usage of an https connection is recommended).
  env_backend.list
  The file has the following content:
  `SystemDb="SQLConnectionString"`
  `ResultDb="SQLReportConnectionString"`

Save all files in the same folder, so that you have the following content:

- `docker-compose.yml`
- `env_frontend.list`
- `env_backend.list`

Adjust the values accordingly, paying attention to connection strings and Azure Storage credentials.
Once all three files are ready, open PowerShell window or a terminal of your choice, navigate to the folder where three files exist and execute the following command:

```
docker login -u <user> -p <password>
aynetnightly.azurecr.io
```

This will login to private Raynet Docker registry. The credentials will be provided by Raynet, if you do not have them yet ask your administrator or contact us. Then, ensure no container is running:

```
docker-compose down
```

Ensure you have the newest version of the image:

```
docker-compose pull
```

Start all required containers and let them run in the background (deamon)

```
docker-compose up -d
```

Finally, sign-out from Docker repository:

```
docker logout
```

You can repeat these steps in future to perform update of your instances with a minimal downtime.