

Manual de Instalação e Configuração do Sistema de Monitoramento Oberon (SMO-Oberon)

1. Introdução

O Sistema de Monitoramento Oberon (SMO-Oberon) é uma solução robusta e multiplataforma desenvolvida para a coleta contínua de métricas e logs de sistema. O SMO-Oberon é essencial para a manutenção da integridade operacional, permitindo a detecção proativa de anomalias e incidentes, com integração direta ao sistema de gerenciamento de incidentes **Jira** para uma resposta rápida e documentada.

Este manual detalha o procedimento de instalação e configuração do agente de monitoramento em ambientes **Windows**, **Linux** e **macOS**.

2. Pré-requisitos do Sistema

Requisito	Windows	Linux	macOS
Java	JRE (Java Runtime Environment) versão 8 ou superior (Necessário para o módulo .jar).	JRE (Java Runtime Environment) versão 8 ou superior.	JRE (Java Runtime Environment) versão 8 ou superior.
Python	Não aplicável (Usa o .exe compilado).	Python 3.x instalado.	Python 3.x instalado.
Permissões	Permissões de Administrador para instalação e execução.	Permissões de sudo ou root para instalação de dependências e permissão de execução.	Permissões de Administrador e permissão de execução.
Estrutura de Dire	Diretório <i>Home</i> (C:\Users\<Usuário>)	Diretório <i>Home</i> (/home/<usuário>)	Diretório <i>Home</i> (/Users/<usuário>)

tó o			
---------	--	--	--

3. Localização do MAC Address (Identificação do Agente)

O MAC Address (Media Access Control Address) é crucial para identificar unicamente cada agente no sistema central Oberon.

3.1. Windows

1. Abra o **Prompt de Comando** (cmd) ou **PowerShell**.
2. Execute o comando: `ipconfig /all`
3. Procure pela seção "**Adaptador Ethernet**" ou "**Adaptador de Rede Sem Fio**".
4. O endereço procurado é o valor listado em "**Endereço Físico**" (formato: XX-XX-XX-XX-XX-XX).

3.2. Linux

1. Abra o **Terminal**.
2. Execute o comando: `ip a` ou `ifconfig` (se instalado).
3. Procure pela interface de rede principal (geralmente `eth0`, `ensXX` ou `wlan0`).
4. O endereço procurado é o valor listado após **ether** ou **HWaddr** (formato: XX:XX:XX:XX:XX:XX).

3.3. macOS

1. Abra o **Terminal**.
2. Execute o comando: `ifconfig` ou `networksetup -listallhardwareports`.
3. Procure pela interface de rede (geralmente `en0` ou `en1`).
4. O endereço procurado é o valor listado após **ether** (formato: XX:XX:XX:XX:XX:XX).

IMPORTANTE: O MAC Address coletado deve ser submetido ao sistema central Oberon para **Cadastro Inicial do Agente** antes da primeira execução. O sistema central irá autorizar o monitoramento com base neste identificador.

4. Procedimento de Instalação por Sistema Operacional

O procedimento padrão para todos os sistemas é **baixar os arquivos do agente na pasta *Home*** do usuário.

4.1. Instalação em Windows

O agente em Windows utiliza o executável compilado (.exe) e o módulo Java (.jar).

1. **Download:** Baixe os arquivos do agente (oberon_monitor.exe e oberon_monitor.jar) e salve-os diretamente na pasta *Home* (C:\Users\<Usuário>).
2. **Configuração (MAC Address):** O sistema Windows é o único que suporta a execução do .exe do Python. Este executável é desenhado para detectar e utilizar o MAC Address automaticamente.
3. **Execução do Módulo Python/EXE:**
 - a. Abra o **Prompt de Comando** como **Administrador**.
 - b. Navegue para a pasta *Home* (se necessário).
 - c. Execute: .\oberon_monitor.exe
4. **Execução do Módulo Java:**
 - a. No mesmo **Prompt de Comando** (ou em um novo), execute: java -jar oberon_monitor.jar
5. **Verificação:** Mantenha os consoles abertos. Se o MAC Address foi cadastrado, ambos os programas devem iniciar o *logging* e o envio de dados.

4.2. Instalação em Linux

O agente em Linux utiliza o script Python (.py) e o módulo Java (.jar).

1. **Download:** Baixe os arquivos do agente (oberon_monitor.py e oberon_monitor.jar) e salve-os diretamente na pasta *Home* (/home/<usuário>).
2. **Permissão de Execução (Python):** No Terminal, conceda a permissão de execução ao script Python:

```
Bash
chmod +x oberon_monitor.py
```

3. Execução do Módulo Python:

- a. Execute o script usando o interpretador Python 3:

```
Bash
python3 oberon_monitor.py
```

- b. O script solicitará o MAC Address durante a primeira execução, se a detecção automática falhar. Use o endereço encontrado na Seção 3.2.

4. Execução do Módulo Java:

- a. Em um novo Terminal, execute:

```
Bash
java -jar oberon_monitor.jar
```

5. **Verificação:** Ambos os *scripts* devem iniciar a coleta de dados e enviar logs, confirmando a comunicação com o sistema central.

4.3. Instalação em macOS

O procedimento é similar ao Linux, utilizando o script Python (.py) e o módulo Java (.jar).

1. **Download:** Baixe os arquivos do agente (oberon_monitor.py e oberon_monitor.jar) e salve-os diretamente na pasta *Home* (/Users/<usuário>).
2. **Permissão de Execução (Python):** No Terminal, conceda a permissão de execução:

```
Bash
chmod +x oberon_monitor.py
```

3. Execução do Módulo Python:

- a. Execute o script:

```
Bash
python3 oberon_monitor.py
```

- b. **Nota:** Se o sistema macOS apresentar um alerta de segurança, vá para **Preferências do Sistema > Segurança e Privacidade** e autorize a execução do script.

4. Execução do Módulo Java:

- a. Em um novo Terminal, execute:

```
Bash
java -jar oberon_monitor.jar
```

5. **Verificação:** Monitore a saída do Terminal para garantir que ambos os programas estão ativos e reportando o status.

5. Gerenciamento de Logs e Resposta a Incidentes (Jira)

Um dos pilares do SMO-Oberon é a integração automatizada e manual com o sistema de gerenciamento de incidentes **Jira**, garantindo a rastreabilidade e a resolução eficiente de problemas.

5.1. Geração e Armazenamento de Logs

Ambos os programas de monitoramento (.jar e .py/.exe) são projetados para gerar logs detalhados em tempo real.

- **Logs Operacionais:** Logs em tempo real são enviados para o *dashboard* central do Oberon.
- **Logs Locais (Backup):** Uma cópia dos logs é armazenada localmente na pasta `~/oberon/logs/` (criada automaticamente na Home) para análise *post-mortem*. Estes logs incluem métricas de desempenho, erros de comunicação e rastreamento de execução.

5.2. Protocolo de Incidente - Integração Jira

Em caso de detecção de uma anomalia grave (por exemplo, uso de CPU > 95% por mais de 5 minutos, falha crítica de I/O, ou desconexão prolongada), o protocolo a seguir será ativado:

1. **Detecção e Alerta:** O sistema central Oberon detecta o incidente com base nos dados do MAC Address cadastrado.
2. **Criação Automática de Ticket:** Um novo *ticket* é criado automaticamente no projeto **INCIDENTES** do Jira (exemplo: INC-1234), com as seguintes informações:
 - a. **Resumo:** [OBERON ALERTA] - Falha Crítica de Desempenho no Agente [MAC ADDRESS]
 - b. **Descrição:** Detalhes da métrica violada, horário de início, e nome do *host* associado.
 - c. **Etiquetas:** SMO-OBERON, CRITICAL, [MAC_ADDRESS].
3. **Anexação de Logs:** O sistema central Oberon anexa automaticamente os últimos logs operacionais do agente ao *ticket* do Jira.
4. **Análise e Resolução:**
 - a. O time de Suporte/Infraestrutura é notificado e assume o *ticket*.
 - b. Em caso de falha persistente ou necessidade de análise aprofundada, a equipe de suporte deve acessar os **Logs Locais** (`~/oberon/logs/`) do sistema afetado.

- c. Os logs locais (.log files) devem ser **compactados e anexados manualmente** ao *ticket* correspondente no Jira para análise completa pela engenharia.
- d. Toda a comunicação, análise de causa raiz e ações de remediação devem ser registradas como comentários no *ticket* do Jira até a sua **Resolução e Fechamento**.

Este fluxo garante que todos os dados de sistema gerados pelos programas de monitoramento estejam diretamente ligados ao processo formal de gestão de incidentes, assegurando conformidade e transparência.