

GetHomepage Integration Guide

Overview

PatchMon provides a seamless integration with [GetHomepage](#) (formerly Homepage), allowing you to display real-time PatchMon statistics in your GetHomepage dashboard. This integration uses authenticated API endpoints to securely fetch and display your patch management data.

Features

Default Widget Display

By default, the GetHomepage widget displays:

- **Total Hosts** - Number of active monitored hosts
- **Hosts Needing Updates** - Hosts with outdated packages
- **Security Updates** - Number of security-related updates available

Additional Available Data

The API provides additional metrics that you can display by customizing the widget mappings:

- **Up-to-Date Hosts** - Hosts with no pending updates
- **Total Outdated Packages** - Aggregate count of packages needing updates
- **Total Repositories** - Number of active repositories
- **Recent Updates (24h)** - Update activity in the last 24 hours
- **Hosts with Security Updates** - Number of hosts requiring security patches
- **OS Distribution** - Breakdown of operating systems across hosts (returned in API but requires custom formatting)

Prerequisites

- PatchMon instance running and accessible
- GetHomepage installed and configured

- Network access from GetHomepage to PatchMon
- HTTPS recommended (but HTTP works with fallback clipboard)

Setup Instructions

Step 1: Create an API Key in PatchMon

1. **Log in to PatchMon** as an administrator
2. Navigate to **Settings → Integrations**
3. Click on the **GetHomepage** tab
4. Click **"New API Key"** button
5. Fill in the token details:
 - **Token Name:** A descriptive name (e.g., "GetHomepage Widget")
 - **Allowed IP Addresses** (Optional): Restrict access to specific IPs
 - **Expiration Date** (Optional): Set an expiration if needed
6. Click **"Create Token"**

Step 2: Save Your Credentials

After creating the token, you'll see a success modal with:

- **Token Key:** Your API username
- **Token Secret:** Your API password (shown only once!)
- **Base64 Encoded Credentials:** Pre-encoded for convenience
- **Complete Widget Configuration:** Ready-to-use YAML

⚠ **Important:** Save the token secret immediately. You won't be able to view it again!

Step 3: Configure GetHomepage

Method A: Copy Complete Configuration (Recommended)

1. In the PatchMon success modal, click **"Copy Config"** button
2. Open your GetHomepage `services.yml` file
3. Paste the copied configuration
4. Save the file
5. Restart GetHomepage

The default configuration displays 3 key metrics:

```
- PatchMon:
  href: http://your-patchmon-url:3000
```

```
description: PatchMon Statistics
icon: http://your-patchmon-url:3000/assets/favicon.svg
widget:
  type: customapi
  url: http://your-patchmon-url:3000/api/v1/gethomepage/stats
  headers:
    Authorization: Basic <base64_encoded_credentials>
  mappings:
    - field: total_hosts
      label: Total Hosts
    - field: hosts_needing_updates
      label: Needs Updates
    - field: security_updates
      label: Security Updates
```

“ **Note:** You can add more fields to the `mappings` section. See [Configuration Options](#) below for all available fields.

Method B: Manual Configuration

If you need to manually create the base64 credentials:

1. Encode your credentials:

```
echo -n "YOUR_API_KEY:YOUR_API_SECRET" | base64
```

2. Create the widget configuration in `services.yml`:

```
- PatchMon:
  href: http://your-patchmon-url:3000
  description: PatchMon Statistics
  icon: http://your-patchmon-url:3000/assets/favicon.svg
  widget:
    type: customapi
    url: http://your-patchmon-url:3000/api/v1/gethomepage/stats
    headers:
      Authorization: Basic <your_base64_credentials>
    mappings:
      - field: total_hosts
```

```
    label: Total Hosts
- field: hosts_needing_updates
    label: Needs Updates
- field: security_updates
    label: Security Updates
```

3. Restart GetHomepage:

```
docker restart gethomepage
# or
systemctl restart gethomepage
```

Configuration Options

Widget Mappings

The default widget configuration displays **3 metrics**: Total Hosts, Hosts Needing Updates, and Security Updates.

You can customize which statistics to display by adding or removing fields in the `mappings` section. The API provides **8 numeric metrics** you can choose from.

How to Customize Mappings

1. **Locate the `mappings` section** in your GetHomepage `services.yml`
2. **Add or remove field entries** - each entry has two parts:
 - `field:` - The metric name from the API (see table below)
 - `label:` - How it appears in GetHomepage (customize as you like)
3. **You can display up to ~6-8 metrics** before the widget becomes crowded
4. **Save and restart GetHomepage** to see changes

Available Fields

Field	Description	Default
<code>total_hosts</code>	Total number of active hosts	☒ Yes
<code>hosts_needing_updates</code>	Hosts with outdated packages	☒ Yes
<code>security_updates</code>	Number of security updates available	☒ Yes
<code>up_to_date_hosts</code>	Hosts with no pending updates	☐ No
<code>total_outdated_packages</code>	Total outdated packages across all hosts	☐ No

Field	Description	Default
<code>hosts_with_security_updates</code>	Hosts requiring security updates	☐ No
<code>total_repos</code>	Number of active repositories	☐ No
<code>recent_updates_24h</code>	Successful updates in last 24 hours	☐ No
<code>top_os_1_count</code>	Count of most common OS (e.g., "Ubuntu: 20")	☐ No
<code>top_os_2_count</code>	Count of 2nd most common OS	☐ No
<code>top_os_3_count</code>	Count of 3rd most common OS	☐ No

“ **Note:** Fields marked with ☐ are available but not included in the default configuration. Add them to your `mappings` section to display them.

OS Distribution: The API also returns `top_os_1_name`, `top_os_2_name`, and `top_os_3_name` (strings like "Ubuntu", "Debian", "Rocky Linux"). However, GetHomepage widgets display these awkwardly. It's better to use just the count fields with custom labels that include the OS name (see examples below).

Quick Start: Adding a Metric

Example: Add "Recent Updates (24h)" to your widget

Before (Default - 3 metrics):

```
mappings:
  - field: total_hosts
    label: Total Hosts
  - field: hosts_needing_updates
    label: Needs Updates
  - field: security_updates
    label: Security Updates
```

After (Custom - 4 metrics):

```
mappings:
  - field: total_hosts
    label: Total Hosts
  - field: hosts_needing_updates
    label: Needs Updates
  - field: security_updates
```

```
label: Security Updates
- field: recent_updates_24h      # ← Added this line
label: Updated (24h)             # ← And this line
```

Result: Your widget now shows 4 metrics including recent update activity.

You can add any combination of the 8 available fields. Just ensure the `field:` name matches exactly as shown in the table above.

Advanced Mapping Examples

Example: Security-Focused Widget

Shows security-critical metrics only:

```
widget:
  type: customapi
  url: http://your-patchmon-url:3000/api/v1/gethomepage/stats
  headers:
    Authorization: Basic <credentials>
  mappings:
    - field: security_updates
      label: Security Patches
    - field: hosts_with_security_updates
      label: Hosts at Risk
    - field: hosts_needing_updates
      label: Total Pending
```

Example: Repository Management Widget

Focus on repository and host counts:

```
widget:
  type: customapi
  url: http://your-patchmon-url:3000/api/v1/gethomepage/stats
  headers:
    Authorization: Basic <credentials>
  mappings:
    - field: total_repos
      label: Repositories
```

- field: total_hosts
label: Managed Hosts
- field: up_to_date_hosts
label: Up-to-Date

Example: Activity Monitoring Widget

Track recent update activity:

```
widget:  
  type: customapi  
  url: http://your-patchmon-url:3000/api/v1/gethomepage/stats  
  headers:  
    Authorization: Basic <credentials>  
  mappings:  
    - field: recent_updates_24h  
      label: Updated (24h)  
    - field: hosts_needing_updates  
      label: Pending Updates  
    - field: up_to_date_hosts  
      label: Fully Patched
```

Example: Package-Focused Widget

Monitor outdated packages:

```
widget:  
  type: customapi  
  url: http://your-patchmon-url:3000/api/v1/gethomepage/stats  
  headers:  
    Authorization: Basic <credentials>  
  mappings:  
    - field: total_outdated_packages  
      label: Outdated Packages  
    - field: security_updates  
      label: Security Updates  
    - field: hosts_needing_updates  
      label: Affected Hosts
```

Example: OS Distribution Widget

Show your infrastructure breakdown by operating system:

```
widget:
  type: customapi
  url: http://your-patchmon-url:3000/api/v1/gethomepage/stats
  headers:
    Authorization: Basic <credentials>
  mappings:
    - field: total_hosts
      label: Total Hosts
    - field: top_os_1_count
      label: Ubuntu Hosts      # Customize these labels based on your actual OS mix
    - field: top_os_2_count
      label: Debian Hosts
    - field: top_os_3_count
      label: Rocky Linux Hosts
```

“ **Pro Tip:** First test the endpoint with `curl` to see what your actual top 3 operating systems are, then customize the labels accordingly. The API returns the OS names in `top_os_1_name`, `top_os_2_name`, and `top_os_3_name`.

Custom Icon

By default, the widget uses PatchMon's favicon. You can customize it:

```
# Use PatchMon's dark logo
icon: http://your-patchmon-url:3000/assets/logo_dark.png

# Use PatchMon's light logo
icon: http://your-patchmon-url:3000/assets/logo_light.png

# Use GetHomepage's built-in icons
icon: server

# Use a local icon in GetHomepage
icon: /icons/patchmon.png
```

API Endpoint Details

Endpoint

```
GET /api/v1/gethomepage/stats
```

Authentication

- **Type:** HTTP Basic Authentication
- **Format:** `Authorization: Basic <base64(key:secret)>`

Response Format

The endpoint returns JSON with the following structure:

```
{
  "total_hosts": 42,
  "total_outdated_packages": 156,
  "total_repos": 12,
  "hosts_needing_updates": 15,
  "up_to_date_hosts": 27,
  "security_updates": 23,
  "hosts_with_security_updates": 8,
  "recent_updates_24h": 34,
  "os_distribution": [
    { "name": "Ubuntu", "count": 20 },
    { "name": "Debian", "count": 12 },
    { "name": "Rocky Linux", "count": 10 }
  ],
  "top_os_1_name": "Ubuntu",
  "top_os_1_count": 20,
  "top_os_2_name": "Debian",
  "top_os_2_count": 12,
  "top_os_3_name": "Rocky Linux",
  "top_os_3_count": 10,
  "last_updated": "2025-10-11T12:34:56.789Z"
}
```

All Available Metrics Explained

All numeric fields can be used in GetHomepage mappings:

Field	Type	Description	Use Case
<code>total_hosts</code>	Number	Total active hosts in PatchMon	Overall infrastructure size
<code>hosts_needing_updates</code>	Number	Hosts with at least one outdated package	Hosts requiring attention
<code>up_to_date_hosts</code>	Number	Hosts with zero outdated packages	Compliant/healthy hosts
<code>security_updates</code>	Number	Total security updates available across all hosts	Critical patches needed
<code>hosts_with_security_updates</code>	Number	Hosts requiring security patches	High-risk hosts
<code>total_outdated_packages</code>	Number	Sum of all outdated packages	Total patching workload
<code>total_repos</code>	Number	Active repositories being monitored	Repository coverage
<code>recent_updates_24h</code>	Number	Successful updates in last 24 hours	Recent patching activity
<code>top_os_1_name</code>	String	Name of most common OS	OS breakdown
<code>top_os_1_count</code>	Number	Count of most common OS	OS breakdown
<code>top_os_2_name</code>	String	Name of 2nd most common OS	OS breakdown
<code>top_os_2_count</code>	Number	Count of 2nd most common OS	OS breakdown
<code>top_os_3_name</code>	String	Name of 3rd most common OS	OS breakdown
<code>top_os_3_count</code>	Number	Count of 3rd most common OS	OS breakdown
<code>os_distribution</code>	Array	Full breakdown of OS types (for advanced use)	Infrastructure composition
<code>last_updated</code>	String (ISO 8601)	Timestamp of when stats were generated	Data freshness

“ **Note:** The API provides top 3 OS distribution data as flat fields (`top_os_*`) that can be easily displayed in GetHomepage widgets. The full `os_distribution` array is also available for custom integrations.

Health Check Endpoint

```
GET /api/v1/gethomepage/health
```

Returns basic health status and API key name.

Managing API Keys

View Existing Keys

1. Go to **Settings** → **Integrations** → **GetHomepage**
2. View all created API keys with:
 - Token name
 - Creation date
 - Last used timestamp
 - Active/Inactive status
 - Expiration date (if set)

Disable/Enable Keys

Click the "**Disable**" or "**Enable**" button on any API key to toggle its status.

Delete Keys

Click the **trash icon** to permanently delete an API key. This action cannot be undone.

Security Features

- **IP Restrictions:** Limit API key usage to specific IP addresses
- **Expiration Dates:** Set automatic expiration for temporary access
- **Last Used Tracking:** Monitor when keys are being used
- **One-Time Secret Display:** Token secrets are only shown once at creation

Troubleshooting

Error: "Missing or invalid authorization header"

Cause: GetHomepage isn't sending the Authorization header correctly.

Solution:

1. Verify the `headers:` section is properly indented in `services.yml`
2. Ensure base64 credentials are correctly encoded
3. Check for extra spaces or line breaks in the configuration
4. Verify you're using `type: customapi` (not another widget type)

Error: "Invalid API key"

Cause: The API key doesn't exist or was deleted.

Solution:

1. Verify the API key exists in PatchMon (Settings → Integrations)
2. Create a new API key if needed
3. Update GetHomepage configuration with new credentials

Error: "API key is disabled"

Cause: The API key has been disabled in PatchMon.

Solution:

1. Go to Settings → Integrations → GetHomepage
2. Click "**Enable**" on the API key

Error: "API key has expired"

Cause: The API key has passed its expiration date.

Solution:

1. Create a new API key without expiration
2. Or create a new key with a future expiration date
3. Update GetHomepage configuration

Error: "IP address not allowed"

Cause: GetHomepage's IP address is not in the allowed list.

Solution:

1. Check GetHomepage's IP address
2. Update the API key's allowed IP ranges in PatchMon
3. Or remove IP restrictions if not needed

Widget Not Showing Data

Checklist:

- ☐ GetHomepage can reach PatchMon URL (test with `curl`)
- ☐ API key is active and not expired
- ☐ Base64 credentials are correct
- ☐ `services.yml` syntax is valid YAML
- ☐ GetHomepage has been restarted after config changes
- ☐ Check GetHomepage logs for error messages

Testing the API Endpoint

Test the endpoint manually to see all available metrics:

```
# Step 1: Encode your credentials
echo -n "your_key:your_secret" | base64
# Output: eW91cl9rZXk6eW91cl9zZWNyZXQ=

# Step 2: Test the endpoint with your credentials
curl -H "Authorization: Basic YOUR_BASE64_CREDENTIALS" \
  http://your-patchmon-url:3000/api/v1/gethomepage/stats
```

Expected response: JSON with all 8 core metrics plus OS distribution:

```
{
  "total_hosts": 42,
  "hosts_needing_updates": 15,
  "security_updates": 23,
  "up_to_date_hosts": 27,
  "total_outdated_packages": 156,
  "hosts_with_security_updates": 8,
  "total_repos": 12,
  "recent_updates_24h": 34,
  "top_os_1_name": "Ubuntu",
  "top_os_1_count": 20,
  "top_os_2_name": "Debian",
  "top_os_2_count": 12,
  "top_os_3_name": "Rocky Linux",
```

```
"top_os_3_count": 10,  
"os_distribution": [...],  
"last_updated": "2025-10-11T12:34:56.789Z"  
}
```

Any of these numeric fields (including `top_os*_count`) can be used in your GetHomepage mappings!

To find out what your top 3 operating systems are, look for the `top_os_1_name`, `top_os_2_name`, and `top_os_3_name` values in the response.

Pretty Print for Easy Reading

Use `jq` to format the output nicely:

```
curl -H "Authorization: Basic YOUR_BASE64_CREDENTIALS" \  
http://your-patchmon-url:3000/api/v1/gethomepage/stats | jq
```

This makes it easier to see what metrics your instance provides.

How to Display Your OS Distribution

Step 1: Discover your top operating systems

Run the curl command and look for these fields:

```
curl -s -H "Authorization: Basic YOUR_BASE64_CREDENTIALS" \  
http://your-patchmon-url:3000/api/v1/gethomepage/stats | jq '{top_os_1_name, top_os_1_count,  
top_os_2_name, top_os_2_count, top_os_3_name, top_os_3_count}'
```

Example output:

```
{  
  "top_os_1_name": "Ubuntu",  
  "top_os_1_count": 35,  
  "top_os_2_name": "Debian",  
  "top_os_2_count": 18,  
  "top_os_3_name": "Rocky Linux",  
  "top_os_3_count": 12  
}
```

Step 2: Add to your GetHomepage widget

Use the count fields (`top_os_*_count`) and label them with your actual OS names:

```
mappings:
  - field: total_hosts
    label: Total Hosts
  - field: top_os_1_count
    label: Ubuntu          # Use your actual OS from top_os_1_name
  - field: top_os_2_count
    label: Debian          # Use your actual OS from top_os_2_name
  - field: top_os_3_count
    label: Rocky Linux     # Use your actual OS from top_os_3_name
```

Step 3: Restart GetHomepage

```
docker restart gethomepage
# or
systemctl restart gethomepage
```

Your widget will now show your infrastructure OS breakdown! ☑☑

Security Best Practices

1. **Use HTTPS:** Always use HTTPS in production for encrypted communication
2. **IP Restrictions:** Limit API key usage to GetHomepage's IP address
3. **Set Expiration:** Use expiration dates for temporary access
4. **Regular Rotation:** Rotate API keys periodically
5. **Monitor Usage:** Check "Last Used" timestamps for suspicious activity
6. **Unique Keys:** Create separate API keys for different GetHomepage instances
7. **Secure Storage:** Store GetHomepage `services.yml` securely with proper permissions

Complete Working Examples

Copy-Paste Ready Configurations

These are complete, working configurations you can copy directly into your `services.yml` file. Just replace the placeholders with your actual values.

Simple Dashboard Widget (Default)

This is the default configuration generated by PatchMon:

```
- PatchMon:
  href: https://patchmon.example.com
  description: Patch Management
  icon: https://patchmon.example.com/assets/favicon.svg
  widget:
    type: customapi
    url: https://patchmon.example.com/api/v1/gethomepage/stats
    headers:
      Authorization: Basic dXNlcjpwYXNzd29yZA==
    mappings:
      - field: total_hosts
        label: Total Hosts
      - field: hosts_needing_updates
        label: Needs Updates
      - field: security_updates
        label: Security Updates
```

Detailed Monitoring Widget (Custom)

This example shows how to display 4 metrics including recent activity:

```
- PatchMon Production:
  href: https://patchmon.example.com
  description: Production Environment Patches
  icon: https://patchmon.example.com/assets/logo_dark.png
  widget:
    type: customapi
    url: https://patchmon.example.com/api/v1/gethomepage/stats
    headers:
      Authorization: Basic dXNlcjpwYXNzd29yZA==
    mappings:
      - field: total_hosts
        label: Total Servers
      - field: hosts_needing_updates
        label: Needs Patching
      - field: security_updates
        label: Security Patches
```


- field: recent_updates_24h
- label: Patched Today

Multiple Environments (Custom)

This example shows different metrics for different environments:

```
# Production - Focus on security
- PatchMon Prod:
  href: https://patchmon-prod.example.com
  description: Production Patches
  icon: https://patchmon-prod.example.com/assets/favicon.svg
  widget:
    type: customapi
    url: https://patchmon-prod.example.com/api/v1/gethomepage/stats
    headers:
      Authorization: Basic <prod_credentials>
    mappings:
      - field: total_hosts
        label: Hosts
      - field: security_updates
        label: Security
      - field: hosts_needing_updates
        label: Pending

# Development - Focus on package count
- PatchMon Dev:
  href: https://patchmon-dev.example.com
  description: Development Patches
  icon: https://patchmon-dev.example.com/assets/favicon.svg
  widget:
    type: customapi
    url: https://patchmon-dev.example.com/api/v1/gethomepage/stats
    headers:
      Authorization: Basic <dev_credentials>
    mappings:
      - field: total_hosts
        label: Hosts
      - field: total_outdated_packages
```

```
    label: Packages
- field: up_to_date_hosts
    label: Updated
```

Maximum Information Widget (All 8 Metrics)

This example shows ALL available metrics (may be crowded):

```
- PatchMon Complete:
  href: https://patchmon.example.com
  description: Complete Statistics
  icon: https://patchmon.example.com/assets/favicon.svg
  widget:
    type: customapi
    url: https://patchmon.example.com/api/v1/gethomepage/stats
    headers:
      Authorization: Basic <credentials>
    mappings:
      - field: total_hosts
        label: Total Hosts
      - field: hosts_needing_updates
        label: Needs Updates
      - field: up_to_date_hosts
        label: Up-to-Date
      - field: security_updates
        label: Security Updates
      - field: hosts_with_security_updates
        label: Security Hosts
      - field: total_outdated_packages
        label: Outdated Packages
      - field: total_repos
        label: Repositories
      - field: recent_updates_24h
        label: Updated (24h)
```

“ **Note:** Displaying all 8 metrics may make the widget tall. Choose 3-5 metrics that are most relevant to your needs.

Integration Architecture



Rate Limiting

The GetHomepage integration endpoints are subject to PatchMon's general API rate limiting:

- Default: **100 requests per 15 minutes** per IP address

- GetHomepage typically polls every 60 seconds
- This allows for normal operation without hitting limits

Support and Resources

- **PatchMon Documentation:** <https://docs.patchmon.net>
- **GetHomepage Documentation:** <https://gethomepage.dev>
- **PatchMon Discord:** <https://patchmon.net/discord>
- **GitHub Issues:** <https://github.com/9technologygroup/patchmon.net/issues>

Changelog

Version 1.0.1 (2025-10-11)

- Added OS distribution support
- New fields: `top_os_1_count`, `top_os_2_count`, `top_os_3_count` for displaying infrastructure OS breakdown
- New fields: `top_os_1_name`, `top_os_2_name`, `top_os_3_name` for identifying operating systems
- Total of 14 displayable metrics now available

Version 1.0.0 (2025-10-11)

- Initial GetHomepage integration release
- Basic authentication support
- Real-time statistics endpoint
- Customizable widget mappings
- IP restriction support
- API key management UI
- 8 core metrics available

Questions or issues? Join our Discord community or open a GitHub issue!

Revision #3

Created 2025-10-11 17:36:47 UTC by lby

Updated 2025-10-11 18:57:46 UTC by lby