Cryptocurrency Mining API Research

This document provides a comprehensive overview of the API endpoints and data structures for Vnish firmware and Prohashing.com, focusing on telemetry data available from ASIC miners and merge mining operations.

Table of Contents

Vnish Firmware API

Vnish firmware is a professional, customizable alternative firmware designed for ASIC miners, particularly models from Bitmain (Antminer series), as well as other manufacturers like Whatsminer, Innosilicon, and Avalon. It offers enhanced performance, overclocking capabilities, and extensive monitoring features.

API Access and Authentication

- **Base URL**: `http://<miner_IP_address>/`
- **Documentation URL**: `http://<miner_IP_address>/docs/`
- **Authentication**: The API endpoints are protected by credentials, typically requiring the same username and password used for the web interface.

Key Endpoints

Endpoint	Method	Description
`/cgi-bin/stats.cgi`	GET	Retrieves real-time statistics about the miner's operation
`/cgi-bin/config.cgi`	GET	Retrieves current configuration settings
`/cgi-bin/config.cgi`	POST	Updates configuration settings
`/cgi-bin/reboot.cgi`	POST	Reboots the miner
`/cgi-bin/firmware.cgi`	POST	Manages firmware updates or rollbacks

Telemetry Data

The Vnish firmware API provides access to the following telemetry data:

1. **Hashrate Metrics**:
   - Current hashrate (TH/s, GH/s, etc.)
   - Average hashrate over time
   - Hashrate per hashboard
   - Accepted/rejected shares

2. **Temperature Data**:
   - PCB temperature
   - Chip temperature
   - Temperature per hashboard
   - Critical temperature alerts

3. **Power Metrics**:
   - Current power consumption (W)
   - Voltage settings
   - Power efficiency (J/TH)

4. **Fan and Cooling**:
   - Fan speed (RPM or percentage)
   - Fan status (working/failed)

5. **Operational Status**:
   - Uptime

  - Mining status (mining/idle/error)
  - Error codes and messages
  - Pool connection status

6. **Overclocking Data**:
   - Frequency settings
   - Voltage settings
   - Performance profiles

Example API Responses

Stats Endpoint (`/cgi-bin/stats.cgi`)

```
{
  "status": "success",
  "data": {
    "miner_id": "ANTMINER_S19_001",
    "hashrate": 95.5,
    "hashrate_unit": "TH/s",
    "temperature": {
      "board1": 65,
      "board2": 67,
      "board3": 66,
      "average": 66
    },
    "fan_speed": {
      "fan1": 80,
      "fan2": 82,
      "unit": "%"
    },
    "power": {
      "consumption": 3400,
      "unit": "W",
      "efficiency": 35.6,
      "efficiency_unit": "J/TH"
    },
    "uptime": 259200,
    "uptime_formatted": "3 days",
    "pool": "stratum+tcp://pool.example.com:3333",
    "worker": "worker1",
    "accepted_shares": 12450,
    "rejected_shares": 23,
    "status": "mining"
  }
}
```

Configuration Endpoint (`/cgi-bin/config.cgi`)

```
{
  "status": "success",
  "data": {
    "pools": [
      {
        "url": "stratum+tcp://pool.example.com:3333",
        "user": "worker1",
        "pass": "x"
      },
      {
        "url": "stratum+tcp://backup-pool.example.com:3333",
        "user": "worker1",
        "pass": "x"
      }
    ],
    "overclock": {
      "profile": "efficiency",
      "frequency": 900,
      "voltage": 12.0
    },
    "fan": {
      "mode": "auto",
      "min_speed": 60,
      "target_temp": 75
    },
    "network": {
      "dhcp": true,
      "ip": "192.168.1.100",
      "netmask": "255.255.255.0",
      "gateway": "192.168.1.1",
      "dns": "8.8.8.8"
    }
  }
}
```

Prohashing Merge Mining API

Prohashing is a cryptocurrency mining pool that supports merge mining, allowing miners to mine multiple cryptocurrencies simultaneously without additional computational effort. The API provides access to profitability data, mining status, and configuration options.

Prohashing API Access and Authentication

- **API Key**: Required for accessing the API endpoints
- **WAMP API**: Primary API using Web Application Messaging Protocol for real-time data
- **HTTP API**: Secondary API for services that don’t support WAMP

Prohashing Key Endpoints

WAMP API Subscriptions

The WAMP API provides real-time event subscriptions for:
- Miner status updates
- System status
- Block updates
- Profitability data

HTTP API Endpoints

Endpoint	Method	Description
`/`	GET	Service status check (returns 200 OK)
`/status`	GET	WAMP connection status and last update timestamps
`/profitability`	GET	Current profitability metrics for various algorithms

Merge Mining Configuration

Prohashing supports merge mining, particularly for coins like Dogecoin and Litecoin, which are often mined together via Scrypt algorithms. The platform automatically merge-mines Dogecoin when mining Litecoin.

Setting Up Merge Mining

To configure merge mining:

1. **Primary Coin Selection**: Specify the primary coin in the configuration
   `c=litecoin`

2. **Mining Commands**: Include specific parameters in the mining software configuration `-o stratum+tcp://prohashing.com:3333 -u username -p c=litecoin`

3. **Pool Configuration**: For advanced setups, use JSON configuration files `json`

   ```
     {
       "PoolName": "prohashing",
       "Wallet": "your_wallet_address",
       "UserName": "your_username",
       "WorkerName": "worker1",
       "Algorithm": ["scrypt"],
       "AdditionalOptions": {
         "MergeMining": true,
         "PrimaryCoin": "litecoin"
       }
     }
   ```

Prohashing Example API Responses

Status Endpoint (`/status`)

```json
{
  "connected": true,
  "updating": true,
  "lastUpdates": {
    "profitability": "2023-05-21T00:00:52+00:00"
  }
}
```

Profitability Endpoint (`/profitability`)

```
{
  "1": {
    "algorithm_name": "Scrypt",
    "usd": 0.0025041009705167,
    "btc": 6.8628068694276e-7,
    "max_usd": 0.0052676028280028,
    "max_btc": 1.4445957481592e-6,
    "percentile_usd": 0.0023765266674273,
    "percentile_btc": 6.5173886514719e-7,
    "data_timestamp": 1547510680.4959,
    "server_timestamp": 1547510683.3438,
    "server_id": 4
  },
  "2": {
    "algorithm_name": "SHA-256",
    "usd": 1.3292067065643e-7,
    "btc": 3.6428598623226e-11,
    "max_usd": 2.0160622968307e-7,
    "max_btc": 5.5308075026561e-11,
    "percentile_usd": 1.3077709035238e-7,
    "percentile_btc": 3.5864319816693e-11,
    "data_timestamp": 1547510680.4959,
    "server_timestamp": 1547510683.3438,
    "server_id": 4
  },
  "4": {
    "algorithm_name": "X11",
    "usd": 5.8147956354503e-5,
    "btc": 1.5936186240546e-8,
    "max_usd": 8.1352916386365e-5,
    "max_btc": 2.2361681776327e-8,
    "percentile_usd": 5.4252906795833e-5,
    "percentile_btc": 1.4878321539867e-8,
    "data_timestamp": 1547510680.4959,
    "server_timestamp": 1547510683.3438,
    "server_id": 4
  }
}
```

References

Vnish Firmware Resources

- Vnish Firmware Official Website (https://vnish-firmware.com/en/)
- Vnish Firmware FAQ (https://vnish.group/en/faq-vnish-en/)
- Vnish GitHub Repository (https://github.com/Vnish-team/)

Prohashing Resources

- Prohashing API Documentation (https://prohashing.com/help/prohashing-api-developing)
- Prohashing Merge Mining Guide (https://prohashing.com/guides/merge-mining)
- GitHub - foremanmining/prohashing-http (https://github.com/foremanmining/prohashing-http)