ntopng scripting how-to

Emanuele Faranda
faranda@ntop.org

Pisa • October 28, 2017

Setting up the environment

Examples and Sources

https://github.com/simonemainardi/ntopng-docker

Docker Installation

```
$ sudo apt-get install docker

$ sudo usermod -a -G docker emanuele

$ newgrp docker
```

ntopng Docker Setup

```
$ git clone https://github.com/simonemainardi/ntopng-docker

$ cd ntopng-docker

$ docker build -t ntopng-docker -f Dockerfile.ntopng .
```

Running ntopng

Docker

```
$ docker run -p 3000:3000 -u root -v `pwd`/workspace:/
home/ntopng/workspace --rm -it ntopng-docker
workspace
```

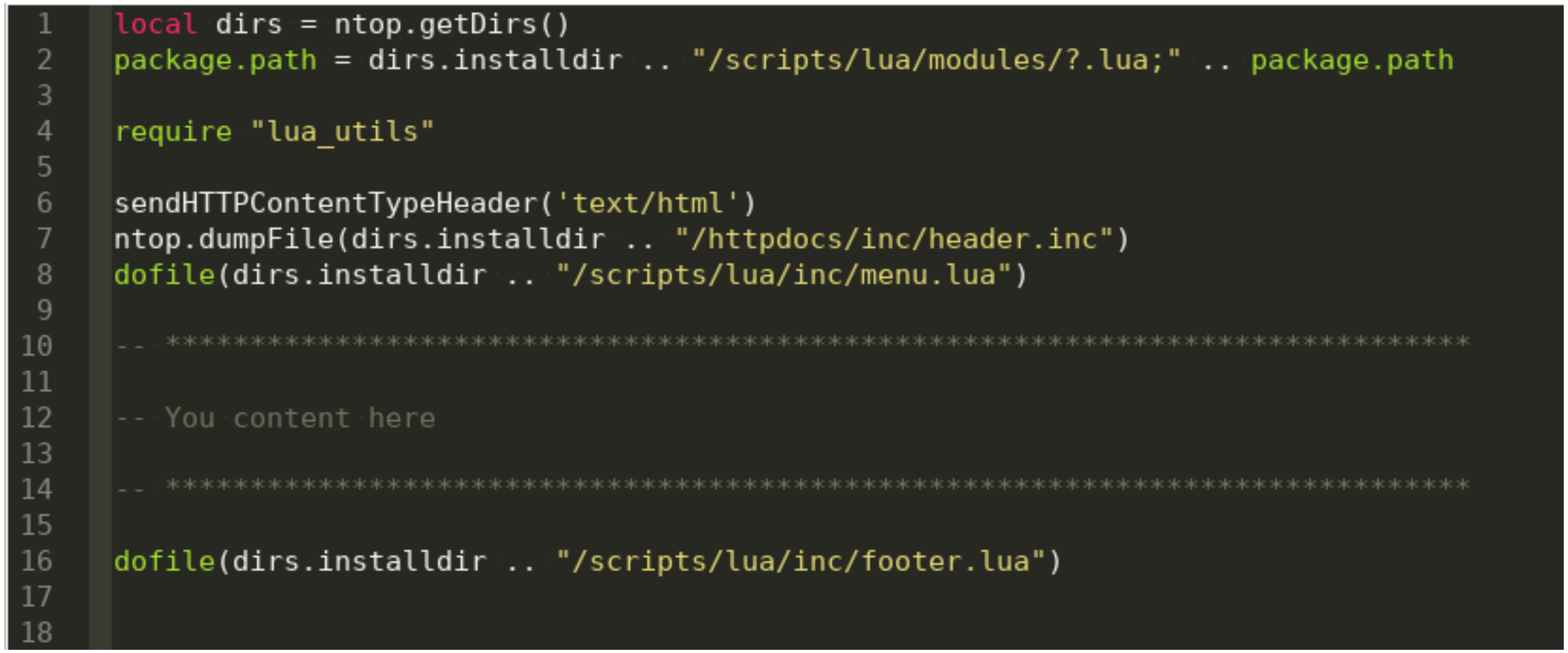
or Locally

```
$ sudo ./ntopng -s -i wlan0
```

Lua Scripting Basics

A Script Skeleton

http://127.0.0.1:3000/lua/myscripts/skeleton.lua

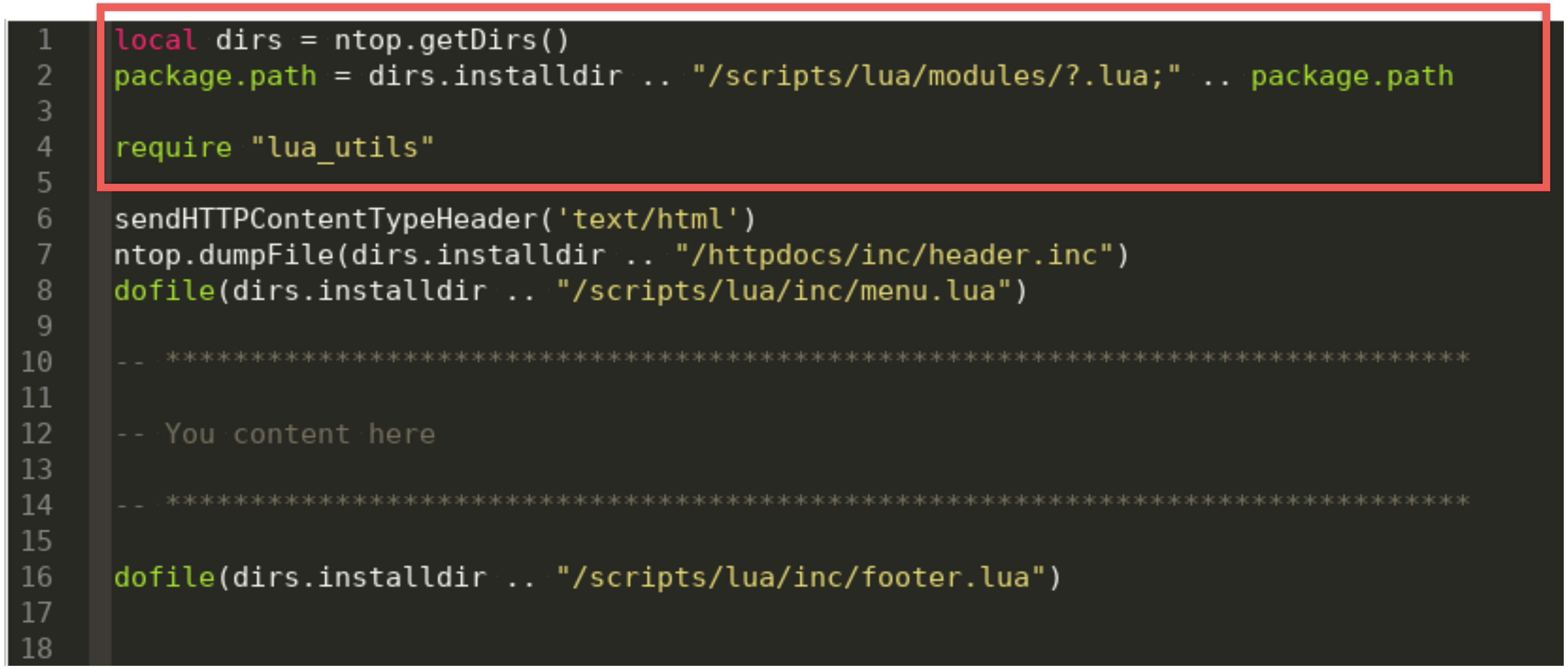


```lua
local dirs = ntop.getDirs()
package.path = dirs.installdir .. "/scripts/lua/modules/?.lua;" .. package.path

require "lua_utils"

sendHTTPContentTypeHeader('text/html')
ntop.dumpFile(dirs.installdir .. "/httpdocs/inc/header.inc")
dofile(dirs.installdir .. "/scripts/lua/inc/menu.lua")

-- ****************************************************************************

-- You content here

-- ****************************************************************************

dofile(dirs.installdir .. "/scripts/lua/inc/footer.lua")

```

A Script Skeleton

http://127.0.0.1:3000/lua/myscripts/skeleton.lua

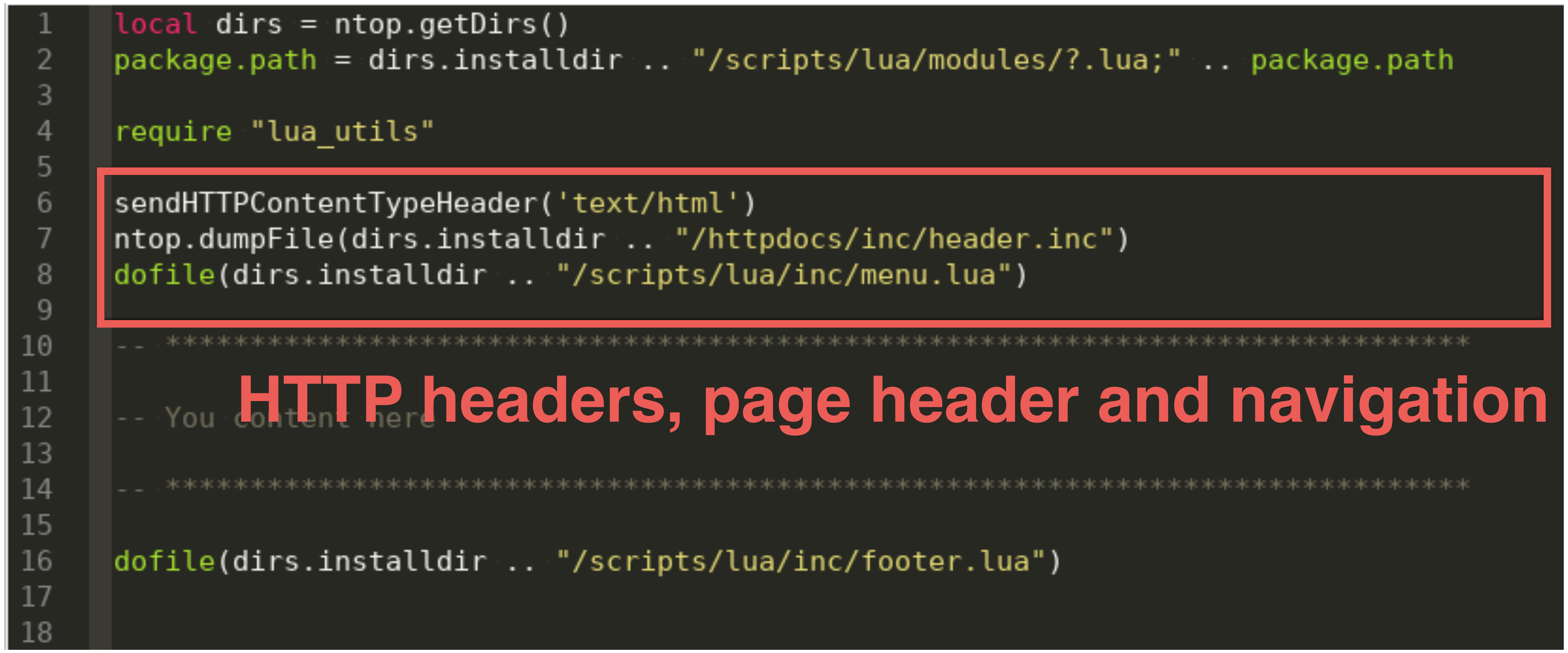
Packages path and includes



```lua
local dirs = ntop.getDirs()
package.path = dirs.installdir .. "/scripts/lua/modules/?.lua;" .. package.path

require "lua_utils"

sendHTTPContentTypeHeader('text/html')
ntop.dumpFile(dirs.installdir .. "/httpdocs/inc/header.inc")
dofile(dirs.installdir .. "/scripts/lua/inc/menu.lua")

-- ****************************************************************************

-- You content here

-- ****************************************************************************

dofile(dirs.installdir .. "/scripts/lua/inc/footer.lua")

```

A Script Skeleton

http://127.0.0.1:3000/lua/myscripts/skeleton.lua

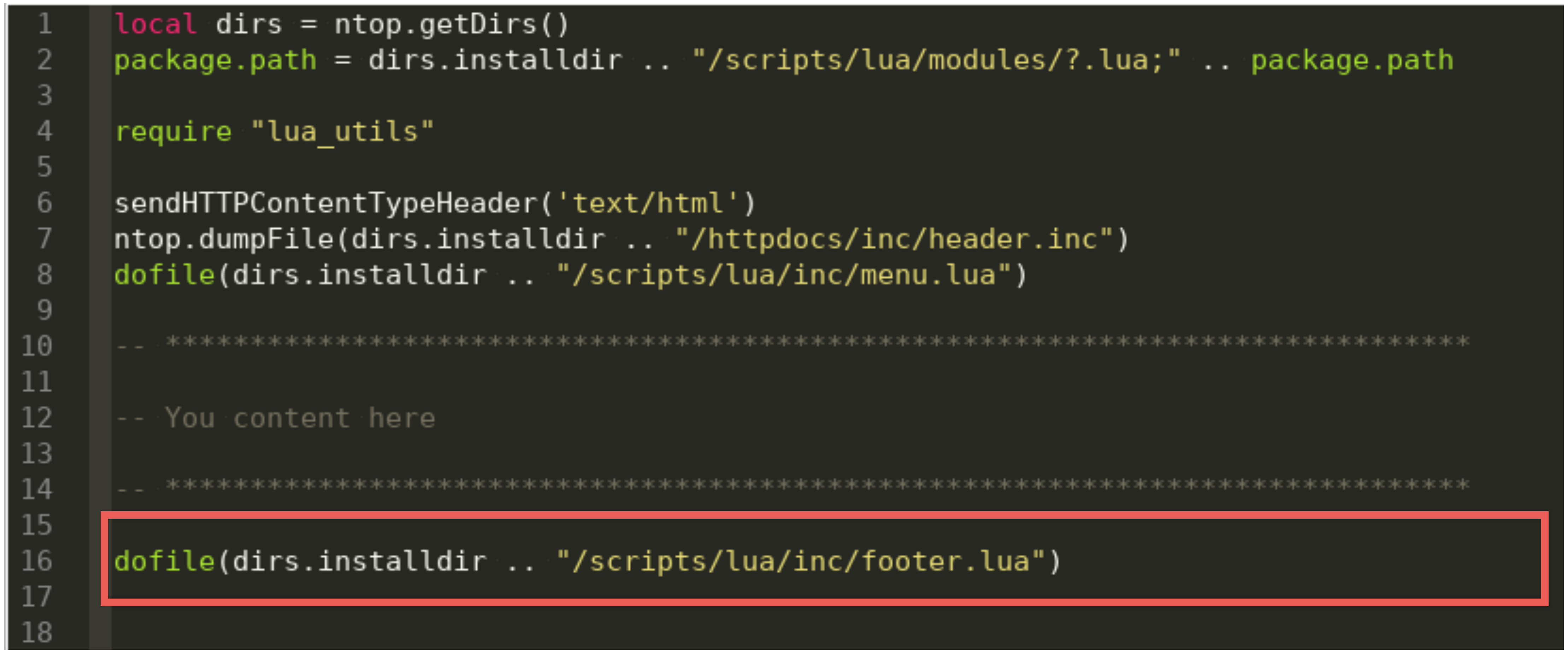


```lua
local dirs = ntop.getDirs()
package.path = dirs.installdir .. "/scripts/lua/modules/?.lua;" .. package.path

require "lua_utils"

sendHTTPContentTypeHeader('text/html')
ntop.dumpFile(dirs.installdir .. "/httpdocs/inc/header.inc")
dofile(dirs.installdir .. "/scripts/lua/inc/menu.lua")

-- ****************************************************************************

-- You content here

-- ****************************************************************************

dofile(dirs.installdir .. "/scripts/lua/inc/footer.lua")
```

HTTP headers, page header and navigation menu

A Script Skeleton

http://127.0.0.1:3000/lua/myscripts/skeleton.lua



```lua
local dirs = ntop.getDirs()
package.path = dirs.installdir .. "/scripts/lua/modules/?.lua;" .. package.path

require "lua_utils"

sendHTTPContentTypeHeader('text/html')
ntop.dumpFile(dirs.installdir .. "/httpdocs/inc/header.inc")
dofile(dirs.installdir .. "/scripts/lua/inc/menu.lua")

-- ****************************************************************************

-- You content here

-- ****************************************************************************

dofile(dirs.installdir .. "/scripts/lua/inc/footer.lua")

```

Footer

Scripts Directories

Docker

- ntopng-docker/workspace/scripts

or Locally

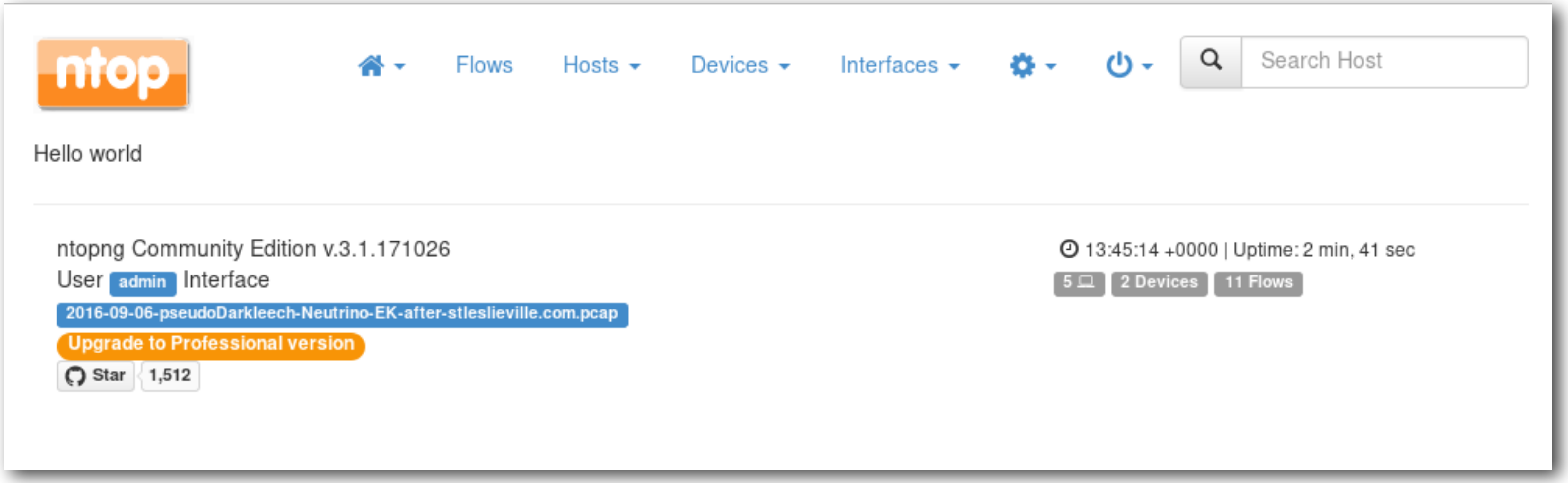
- ntopng/scripts

Examples presented in this how-to can be found at

ntopng-docker/workspace/myscripts

Hello World script

http://127.0.0.1:3000/lua/myscripts/hello.lua



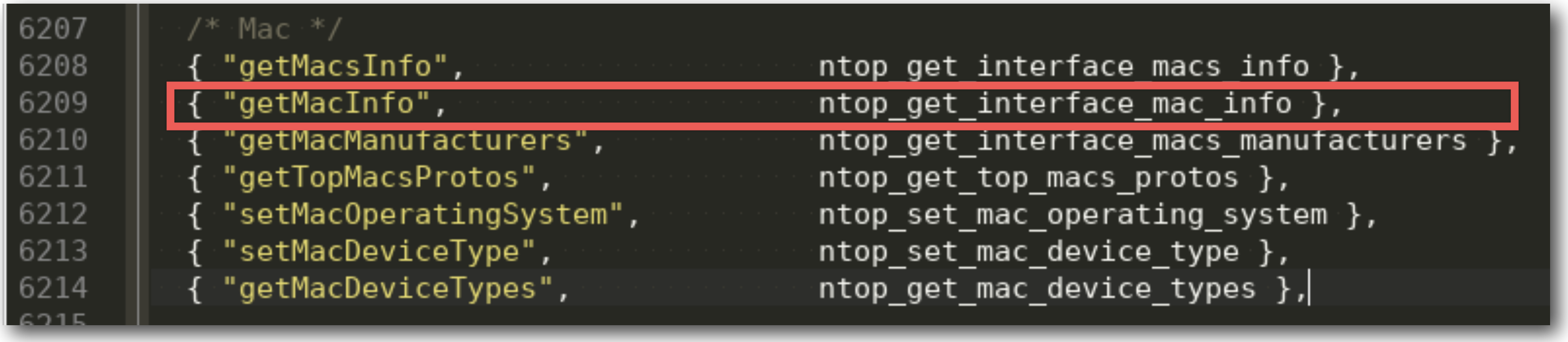
Hello world

ntopng Community Edition v.3.1.171026
User admin Interface
2016-09-06-pseudoDarkleech-Neutrino-EK-after-stleslieville.com.pcap
Upgrade to Professional version
Star 1,512

13:45:14 +0000 | Uptime: 2 min, 41 sec
5 | 2 Devices | 11 Flows

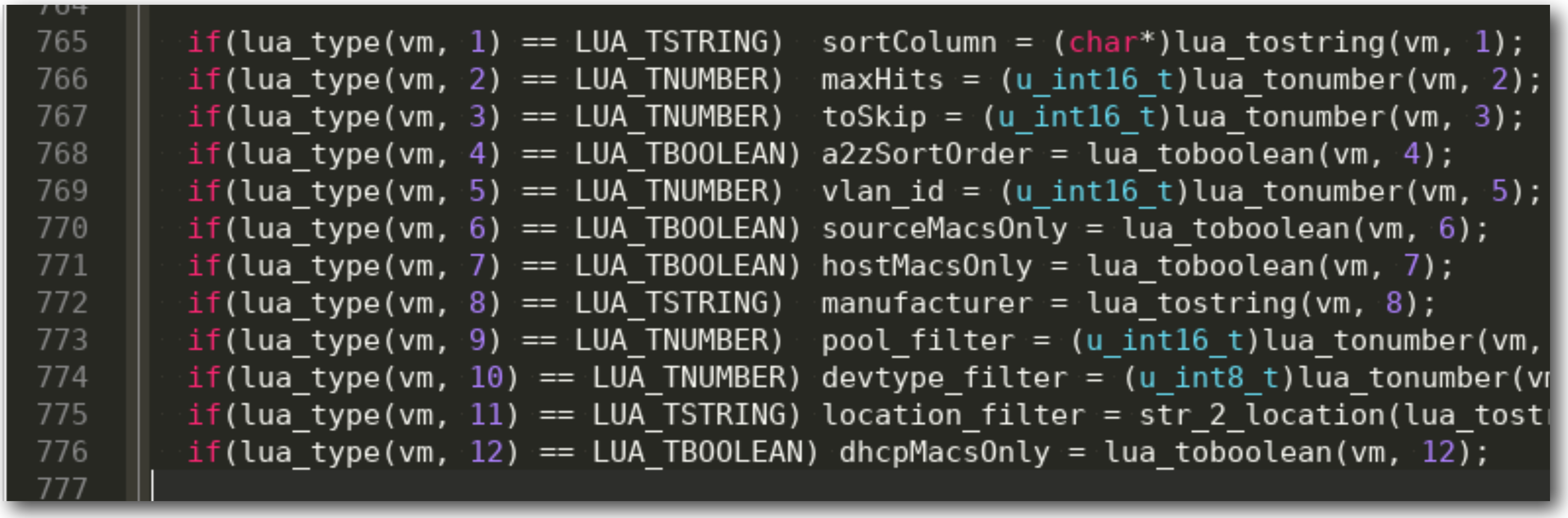
Script example: Devices in my network

Lua C Api



```
/* Mac */
{ "getMacsInfo",                    ntop_get_interface_macs_info },
{ "getMacInfo",                     ntop_get_interface_mac_info },
{ "getMacManufacturers",            ntop_get_interface_macs_manufacturers },
{ "getTopMacsProtos",               ntop_get_top_macs_protos },
{ "setMacOperatingSystem",          ntop_set_mac_operating_system },
{ "setMacDeviceType",               ntop_set_mac_device_type },
{ "getMacDeviceTypes",              ntop_get_mac_device_types },
```

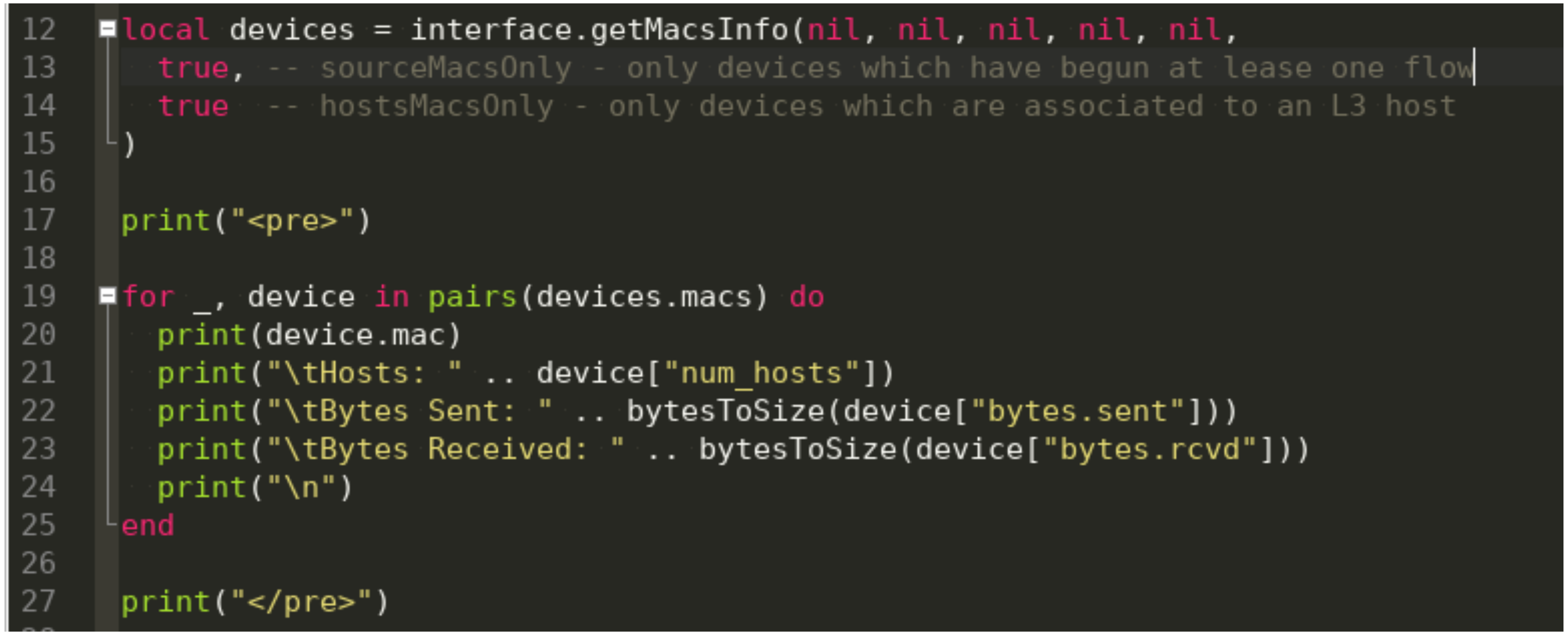
interface.getMacsInfo



```
if(lua_type(vm, 1) == LUA_TSTRING)  sortColumn = (char*)lua_tostring(vm, 1);
if(lua_type(vm, 2) == LUA_TNUMBER)  maxHits = (u_int16_t)lua_tonumber(vm, 2);
if(lua_type(vm, 3) == LUA_TNUMBER)  toSkip = (u_int16_t)lua_tonumber(vm, 3);
if(lua_type(vm, 4) == LUA_TBOOLEAN) a2zSortOrder = lua_toboolean(vm, 4);
if(lua_type(vm, 5) == LUA_TNUMBER)  vlan_id = (u_int16_t)lua_tonumber(vm, 5);
if(lua_type(vm, 6) == LUA_TBOOLEAN) sourceMacsOnly = lua_toboolean(vm, 6);
if(lua_type(vm, 7) == LUA_TBOOLEAN) hostMacsOnly = lua_toboolean(vm, 7);
if(lua_type(vm, 8) == LUA_TSTRING)  manufacturer = lua_tostring(vm, 8);
if(lua_type(vm, 9) == LUA_TNUMBER)  pool_filter = (u_int16_t)lua_tonumber(vm,
if(lua_type(vm, 10) == LUA_TNUMBER) devtype_filter = (u_int8_t)lua_tonumber(v
if(lua_type(vm, 11) == LUA_TSTRING) location_filter = str_2_location(lua_tost
if(lua_type(vm, 12) == LUA_TBOOLEAN) dhcpMacsOnly = lua_toboolean(vm, 12);
```

Retrieving devices data

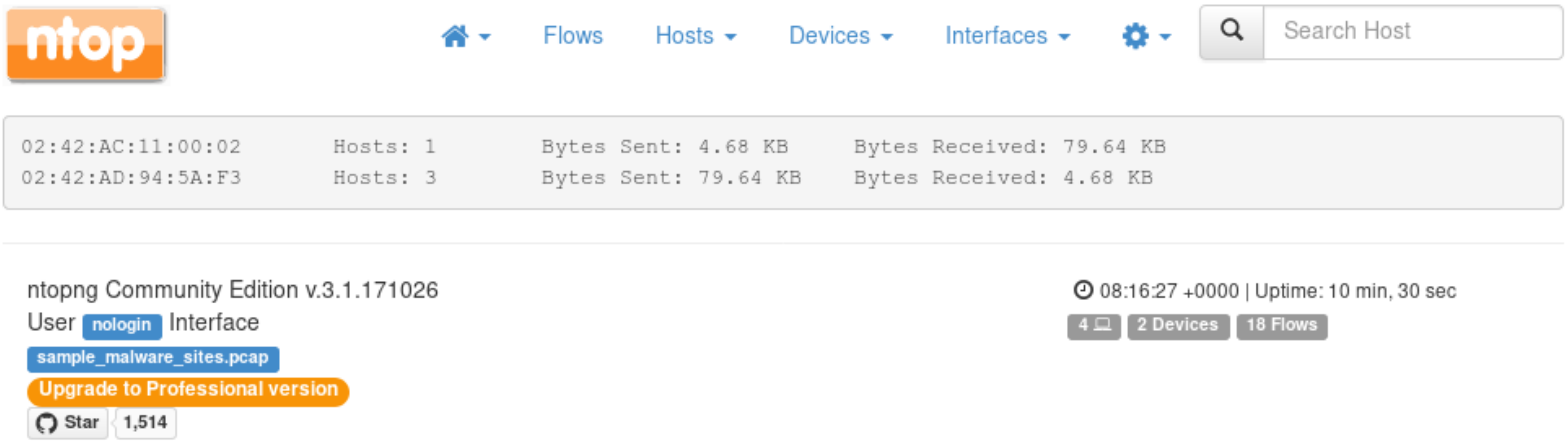
http://127.0.0.1:3000/lua/myscripts/devices.lua



```
local devices = interface.getMacsInfo(nil, nil, nil, nil, nil,
  true, -- sourceMacsOnly - only devices which have begun at lease one flow
  true  -- hostsMacsOnly - only devices which are associated to an L3 host
)

print("<pre>")

for _, device in pairs(devices.macs) do
  print(device.mac)
  print("\tHosts: " .. device["num_hosts"])
  print("\tBytes Sent: " .. bytesToSize(device["bytes.sent"]))
  print("\tBytes Received: " .. bytesToSize(device["bytes.rcvd"]))
  print("\n")
end

print("</pre>")
```

Final Result

http://127.0.0.1:3000/lua/myscripts/devices.lua



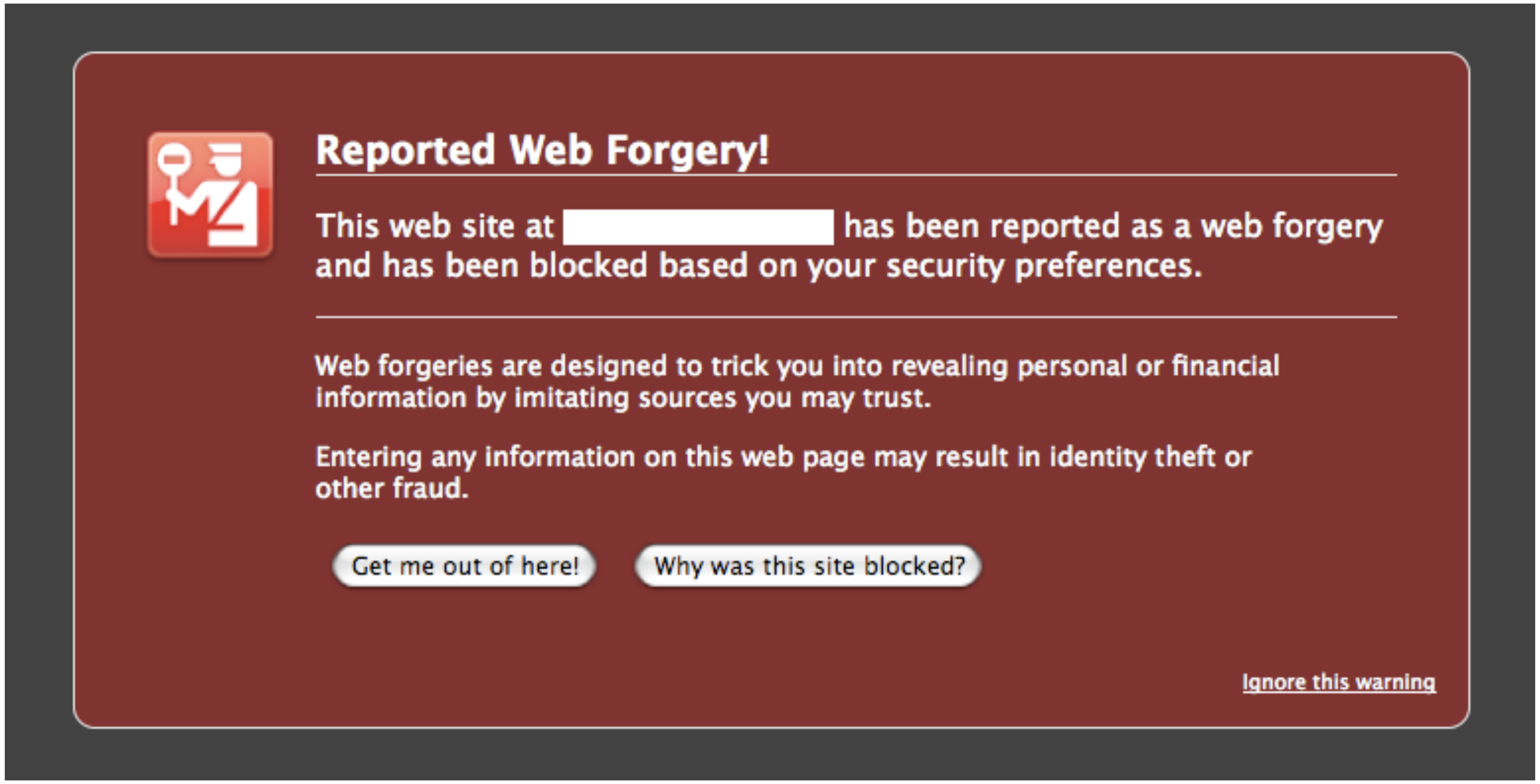
```
02:42:AC:11:00:02       Hosts: 1        Bytes Sent: 4.68 KB     Bytes Received: 79.64 KB
02:42:AD:94:5A:F3       Hosts: 3        Bytes Sent: 79.64 KB    Bytes Received: 4.68 KB
```

ntopng Community Edition v.3.1.171026
User nologin Interface sample_malware_sites.pcap
Upgrade to Professional version
Star 1,514

08:16:27 +0000 | Uptime: 10 min, 30 sec
4 | 2 Devices | 18 Flows

Script example:
Processing flows for walware analysis

Google Safe Browsing



Reported Web Forgery!

This web site at [redacted] has been reported as a web forgery and has been blocked based on your security preferences.

Web forgeries are designed to trick you into revealing personal or financial information by imitating sources you may trust.

Entering any information on this web page may result in identity theft or other fraud.

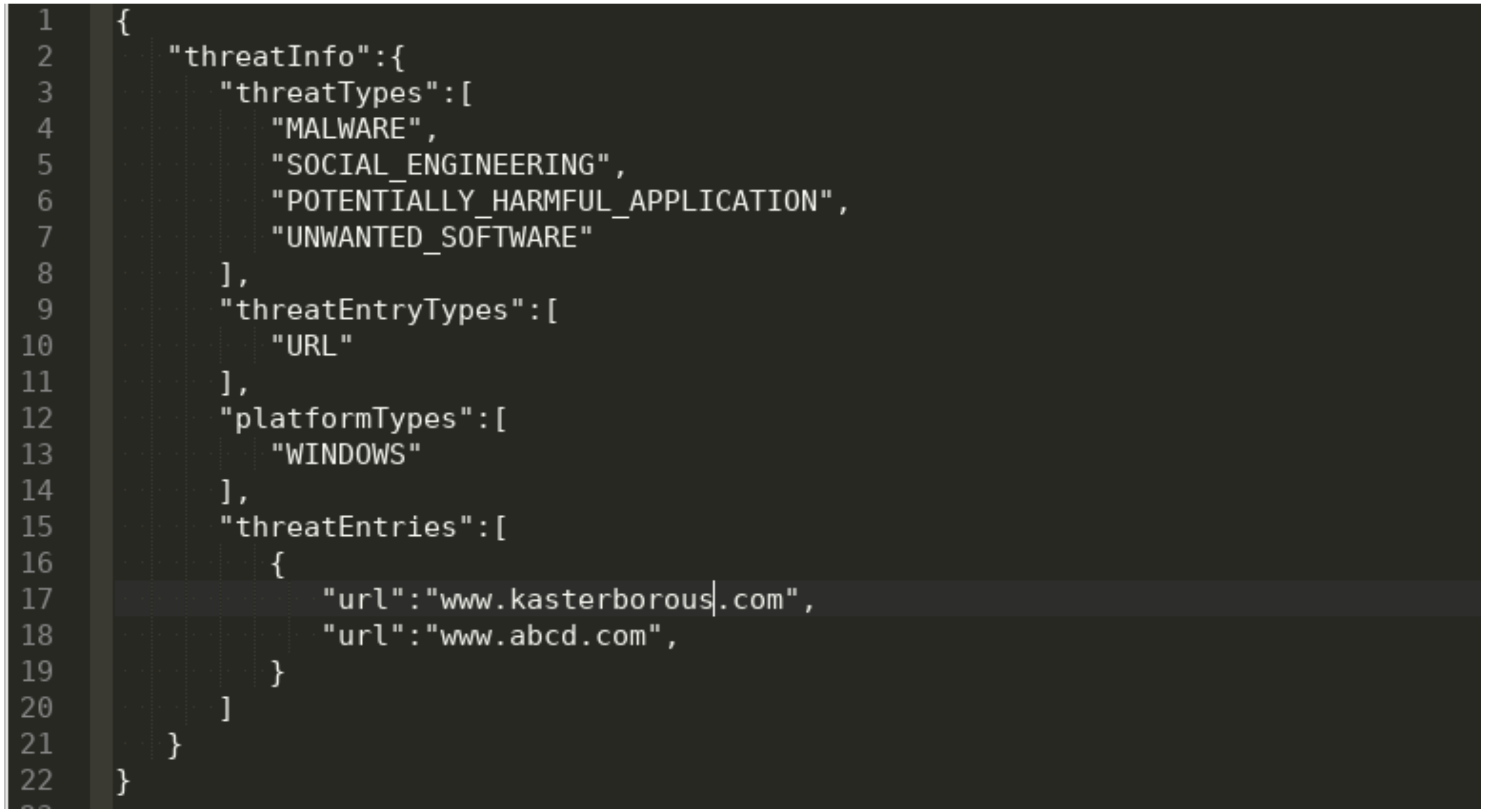
- Deployed in Chrome, Safari, Firefox, Android
- Provides an API to performs queries
- Note: with Lookup API v4 URLs are not hashed so the server knows which URLs you look up

Safe Browsing API

https://developers.google.com/safe-browsing/v4/lookup-api

- POST https://safebrowsing.googleapis.com/v4/threatMatches:find
- URL parameter "key=API_KEY"
- JSON body contains request data

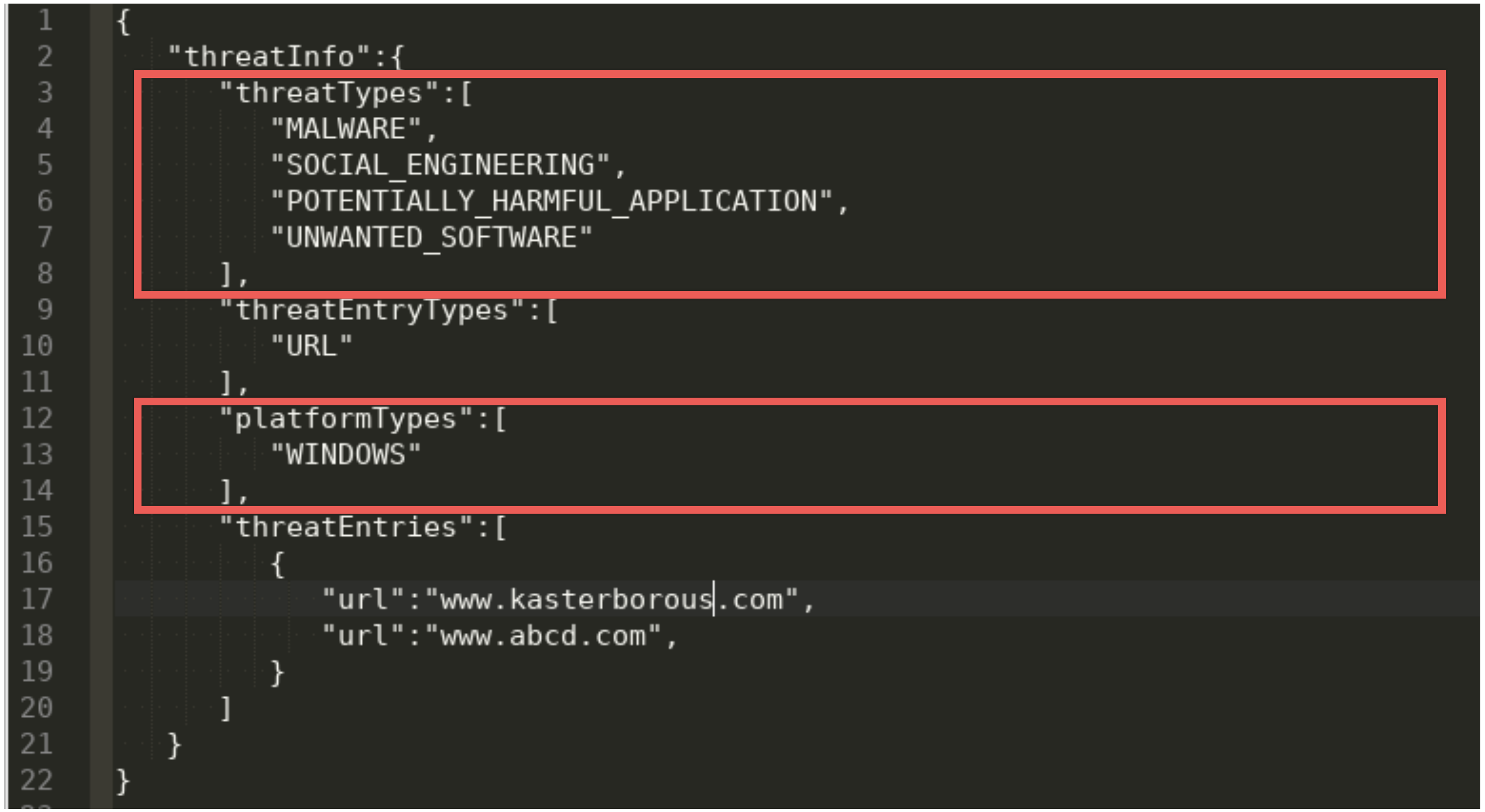
Sample Request



```
{
    "threatInfo":{
        "threatTypes":[
            "MALWARE",
            "SOCIAL_ENGINEERING",
            "POTENTIALLY_HARMFUL_APPLICATION",
            "UNWANTED_SOFTWARE"
        ],
        "threatEntryTypes":[
            "URL"
        ],
        "platformTypes":[
            "WINDOWS"
        ],
        "threatEntries":[
            {
                "url":"www.kasterborous.com",
                "url":"www.abcd.com",
            }
        ]
    }
}
```

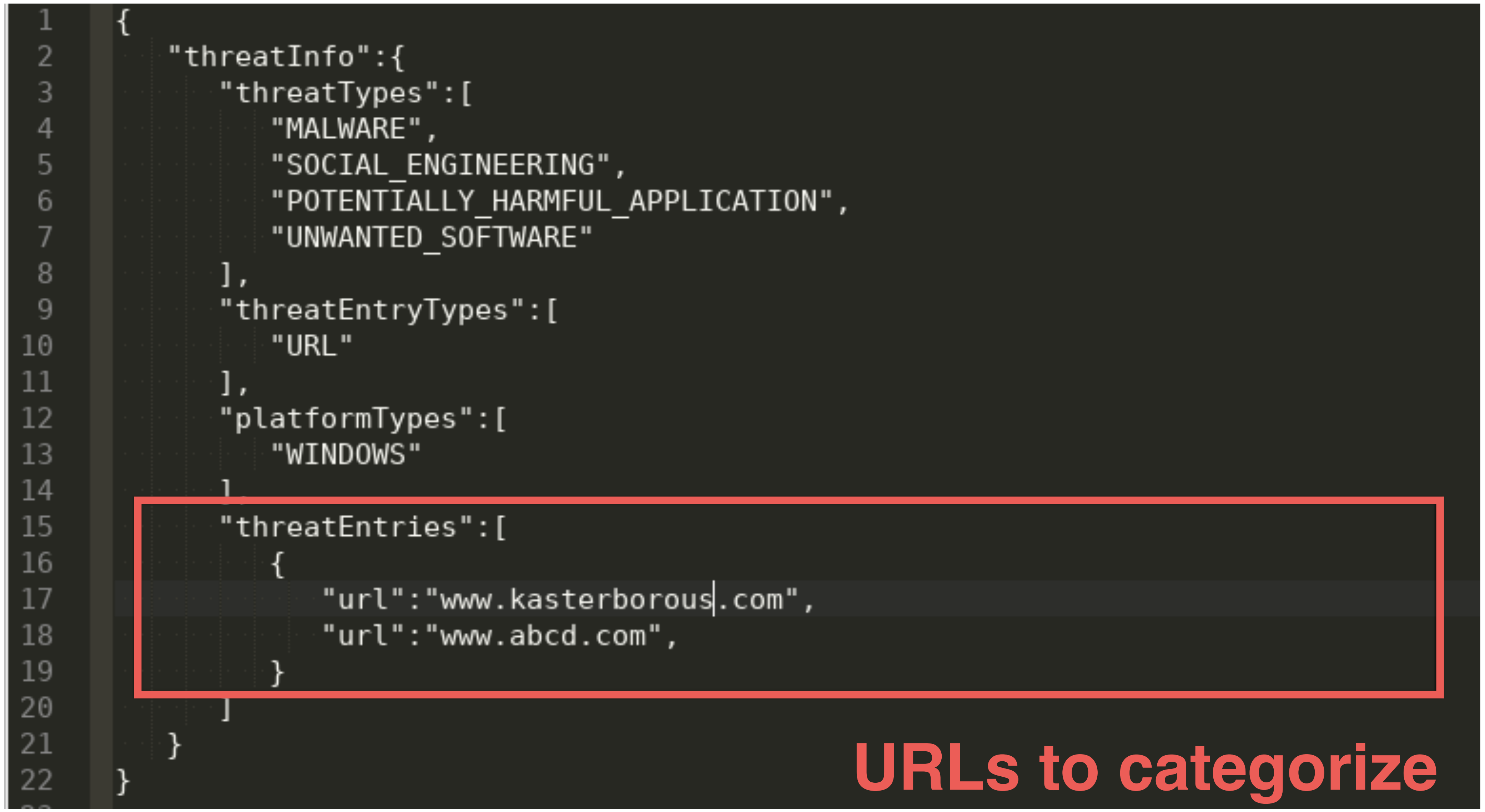
Sample Request

Threat information of interest



```
{
    "threatInfo":{
        "threatTypes":[
            "MALWARE",
            "SOCIAL_ENGINEERING",
            "POTENTIALLY_HARMFUL_APPLICATION",
            "UNWANTED_SOFTWARE"
        ],
        "threatEntryTypes":[
            "URL"
        ],
        "platformTypes":[
            "WINDOWS"
        ],
        "threatEntries":[
            {
                "url":"www.kasterborous.com",
                "url":"www.abcd.com",
            }
        ]
    }
}
```

Sample Request

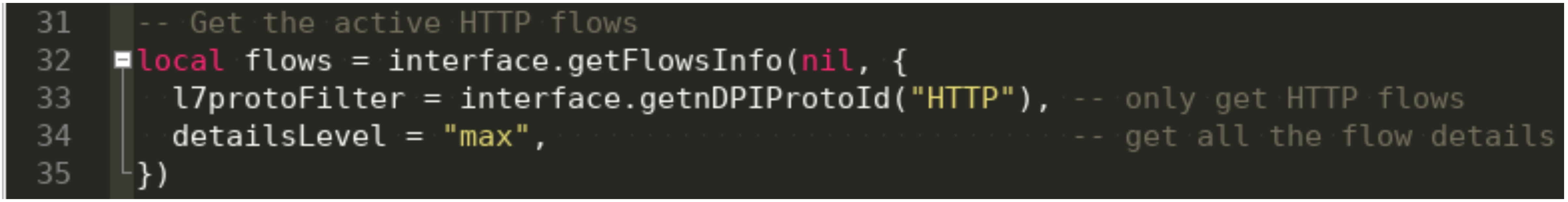


```
{
    "threatInfo":{
        "threatTypes":[
            "MALWARE",
            "SOCIAL_ENGINEERING",
            "POTENTIALLY_HARMFUL_APPLICATION",
            "UNWANTED_SOFTWARE"
        ],
        "threatEntryTypes":[
            "URL"
        ],
        "platformTypes":[
            "WINDOWS"
        ],
        "threatEntries":[
            {
                "url":"www.kasterborous.com",
                "url":"www.abcd.com",
            }
        ]
    }
}
```

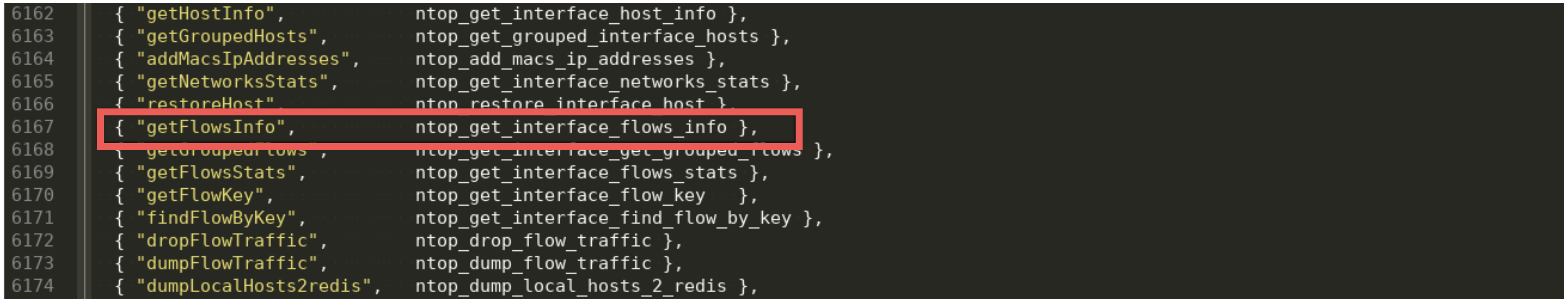
URLs to categorize

Query Active HTTP Flows

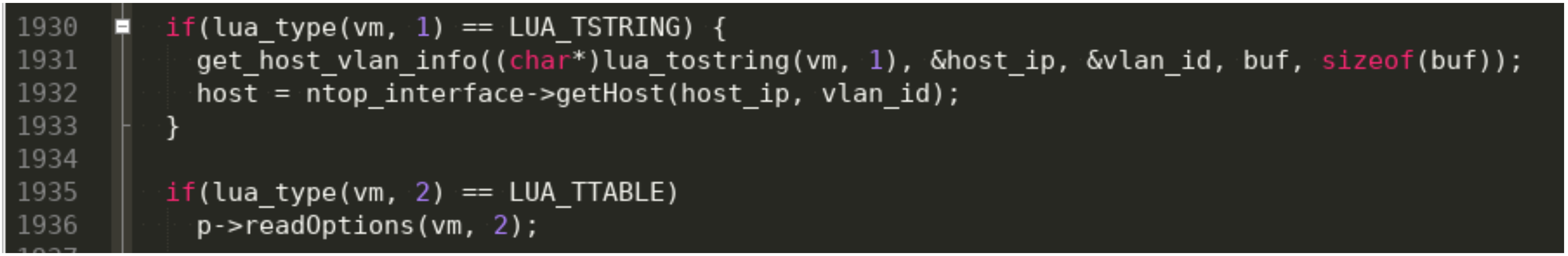
http://127.0.0.1:3000/lua/myscripts/safe_browsing_simple.lua



```lua
-- Get the active HTTP flows
local flows = interface.getFlowsInfo(nil, {
  l7protoFilter = interface.getnDPIProtoId("HTTP"), -- only get HTTP flows
  detailsLevel = "max",                             -- get all the flow details
})
```



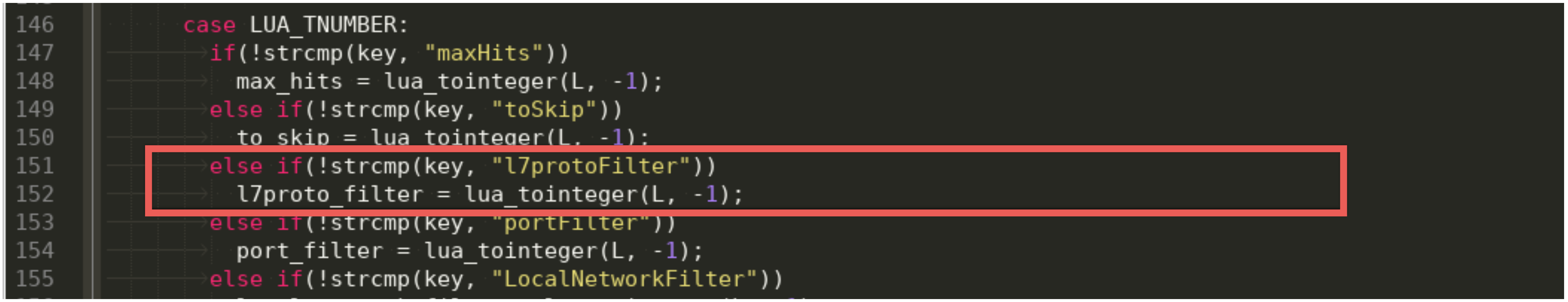
```c
{ "getHostInfo",          ntop_get_interface_host_info },
{ "getGroupedHosts",      ntop_get_grouped_interface_hosts },
{ "addMacsIpAddresses",   ntop_add_macs_ip_addresses },
{ "getNetworksStats",     ntop_get_interface_networks_stats },
{ "restoreHost",          ntop_restore_interface_host },
{ "getFlowsInfo",         ntop_get_interface_flows_info },
{ "getGroupedFlows",      ntop_get_interface_get_grouped_flows },
{ "getFlowsStats",        ntop_get_interface_flows_stats },
{ "getFlowKey",           ntop_get_interface_flow_key   },
{ "findFlowByKey",        ntop_get_interface_find_flow_by_key },
{ "dropFlowTraffic",      ntop_drop_flow_traffic },
{ "dumpFlowTraffic",      ntop_dump_flow_traffic },
{ "dumpLocalHosts2redis", ntop_dump_local_hosts_2_redis },
```



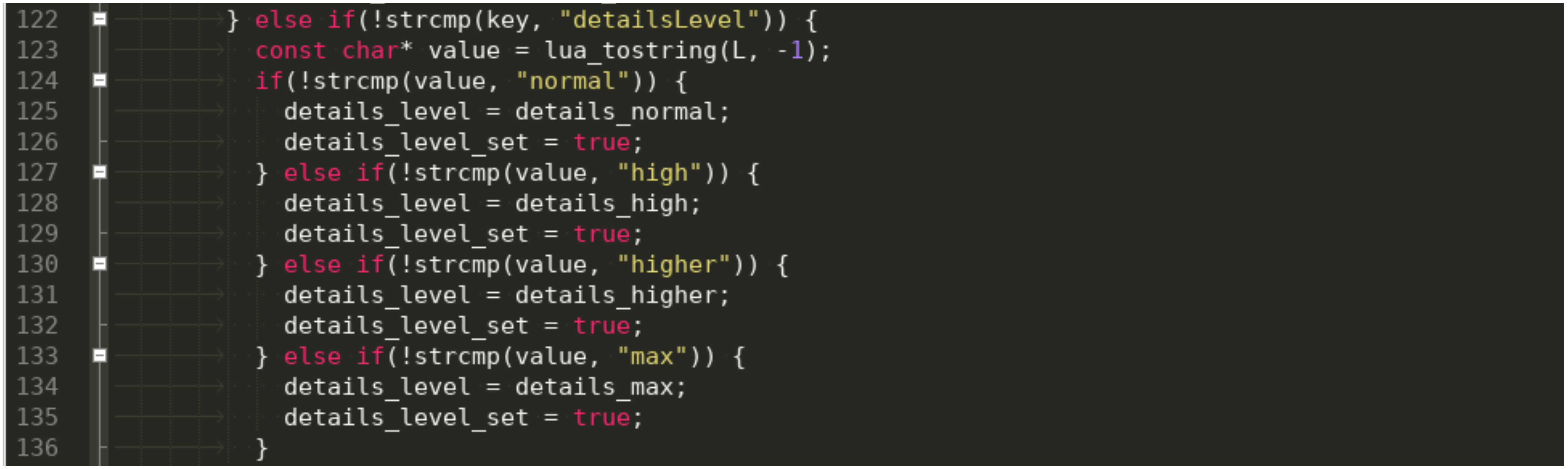
```c
  if(lua_type(vm, 1) == LUA_TSTRING) {
    get_host_vlan_info((char*)lua_tostring(vm, 1), &host_ip, &vlan_id, buf, sizeof(buf));
    host = ntop_interface->getHost(host_ip, vlan_id);
  }

  if(lua_type(vm, 2) == LUA_TTABLE)
    p->readOptions(vm, 2);
```

Paginator Options

Paginator.cpp



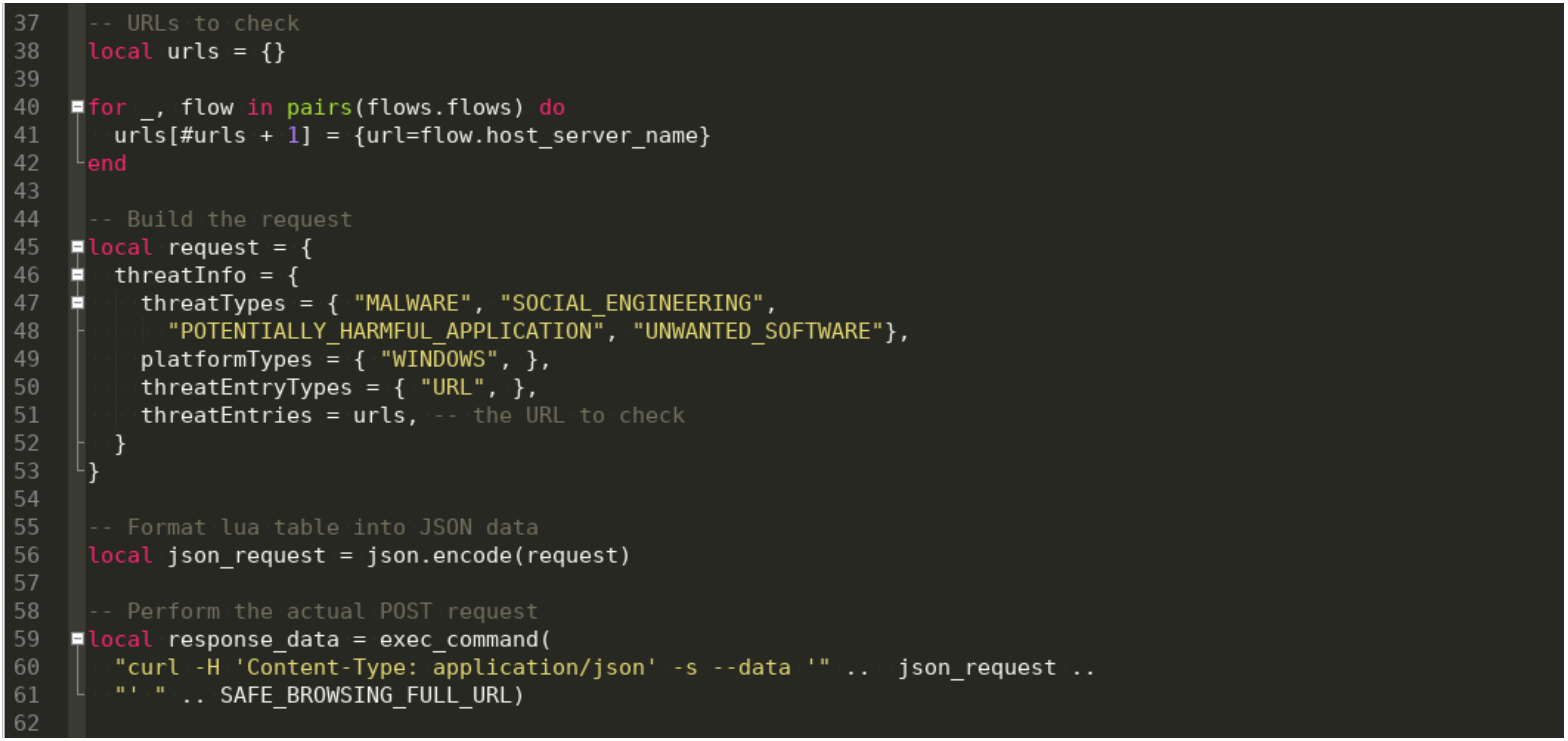
```
        case LUA_TNUMBER:
          if(!strcmp(key, "maxHits"))
            max_hits = lua_tointeger(L, -1);
          else if(!strcmp(key, "toSkip"))
            to_skip = lua_tointeger(L, -1);
          else if(!strcmp(key, "l7protoFilter"))
            l7proto_filter = lua_tointeger(L, -1);
          else if(!strcmp(key, "portFilter"))
            port_filter = lua_tointeger(L, -1);
          else if(!strcmp(key, "LocalNetworkFilter"))
```



```
          } else if(!strcmp(key, "detailsLevel")) {
            const char* value = lua_tostring(L, -1);
            if(!strcmp(value, "normal")) {
              details_level = details_normal;
              details_level_set = true;
            } else if(!strcmp(value, "high")) {
              details_level = details_high;
              details_level_set = true;
            } else if(!strcmp(value, "higher")) {
              details_level = details_higher;
              details_level_set = true;
            } else if(!strcmp(value, "max")) {
              details_level = details_max;
              details_level_set = true;
            }
```

Send Request Data

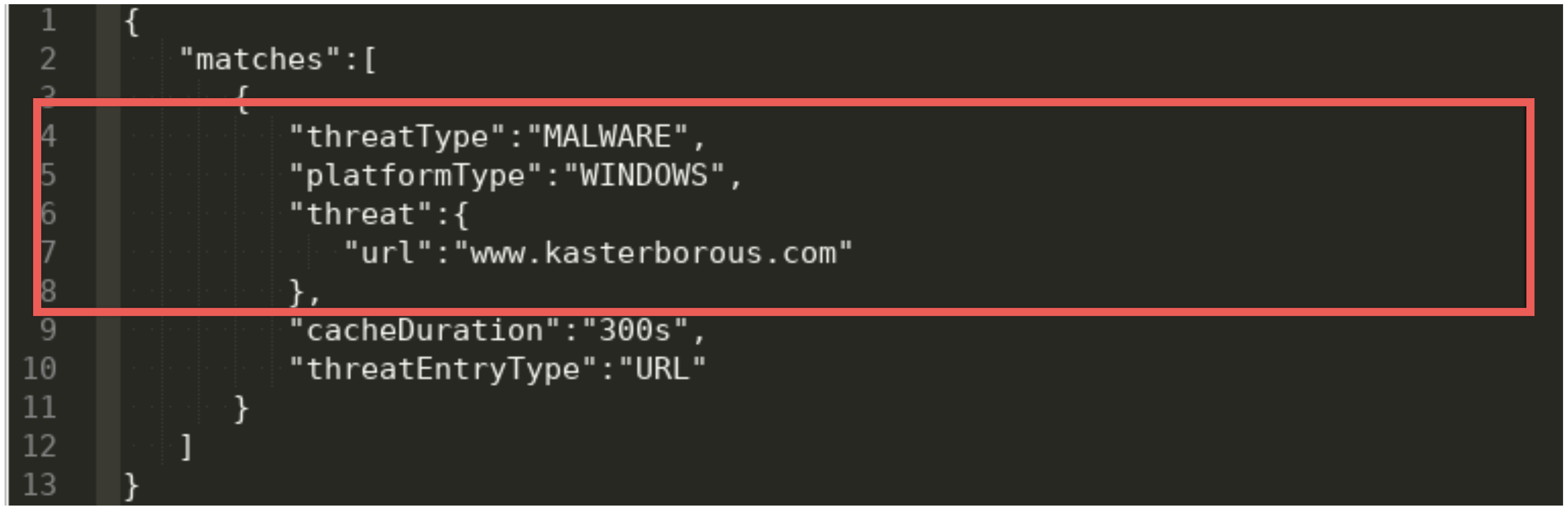
http://127.0.0.1:3000/lua/myscripts/safe_browsing_simple.lua



```lua
-- URLs to check
local urls = {}

for _, flow in pairs(flows.flows) do
  urls[#urls + 1] = {url=flow.host_server_name}
end

-- Build the request
local request = {
  threatInfo = {
    threatTypes = { "MALWARE", "SOCIAL_ENGINEERING",
      "POTENTIALLY_HARMFUL_APPLICATION", "UNWANTED_SOFTWARE"},
    platformTypes = { "WINDOWS", },
    threatEntryTypes = { "URL", },
    threatEntries = urls, -- the URL to check
  }
}

-- Format lua table into JSON data
local json_request = json.encode(request)

-- Perform the actual POST request
local response_data = exec_command(
  "curl -H 'Content-Type: application/json' -s --data '" ..  json_request ..
  "' " .. SAFE_BROWSING_FULL_URL)

```

Sample Reply

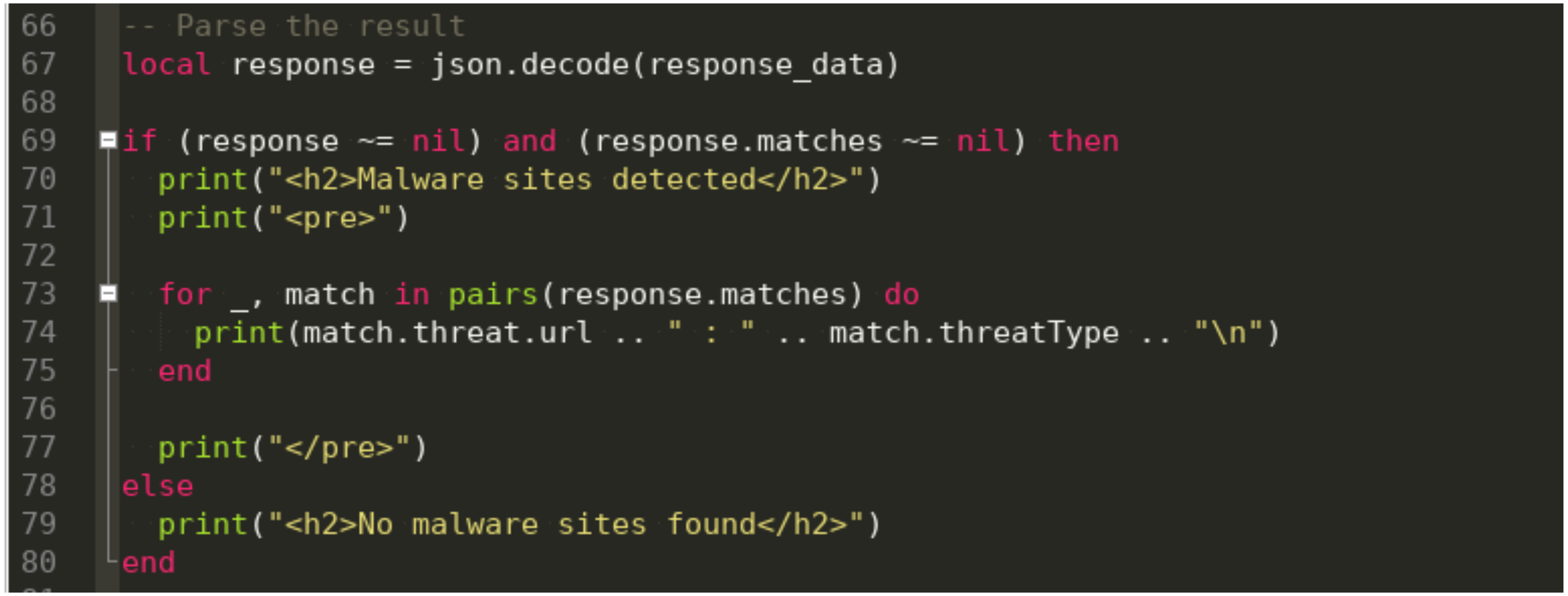
It's a Windows malware!



```
{
  "matches":[
    {
      "threatType":"MALWARE",
      "platformType":"WINDOWS",
      "threat":{
        "url":"www.kasterborous.com"
      },
      "cacheDuration":"300s",
      "threatEntryType":"URL"
    }
  ]
}
```

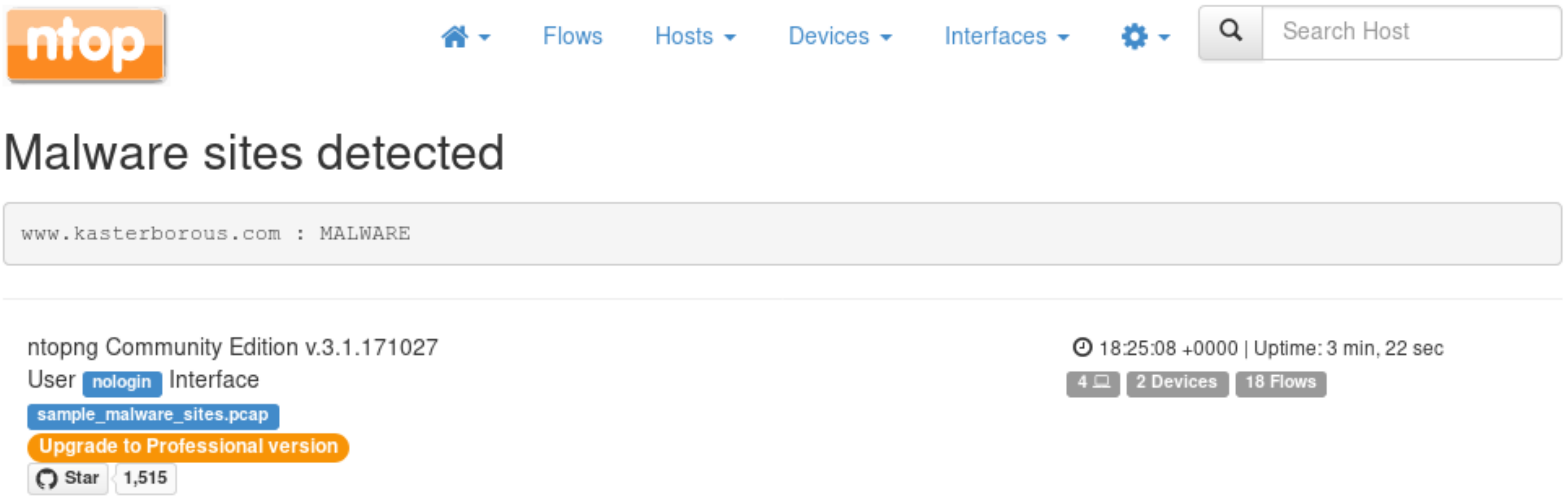
Get Response Data

http://127.0.0.1:3000/lua/myscripts/safe_browsing_simple.lua



```
-- Parse the result
local response = json.decode(response_data)

if (response ~= nil) and (response.matches ~= nil) then
  print("<h2>Malware sites detected</h2>")
  print("<pre>")

  for _, match in pairs(response.matches) do
    print(match.threat.url .. " : " .. match.threatType .. "\n")
  end

  print("</pre>")
else
  print("<h2>No malware sites found</h2>")
end
```

Result



Malware sites detected

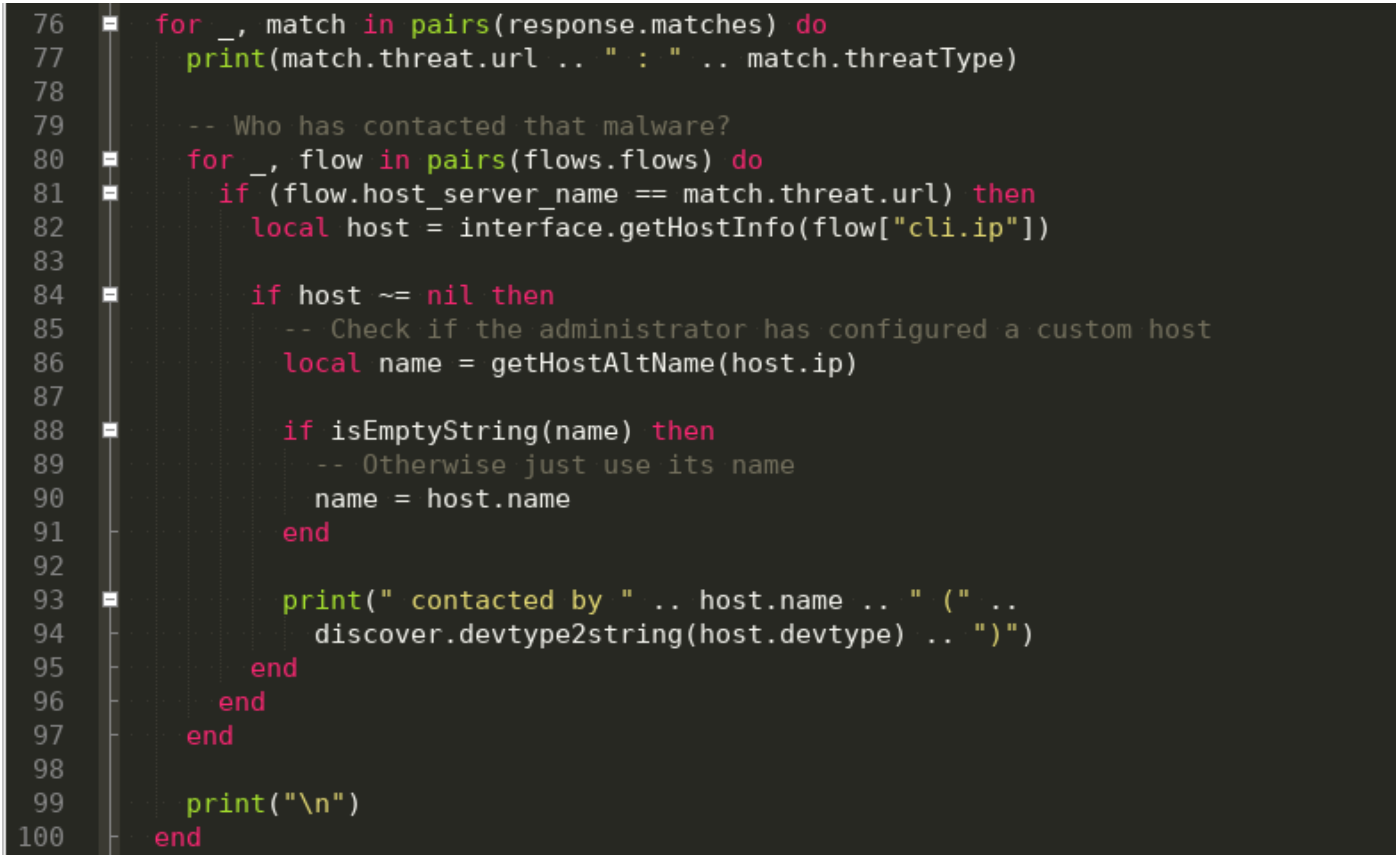
```
www.kasterborous.com : MALWARE
```

ntopng Community Edition v.3.1.171027
User nologin Interface sample_malware_sites.pcap
Upgrade to Professional version
Star 1,515

18:25:08 +0000 | Uptime: 3 min, 22 sec
4 | 2 Devices | 18 Flows

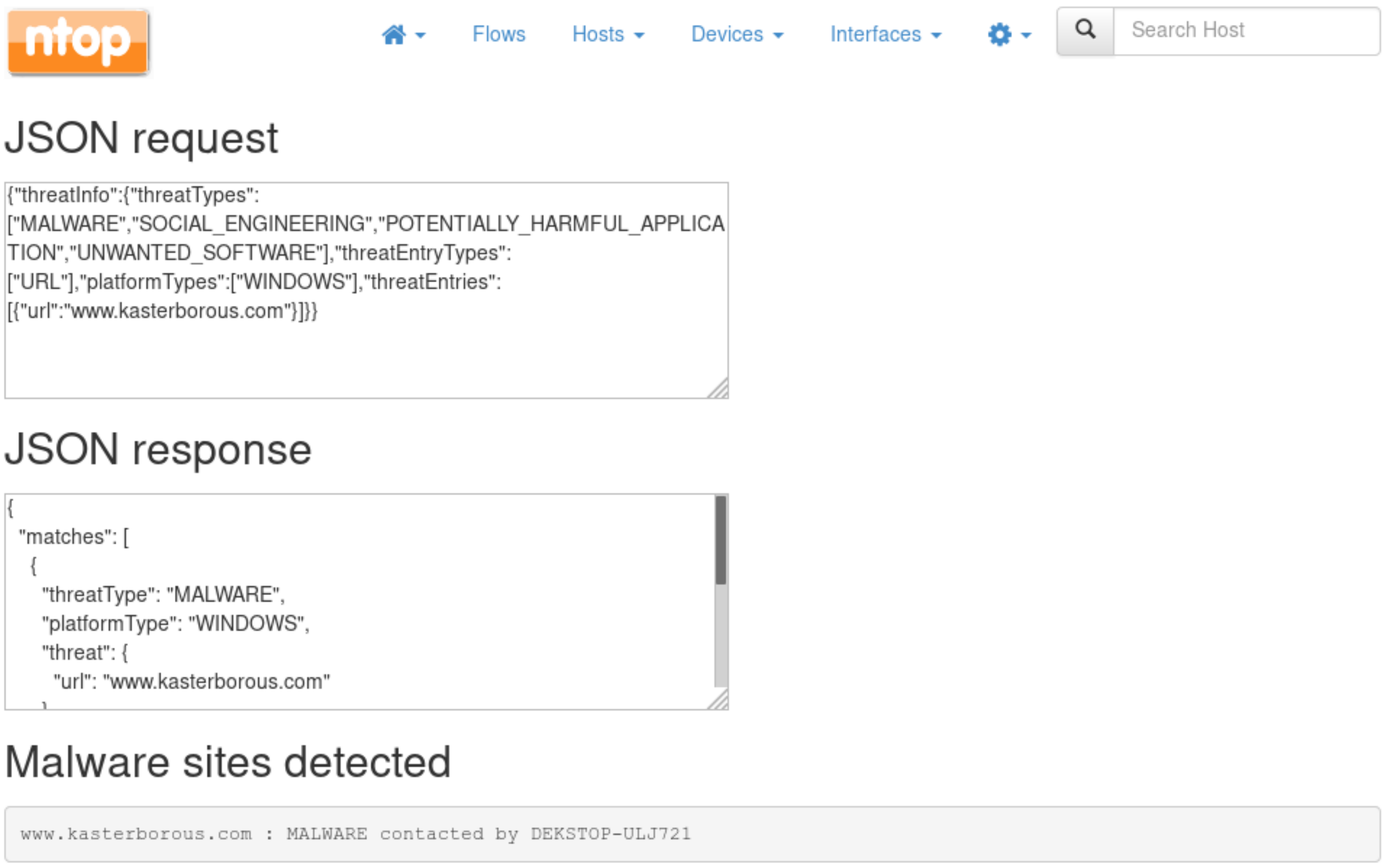
Adding Hosts Information

http://127.0.0.1:3000/lua/myscripts/safe_browsing_full.lua



```lua
for _, match in pairs(response.matches) do
  print(match.threat.url .. " : " .. match.threatType)

  -- Who has contacted that malware?
  for _, flow in pairs(flows.flows) do
    if (flow.host_server_name == match.threat.url) then
      local host = interface.getHostInfo(flow["cli.ip"])

      if host ~= nil then
        -- Check if the administrator has configured a custom host
        local name = getHostAltName(host.ip)

        if isEmptyString(name) then
          -- Otherwise just use its name
          name = host.name
        end

        print(" contacted by " .. host.name .. " (" ..
          discover.devtype2string(host.devtype) .. ")")
      end
    end
  end

  print("\n")
end
```

Final Result



JSON request

```
{"threatInfo":{"threatTypes":
["MALWARE","SOCIAL_ENGINEERING","POTENTIALLY_HARMFUL_APPLICA
TION","UNWANTED_SOFTWARE"],"threatEntryTypes":
["URL"],"platformTypes":["WINDOWS"],"threatEntries":
[{"url":"www.kasterborous.com"}]}}
```

JSON response

```
{
 "matches": [
  {
    "threatType": "MALWARE",
    "platformType": "WINDOWS",
    "threat": {
      "url": "www.kasterborous.com"
```

Malware sites detected

```
www.kasterborous.com : MALWARE contacted by DEKSTOP-ULJ721
```

Thank you for your attention

Happy Scripting

https://github.com/ntop/ntopng