

**TEAM***:* Pentagon

Alexandra Cherry,

Declan Woodham,

Owen Nelson,

Stewart Anderson,

Velislav Velchev

PLugin Creation guide

Version Number: 1.0

Version Date: 10/03/2020

Version History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version Number** | **Implemented By** | **Revision Date** | **Approved By** | **Approval Date** | **Description of Change** |
| 1.0 | Owen  Nelson |  |  |  | The creation of the whole document |

Contents

[2 Introduction 3](#_Toc37627669)

[2.1 Purpose 3](#_Toc37627670)

[2.2 Use 3](#_Toc37627671)

[2.3 Included Plugins 3](#_Toc37627672)

[3 Creating The Plugin Instructions 4](#_Toc37627673)

[3.1 - Creating the Files 4](#_Toc37627674)

[3.2 – Programming 4](#_Toc37627678)

[4 Testing The Plugin 5](#_Toc37627679)

[Appendix A: Standard Plugins 6](#_Toc37627680)

[Appendix B: Key Terms 7](#_Toc37627681)

[Appendix C: Brand Design Specification 7](#_Toc37627682)

# 

# Introduction

## Purpose

The purpose of this document is to assist and inform users through the process of creating their own unique plugins compatible with The Network Guardian. This document will provide linear instructions for the specific example of a Check Network Connectivity plugin, which establishes the process requirements used for creating all future plugins.

## Use

This information will be specific to The Network Guardian and its framework. It can be used to both create and test plugins to be used by the program.

## Included Plugins

The Network Guardian is provided with 8 Plugins, however including the ability for users to create their own allows for more unique and diverse plugins that are relevant to the individual users’ network(s). This also gives users the opportunity to implement plugins from external online sources and have them work on The Network Guardian.

# Creating your first plugin

## 3.1 - Creating the Files

## Navigate to the “plugins” folder within the Network Guardian location.

1. Create a new folder, and name it “Check Internet Connectivity”.

1. Within the Check Internet Connectivity folder, create a new .py file and call it “plugin”.

## Still within this folder, create a new .html file and call it “template”.

## A screenshot of a cell phone Description automatically generated

## 3.2 – Programming

1. Open the “plugin” file created in the last section.
2. Import the standard required functions from the network guardian, as well as any other necessary functions from the relevant directories and modules – for this example, see the image below.

A screenshot of a social media post

Description automatically generated

1. Register the plugin in the Network Guardian, as described in the image below, with your plugin details. (Plugin Name, Plugin Type, Creator, Version)

A screenshot of a social media post

Description automatically generated

1. Create a class with the same name as your plugin, putting “AbstractPlugin” into the parenthesis.

A screenshot of a social media post

Description automatically generated

1. Include the plugin description by creating a docstring.

A screenshot of a social media post

Description automatically generated

1. A screenshot of a social media post

   Description automatically generatedCreate a function called “execute”, and above this include a link to the “template” file created in Part 1. This function will hold the main code for the plugin.
2. A screenshot of a social media post

   Description automatically generatedInside the execute function, declare an array called “urls”. Populate this with multiple URLs to avoid any false positives.
3. Create a loop for the urls array.
4. Inside the loop, create a try, using the “urlopen” function.
5. On the line below, use the “urlopen” function to open the variable “url”. Include a timeout of 5 seconds.
6. Below this, set the URL within the array to true.
7. Create an except using “URLError”.
8. Conclude the except using continue.

A screenshot of a social media post

Description automatically generated

1. Return the array in a variable named “results”.

A screenshot of a social media post

Description automatically generated

# Testing The Plugin

By using test\_plugin, users can test the plugin inside their IDE so they can learn how it works as well as testing its performance while in the process of creating it without having to load the whole program.

1. Comment the register plugin line.
2. Below this, put “@test\_plugin”.
3. Run the build.

# Appendix A: Standard Plugins

The standard plugins that come with Network Guardian are displayed in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Plugin Name** | **Category** | **Supported Platforms** | **Author** | **Version** |
| Internet Connectivity | Networking | Windows, Linux, Darwin | Velislav V. | 1.0 |

This plugin determines whether the local machine has access to the internet. It works by looping through three different URL’s and trying to connect to them and then returning the results back to the user.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Plugin Name** | **Category** | **Supported Platforms** | **Author** | **Version** |
| Netstat Information | Networking | Windows, Linux, Darwin | Owen N. | 1.0 |

This plugin is based on and uses the psutil (process and system utilities) cross-platform library to retrieve information and store it into variables. The plugin tells you about the different connections and information about them such as the protocol, local address, remote address, status, PID and the program name.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Plugin Name** | **Category** | **Supported Platforms** | **Author** | **Version** |
| Network Interface Information | Networking | Windows, Linux, Darwin | Owen N. | 1.0 |

This plugin again is based on and uses the psutil (process and system utilities) cross-platform library. It works by gathering information about all the network interfaces and then it puts it to a list which is then passed on to the table which shows things such as whether the device is online or not, the IP, Broadcast Address, netmask, mac address, packets information, speed, dropped packets and more.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Plugin Name** | **Category** | **Supported Platforms** | **Author** | **Version** |
| System Information | System | Windows, Linux, Darwin | Declan W. | 1.0 |

This plugin returns information about the device it has been run on such as the system name, username, the platform of the device, operating system, processor and memory size and it formats it in a user-friendly format using a table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Plugin Name** | **Category** | **Supported Platforms** | **Author** | **Version** |
| User Enumeration | Enumeration | Windows, Linux, Darwin | Alexandra C. | 1.0 |

This plugin uses the grp module that provides access to the Unix group Database. It is a Unix specific service, so all the methods of this module are available on Unix versions only.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Plugin Name** | **Category** | **Supported Platforms** | **Author** | **Version** |
| Local Firewall Status | Enumeration | Windows, Darwin | Velislav V. | 1.0 |

This plugin uses the grp module that provides access to the Unix group Database. It is a Unix specific service, so all the methods of this module are available on Unix versions only.

# Appendix B: Key Terms

# Appendix C: Brand Design Specification

Acceptable Logo’s:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Logo | Text Logo | Icon |

Heading Fonts: Segoe UI Semibold

Paragraph Fonts: Segoe UI

Font Size: 10px – 14px