Plugin-Daten	
Autor	Christian Fenzl
Logo	
Status	STABLE
Version	0.2.2
Min. LB Version	0.2.3
Release Download	https://github.com/christianTF/LoxBerry-Plugin-Any/archive/0.2.2.zip
Pre-Release Download	https://github.com/christianTF/LoxBerry-Plugin-Any/archive/refs/tags/0.3.1.zip
Beschreibung	Mit dem Plugin können am LoxBerry Shell-Befehle ausgeführt werden.
Sprachen	EN, DE
Diskussion	https://www.loxforum.com/forum/projektforen/loxberry/plugins/124449-plugin-any-plugin-f%C3%BCr-loxberry

Any Plugin

Version History...

Version 0.01

- Entwicklung

Version 0.02

- Lauffähiger Stand

Version 0.04

- Es können im Kommando für die Rückgabe unterschiedliche Miniserver adressiert werden
- Das Plugin prüft und meldet in Klartext, ob der Socket schon belegt ist (weitere Instanz läuft?)
- Ping-Funktion

Version 0.1.1 UNSTABLE

- Erster UNSTABLE Release
- Webinterface fertig

Version 0.1.2 UNSTABLE

- Besseres Service-Handling
- Es wird eine genauere Fehlermeldung ausgegeben, warum kein TCP-Socket geöffnet werden kann
- Jeder Output wird in /opt/loxberry/log/plugins/anyplugin/tcp2shell.log ausgegeben

Version 0.1.3 UNSTABLE

- Logfile-Button im GUI

Version 0.1.4 RELEASE

- Speichere deine eigenen Scripts in /opt/loxberry/data/plugins/anyplugin/commands
- Das ist jetzt der Standardpfad für die Ausführung von Scripts
- Beim Neustart des Services ("Speichern" im WebUI) werden die Scripts automatisch executable (+x) gesetzt.
- Beim Ausführen: "Punkt Slash" vor dem Scriptnamen nicht vergessen!
- Die Scripts werden bei einem Plugin-Update gesichert und wiederhergestellt.
- Wesentlich verbessertes Logging

Version 0.2.0 Pre-Release

- Any-Plugin ist jetzt multithreaded
    - Während Commands ausgeführt werden, bleibt das Plugin für neue Verbindungen verfügbar.
    - Mehrere Commands werden parallel ausgeführt.

Version 0.2.1 RELEASE

- Introduced ***multi-language support*** (use the Translate widget to translate to your own language)
- Basic languages are ***English*** and German
- Removed fields from the form that currently are not available
- Fixed minor form and spelling mistakes.

Version 0.2.2 RELEASE

- Fixes a major bug introduced with 0.2.1 (by removing a form field from the WebIf), that the security_mode=UNSECURE parameter was deleted on WebIf-Save. As currently only the UNSECURE mode is implemented, this leads to deny every command (if you never have saved the config, this did not apply to your installation).
- Enhances trimming of the input line - `\r\n` are now stripped of the input line, making tests with telnet more easy. It tooks me hours to study an own issue using telnet, but telnet also under Linux, telnet closes with \r\n and no command worked.
- Also therefore, enhanced logging to the logfile

Version 0.3.1 Pre-Release

LoxBerry 3.0 deprecates the Plugin Interface 1.0 (for LoxBerry 0.x), therefore it is required to update the plugin to Plugin Interface 2.0. Some very old code was replaced by LoxBerry library functions, so stay calm and create an issue if somethings fail.

- **Fixes** the created log to have loxberry as owner, for LoxBerry Log Maintenance to clean up the log.
- **Fixes** Wiki links to the new LoxBerry Wiki https://wiki.loxberry.de
- **Updated** much code to use LoxBerry SDK features

Artikel-Updates

Lass dich über Updates in diesem Artikel informieren, indem du rechts oben im Menü auf *Beobachtung* klickst.

Sicherheitshinweis

Dieses Plugin ist nur für Benutzer mit Linux-Erfahrung geeignet. Durch die eingeschränkten Möglichkeiten des Loxone Miniservers ist keine ausreichend sichere Kommunikation zwischen Miniserver und LoxBerry möglich.

Trotz der eingebauten Sicherheitsfunktionen (Einschränkung auf Quell-IP-Adressen, Einschränkung auf das eigene Subnet) könnte ein Eindringling mit einem gezielten Angriff (IP-Spoofing) die Kontrolle des LoxBerry's übernehmen und damit Zugriff auf Miniserver-Zugangsdaten erhalten.

Download

Alle Releases: https://github.com/christianTF/LoxBerry-Plugin-Any/releases

Dort beim aktuellsten Release

- entweder den "Source code (zip)" herunterladen, am LoxBerry auswählen und installieren,
- oder den "Source code (zip)"-Link in die Zwischenablage kopieren, diesen in der Pluginverwaltung bei der URL einfügen, und installieren.

Repository bei Github: https://github.com/christianTF/LoxBerry-Plugin-Any

Installation

Das Plugin kann normal installiert werden. Nach der Installation muss der LoxBerry neu gestartet werden. In LoxBerry muss ein Miniserver konfiguriert sein, an den die Antwort gesendet wird.

Funktion des Plugins

Mit dem Plugin wird ein TCP-Port geöffnet, mit dem der Miniserver mit dem Plugin kommunizieren kann.

Über diesen Port können direkt Shell-Kommandos (Phase 1) oder vordefinierte Makros (Phase 2) aufgerufen werden.

Bei jedem Aufruf kann als Parameter übergeben werden, ob der Return-Code des Aufrufs in einen virtuellen Eingang, oder der Output des Aufrufs per UDP übermittelt werden soll.

Senden von Kommandos (Loxone Config)

Virtuellen Ausgang einrichten

Dafür einen *Virtuellen Ausgang* erstellen:

Eigenschaft	Wert	Anmerkung
Bezeichnung und Beschreibung	Nach Belieben	Ist für die Funktion nicht relevant
Adresse	tcp://<loxberry>:9095	<loxberry> ist der Hostname oder die IP des LoxBerry. Der Port ist als Standard 9095. kann aber in der Plugin-Konfiguration geändert werden
Verbindung nach Senden schließen	Aktiv	

Virtuellen Ausgangsbefehl anlegen

Einen virtuellen Ausgangsbefehl anlegen und bei *Befehl bei EIN* das Any-Kommando eingeben.

Die Syntax sieht so aus:

`<Eindeutiger_Name> <Rückgabemodus> command <Shell-Befehl>`

`<Eindeutiger_Name> <Rückgabemodus> macro <Makro-Name>` (Makros noch nicht implementiert!)

Hier die ausführliche Parameterbeschreibung

Parameter		
1.Parameter	`<Name>`	Ein eindeutiger Name, um den Request zu identifizieren. Der Name wird für die Rückgabe verwendet.

2.Parameter	`off`
	`rc`
	`udp`
	`rcudp`
	`rc.2/udp.2/rcudp.2`
		→ Das Ergebnis des Aufrufs wird nicht zurückgeliefert.
		→ Der Exit Code des Aufrufs wird in einen virtuellen Eingang mit der Bezeichnung <Name> geschrieben.
		→ Der Output des Scripts wird per UDP an den Miniserver gesendet.
		→ Der Exit Code wird in <Name> geschrieben und der Output per UDP gesendet.
		→ Bei dieser Angabe wird die Rückgabe an den zweiten (dritten, vierten usw.) Miniserver gesendet.

3.Parameter	`command <Shell-Befehl>`
	`makro <Makroname>`
	`ping`
		→ Der Befehl nach "command" wird komplett am LoxBerry ausgeführt.
		→ Das Makro namens <Makroname> wird ausgeführt. (noch nicht implementiert!)
		→ Entsprechend des Sendetyps wird die aktuelle Epoch-Time am LoxBerry übermittelt.

Beim Rückgabetyp (rc, udp oder rcudp) kann die Sequenz "Punkt Nummer" (z.B. `rc.2` oder `udp.1`)

für die Miniservernummer (von LoxBerry) mitgegeben werden, um jenem die Antwort zu senden. Wird dies nicht angegeben, wird immer der erste Miniserver verwendet.

Beispiel:

```
# Sendet das Directorylisting per UDP
Dir_Listing udp command ls -l

# Sendet den Linux Hostnamen per UDP und setzt den Exit Code des Aufrufs in
den virtuellen Eingang <LB_Hostname>
LB_Hostname rcudp command hostname

# Sendet die Linux Epoch-Zeit an den virtuellen Eingang <LB_Epoch>
LB_Epoch rc.1 ping

# Sendet die Zeit in Sekunden seit dem letzten Miniserverbackup (Plugin) per
UDP an den _dritten_ Miniserver (001_MSOG ist das Backupverzeichnis)
LB_Hostname udp.3 command echo $(($(date +%s) - $(date +%s -r
/opt/loxberry/webfrontend/html/plugins/miniserverbackup/files/001_MSOG)))
```

Konfigurationsoptionen

In der Plugin-Konfiguration kann definiert werden, welche Sicherheitsstufe vom Plugin angewendet wird:

Security Mode (security_mode=)	**Beschreibung**
UNSECURE (default)	Shell-Kommandos können direkt übergeben werden und werden ausgeführt. Das ist unsicher, da am LoxBerry jeglicher Befehl ausgeführt werden kann. (PHASE 1)
RESTRICTED	In der Plugin-Konfiguration am LoxBerry werden Shell-Befehle als Makros vordefiniert. Nur diese Makros dürfen ausgeführt werden. (PHASE 2)

Authentifizierung (authentication=)	**Beschreibung**
off (default)	Es gibt keine Authentifizierung. Alle Kommandos werden entsprechend des Security Mode ausgeführt.
on	Eine 2-Phasen-Authentifizierung soll sicherstellen, dass der Aufruf tatsächlich vom Miniserver kommt. IDEEN Wer hier eine gute Idee hat, die aus Ausführen von TCP-Befehlen in Loxone nicht sehr verkompliziert, bitte melden!

Auf Subnet einschränken (restrict_subnet=)	**Beschreibung**
off	Das Subnet des Clients wird nicht geprüft.
on (default)	Nur Clients aus dem gleichen Subnet dürfen sich anmelden.

Auf IPs einschränken (`allowed_remote_ips=`)	**Beschreibung**
leer (default)	Keine Einschränkung auf einzelne IPs
<IP1>, <IP2>, <IP3>	Eine mit Komma getrennte Auflistung von IP-Adressen (z.B. 192.168.1.77, 192.168.1.78). Der Filter sollte grundsätzlich auch mit IPv6 funktionieren, ist aber damit noch nicht getestet.

Eigene Scripte erstellen und ausführen

Das Plugin erzeugt bei der Installation das Verzeichnis /opt/loxberry/data/plugins/anyplugin/commands (bzw. von Windows aus: `\\loxberry\loxberry\data\plugins\anyplugin\commands`).

In diesem Verzeichnis können eigene Shell-Scripts erstellt werden - das Verzeichnis wird bei einem Plugin-Update gesichert.

Der Pfad ist auch der Standardpfad, wenn kein Pfad bei der Ausführung mitgegeben wird. Achte deswegen darauf, dass du `./deinscript.sh` angibtst (Punkt Schrägstrich), damit das Script ausgeführt wird. Bei Scripts in anderen Pfaden musst du den vollen Pfad beim Aufruf mitgeben.

Da die Dateien im commands-Verzeichnis, von Windows aus erstellt, nicht executeable sind, setzt das Service bei jedem Neustart alle Scripts auf Executeable (+x). Dafür musst du nur "Speichern" im WebUI drücken.

Beim Bearbeiten von Scripts von Windows aus beachte, dass der Zeilenumbruch unter Linux anders ist. Ich empfehle zum Bearbeiten von Scripts unter Windows den Editor Notepad++. Im Menü *Bearbeiten* / *Format Zeilenende* kannst du *Konvertiere zu UNIX* ausführen, um das richtige Format zu haben. Stelle den Zeichensatz einer neuen Datei außerdem auf UTF-8 (*UTF-8 ohne BOM*).

Roadmap

Phase 2 wird erst fortgesetzt, wenn LoxBerry 0.3 released it, und wenn zum Plugin mehr Feedback da ist. Fehler werden gefixt.

PHASE 1

- DONE Direkte Aufrufe von Shell-Commandos per TCP (UNSECURE mode).
- DONE Rückgabe des Exit-Codes
- DONE Rückgabe des Aufruf-Outputs per UDP
- DONE Webinterface zur vollständigen Konfiguration des Plugins

PHASE 2

- DONE Sicherheit: Einschränkung auf Quell-IP-Adressen
- DONE Sicherheit: Einschränkung auf das eigene Subnet
- DONE Sicherheit: Wahlweise Ausführung als User loxberry oder root

- OPEN Konfiguration von Makros am LoxBerry möglich (RESTRICTED mode)
- OPEN Service-Healthcheck direkt am LoxBerry
- DONE Service-Healthcheck vom Miniserver aus ("ping")
- OPEN Parsen von #Name# and #Value# in Makro-Definitionen zum Einsetzen des Namens und des übergebenen Analogwertes

PHASE 3

- DONE Ausführung der Commands als eigener Prozess (fork)
- OPEN Sicherheit: Authentifizierung mittels Handshake.
- OPEN Erweiterung der Makrofunktion um Variablen → Im Makro können mehrere Loxone-Werte ausgewertet und im Befehl verwendet werden
    - #TempWohnzimmer# liest das Loxone-Element "TempWohnzimmer" und setzt es als Wert in den Aufruf ein
    - #MS2.TempWohnzimmer# liest das Loxone-Element "TempWohnzimmer" vom zweiten Miniserver

Fragen stellen und Fehler melden

Fragen stellen im LoxForum
Thread: https://www.loxforum.com/forum/projektforen/loxberry/plugins/124449-plugin-any-plugin-f%C3%BCr-loxberry

Fehler melden im GitHub Repo: https://github.com/christianTF/LoxBerry-Plugin-Any/issues

From:
https://wiki.loxberry.de/ - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:
https://wiki.loxberry.de/plugins/any_plugin/start

Last update: **2023/01/10 17:55**