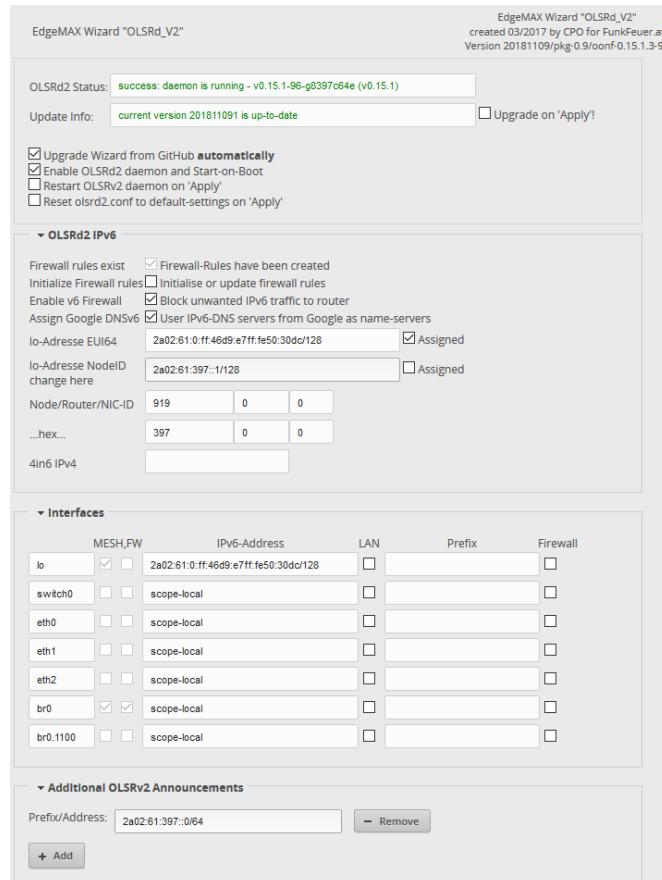


OLSRd V2 Wizard

funkfeuer.at

Dokumentation basiert auf Version

2018.11.19



Inhalt

Installation.....	2
Default-Konfiguration von olsrd2.....	2
Auswahl der Mesh-Interfaces	3
Wizard-IPv6-Firewall-Rules	3
Quelle und Source	3
Feldbeschreibung der Wizard-Elemente.....	3
Top-Box Basic Infos.....	3
OLSRd2 IPv6.....	3
Interfaces.....	4
Additional OLSRv2 Announcements.....	4
Funktionsumfang des Wizards – Einschränkungen.....	4
Ändern von olsrd2 Mesh-Interfaces.....	4
Konfiguration von "4in6" im Sinne des v642-Konzepts.....	4
Definition von mehreren LAN Interfaces	5
Konfiguration von spezifischen Timings.....	5
Konfiguration von spezifischen Metrics	5
Statusanzeige und OLSRv2-Linkpartner	5
OLSRv2-Link via OpenVPN-Tunnel.....	6
Technische Details.....	6
Access-List für http:8000 und telnet:2009 plugins.....	6
Logfiles.....	6
DNS und FQDNs für IPv6-Adressen	7
OLSRv2 Topologie und Wien-Map.....	7
Installation via CLI.....	7
Known Bugs	8
Uninstall/Removal	8
Weiterführende Links.....	8

Installation

Das Wizard tar.gz-File kann über die EdgeOS-eigene Web-UI normal installiert werden: Wizards → [+] → Name: "OLSRd_V2", File auswählen und nach Upload "Save" klicken. Ein aktuelles Release findet sich idR als Attachment hier: https://github.com/pocki80/ER-wizard-OLSRd_V2/releases/latest.

Die Installation kann alternativ auch via CLI erfolgen (benötigt Internetverbindung): siehe Kapitel "[Installation via CLI](#)".

Nach dem 1. Öffnen des Wizards im UI installiert der Wizard das OLSRd2-Paket und macht ggf. eine Erstkonfiguration – das kann 5 bis 10 Sekunden dauern.

Bei einem "Clean Install" (kein olsrd2 war zuvor installiert) erledigt die Install-Route folgende Ersteinrichtung:

- IPv6-Adressen beider Google-DNS werden eingetragen
- Default-Config: olsrd2-mesh interfaces: lo, br0
- Olsrv2-Daemon wird sofort gestartet (nur wenn am Interface "lo" eine IPv6-adresse schon konfiguriert ist)
- Autostart wird aktiviert, wenn kein olsrd6 (olsrd-v1 für IPv6) aktuell läuft

Nach der Erstinstallation sind noch folgende Schritte nötig, um den OLSRd2 zu starten:

- IPv6-Adresse für Interface "lo" konfigurieren – eine von folgenden 2 Optionen nutzen:
 1. EUI64-Adresse
diese IP wird aus der Seriennummer des Routers ermittelt und kann nicht geändert werden. Das Eingabefeld ist read-only und immer befüllt, sofern die Router-S/N erkannt wurde. Die Checkbox "Assigned" hinter dem Feld zeigt an/steuert, ob diese IPv6-Adresse dem lo-Interface zugeordnet ist bzw. werden soll.
 2. NodeID-IPv6-Adresse aus dem Userblock
diese IP enthält die NodeID laut Redeemer als Hexadezimale. Der Wizard versucht die NodeID über die IPv4-Connectivity im Onlinefall selbst zu erkennen; die NodeID wird in den beiden Feldern unter der IP-Adresse angezeigt sofern erkannt (dezimal und als hexadezimale). Schlägt die automatische Erkennung fehl (weil Offline, weil keine IPv4-Connectivity, weil genutzte olsr1/ipv4-adresse von einem Map-Hidden-Knoten genutzt wird), wird "0" angezeigt.
Ist dem lo-Interface bereits eine IPv6-Adresse aus dem Userblock zugeordnet, zeigen die Detailfelder unter der IPv6 die Werte von NodeID, Router-ID und Nic-ID an.
Das Feld für die NodeID-IPv6-Adresse ist editierbar: man kann die Adresse (insbesondere zum Zwecke der Definition von Router-ID, Nic-ID und nötigenfalls Node-ID) ändern und so die IPv6-Adresse (=olsr2-originator) wunschgemäß definieren.
- Checkbox "Restart olsrd2 on apply" aktivieren
- Apply clicken

Default-Konfiguration von olsrd2

- Interfaces: lo, br0 (Annahme: olsrv1-Wizard läuft über br0, daher ebtables-P-DROP gesetzt)
- Mesh-Port: 269
- telnet: port 2009, only localhost
- www: port 8000, only Funkfeuer-IP-Ranges, localhost, RFC1918 private IPv4 ranges
- Default-Metric: rx_bitrate = 10M

Auswahl der Mesh-Interfaces

Per Default ist das Interface "br0" für olsrd2 aktiviert, weil in den meisten Anwendungsfällen bisher auch der OLSRd_V1-Wizard läuft und alle Mesh-Interfaces in einer gemeinsamen "client-isolated" Bridge zusammengefasst liegen. Der OLSRd_V1-Wizard erledigt die Client-Isolation für die Olsr(v1)-aktivierte Bridge mittels "ebtables -P drop".

Läuft auf dem Router kein olsr(v1), können die einzelnen eth-Ports auch direkt (also unbridged) und einzeln für OLSRv2 aktiviert werden (siehe Kapitel "[Ändern von olsrd2 Mesh-Interfaces](#)").

Wizard-IPv6-Firewall-Rules

- WebPorts allowed to LOCAL:
10,22,80,81,443,2006,2007,8000,8080,8081,9090,9091,8443,10883
- Allowed to LOCAL: olsrd6@698, olsrv2@269, snmp@161, icmp/traceroute/ping
- Allowed to LAN: icmp/traceroute/ping, state: established/related

Quelle und Source

https://github.com/pocki80/ER-OLSRd_V2 → Paket mit olsrd2 binary und default-config

https://github.com/pocki80/ER-wizard-OLSRd_V2 → Wizard mit eingebettetem Paket

Feldbeschreibung der Wizard-Elemente

Top-Box Basic Infos

- OLSRd2 Status: success, daemon is running, <>version/commit>>
Beim UI-Laden wird versucht, die aktuelle Version auf dem GitHub-Repo zu erkennen. Ist diese neuer als die aktuell installierte Version des Wizards, wird ein Update vorgeschlagen. Die Versionsnummern sind als Datum des Commits aufgebaut mit einer anschliessenden fortlaufenden Nummer, z.B. 201811194 (19.Nov.2018, fünfte Tagesversion). Die Anzeige kann nur bei einer funktionierenden Internetconnectivity erfolgen.
- Upgrade on Apply
Im Falle eines verfügbaren Updates kann dieses geladen und sofort installiert werden. Das Update kann nur bei einer funktionierenden Internetconnectivity erfolgen.
- Upgrade Wizard from GitHub automatically
Der tägliche Cronjob um 06.25am prüft auf eine neuere Wizard-Version im GitHub-Repo: wenn verfügbar und für das AutoUpdate auch freigeschalten, wird diese automatisch heruntergeladen und installiert.
- Enable OLSRd2 daemon and Start-on-Boot
Wenn aktiviert wird das Startup-Script /config/scripts/post-config.d/start-4in6.sh abgelegt, das nach einem Bootvorgang (auch nach einem Firmware-Upgrade) den olsr2-daemon startet.
- Restart OLSRv2 daemon on Apply
Startet den olsrd2 neu (bei Konfig.-Änderungen erforderlich)
- Reset olsrd2.conf to default-settings on Apply
Löscht die olsrd2.conf mit olsrd2_default.conf und trägt die im Wizard eingebauten UI-Feldinformationen neu ein (LAN, Firewall, attached-network announcements). Manuell im CLI gemachte Änderungen der olsrd2.conf gehen verloren.

OLSRd2 IPv6

- Firewall rules exist Firewall-Rules have been created
Zeigt an, ob die Firewall-Names lan_out0xff-6, wan_local0xff-6 und Portgroup WebServices0xFF angelegt wurden
- Initialize Firewall rules Initialise or update firewall rules
Legt die Policies für die IPv6-Firewall an oder aktualisiert sie aus der aktuellen Wizardversion.
- Enable v6 Firewall Block unwanted IPv6 traffic to router
Aktiviert die Firewall LOCAL-IN, beschränkt also den Zugriff zum Router-OS.
- Assign Google DNSv6 User IPv6-DNS servers from Google as name-servers
Konfiguriert 2001:4860:4860::8888 und ::8844 als (zusätzliche) Nameserver

- Io-Adresse EUI64 [2a02:61:0:ff:..../128] [__] Assigned
Ermittelt und konfiguriert die IPv6-Adresse im EUI64-Format
- Io-Adresse NodeID (change here) [2a02:61:397:1ae1::1/128] [__] Assigned
Zeigt die vorhandene oder vorgeschlagene IPv6-Adresse aus dem "User-Block" an. Die Adresse kann in diesem Eingabefeld auch angepasst werden. Für ein aktivierte LAN-Interface wird dieser Prefix mit Länge /64 verwendet. 2a02:61: ist dabei fix beizubehalten, das Bytepaar danach (hier 0397) repräsentiert die NodeID laut Redeemer. Das letzte Bytepaar in diesem /64-Prefix (hier 1ae1) darf frei gewählt werden und ist also "Router-ID" (erste 2 Bit) und NetworkInterfaceID (letzte 2 Bit) konzipiert.
- Node/Router/NicID: [919][18][225] decimal
Zeigt die dezimale Knoten-ID aus dem Redeemer an (aus einer vorhandenen IPv6-Userblock-Adresse ausgelesen, oder falls keine IPv6 konfiguriert ist, wird falls möglich die NodeID automatisch erkannt). Router-ID und Nic-ID können nur aus einer vorhandenen User-Block-IPv6-Adresse ausgelesen werden.
- Node/Router/NicID: [397][1a][e1] hex
Zeigt die darüberstehenden 3 Kennzahlen als Hexadezimale an (also so wie diese aus der IPv6-Adresse rausgeschnitten werden)
- 4in6 IPv4 [__]
Zeigt eine erkannte v642-4in6-IPv4-Adresse an, z.B.: 185.194.20.60.

Interfaces

- Interface Name: lo, br0, eth0,...
- MESH: [x] olsrd2 für dieses interface ist aktiviert (default: lo, br0), read-only
- FW: [__] firewall LOCAL-IN für dieses Interface ist aktiviert, read-only
- IPv6-Address: IPv6 Adresse auf diesem Interface (nur die erste wird angezeigt), read-only
- LAN: [__] aktiviert für (ein, 1) LAN-Interface: setzt eine Inside-IPv6-Adresse ::51de:b00b/64, aktiviert Prefix-Delegation und kündigt diesen Prefix auch im Olsrd2-Announcement an. Der Prefix kann im Feld "Io-Adresse NodeID" definiert werden. Bei einer Änderung: olsrd2 neu starten
- Firewall [__]: firewall LAN-OUT für dieses Interface ist aktiviert

Additional OLSRv2 Announcements

- Prefix/address: [__]: Hier können zusätzliche (über das oben für LAN benötigte Prefix hinausgehende) Prefixe/HNAs/Attached-Network-Prefixes eingetragen werden. Bei Änderungen: olsrd2 restarten

Funktionsumfang des Wizards – Einschränkungen

Der Wizard ist noch unfertig und unterstützt folgende Vorgänge noch nicht:

Ändern von olsrd2 Mesh-Interfaces

Nur die Anzeige von aktivieren Mesh-Interfaces im UI ist implementiert. Sofern die defaults (lo, br0) nicht genutzt werden, muss man im CLI die Settings direkt in der File /config/olsrd2/olsrd2.conf vornehmen/ergänzen.

Konfiguration von "4in6" im Sinne des v642-Konzepts

Wird der Router ohne olsr(v1)/IPv4-Connectivity genutzt und soll über einen 4in6 Wrapper laufen, muss das noch manuell konfiguriert werden. Der Wizard zeigt nur eine eventuell eingerichtete IPv4-Adresse aus der Range 185.194.20.0/23 an.

Beispiel (IP-Adressen hier: https://wiki.funkfeuer.at/wiki/Projekte/v642/IPv4_Liste eintragen und im Code-Snippet entsprechend **anpassen**: das letzte Bytepaar **1460** steht hier für die hinteren zwei umgerechneten IPv4-Octets 185.194.**20.60**: $20_{10}=14_{16}$ und $60_{10}=3C_{16}$)

1. Die IPv6 als "additional LAN announcement" im Wizard ankündigen (für sofortige Wirksamkeit den olsrd2 restarten): 2a02:61:0:ee:**1**:ffff:b9c2:**1460**/128

2. CLI: /config/scripts/post-config.d/start-4in6.sh erstellen:

```
#!/bin/sh
ip=185.194.20.60
ip -6 addr add 2a02:61:0:ee:1:ffff:$ip/128 dev lo preferred_lft 0
ip -6 tunnel add wrap1 mode ipip6 remote 2a02:61:0:ee:0:ffff:$ip local 2a02:61:0:ee:1:ffff:$ip dev lo
ip -6 link set dev wrap1 up
ip -4 addr add $ip/32 dev wrap1
ip -4 route add 0.0.0.0/0 via $ip dev wrap1 metric 4
exit 0
```

3. Router rebooten oder das erstellte start-4in6.sh einmalig ausführen

Zum Stoppen des 4in6-Wrappers kann man das start-4in6.sh nach /config/user-data verschieben, das Announcement von olsr2 entfernen und rebooten. Ohne Reboot geht es durch entfernen der Zuweisungen in umgekehrter Command-Reihenfolge:

```
ip=185.194.20.60
ip -4 route del 0.0.0.0/0 via $ip dev wrap1 metric 4
ip -4 addr del $ip/32 dev wrap1
ip -6 link set dev wrap1 down
ip -6 tunnel del wrap1
ip -6 addr del 2a02:61:0:ee:1:ffff:$ip/128 dev lo
```

Smokeping für alle v642-Ziele (sehr langsam, benötigt eventuell 2-3 Requests für einen Erfolg):

<http://smokeping.funkfeuer.at/smokeping/housing/?target=v642>

Definition von mehreren LAN Interfaces

Nur die Anzeige der genutzten LAN Interfaces und Inside-Adressen ist im UI implementiert. Werden mehrere IFs dafür benötigt, muss die Inside-IPv6-Adresse und das prefix-advertisement im Config-Tree manuell konfiguriert werden UND der verwendete LAN-Prefix als "additional LAN announcement" im Wizard eingetragen werden.

Konfiguration von spezifischen Timings

Settings direkt in der File /config/olsrd2/olsrd2.conf vornehmen/ergänzen

Konfiguration von spezifischen Metrics

per default ist rx_bitrate=10M gesetzt, um andere Interfaces oder Link-Nachbarn (basieren auf deren MAC-Adresse) sowohl besser (>10M) als auch schlechter (<10M) zu machen. Diese Settings muss man derzeit Settings direkt in der File /config/olsrd2/olsrd2.conf vornehmen/ergänzen.

Mit dem Key "l2neighbor" in der Sektion eines Interfaces kann man auch bestimmte Nachbarn spezifisch setzen durch Angabe der Mac-Adresse. Beispiel:

```
[interface=br0]
l2neighbor rx_bitrate 50000000 80:2a:a4:f5:6c:44
```

Die MAC-Adressen der eigenen Nachbarn findet man im infoplugin unter "nhdpinfo-link", Details im Kapitel "[Statusanzeige und OLSRv2-Linkpartner](#)"

Statusanzeige und OLSRv2-Linkpartner

Der Wizard selbst zeigt nur den Daemon-Status und die Version. Daten über die Topologie oder Linkpartner werden nicht im Wizard angezeigt.

Im Wizard "ER-Wizard-WSLE" (0xFF WebStatus and Let's Encrypt) ist eine Status-Seite implementiert, die von OLSR(v1) und auch OLSR2 die jeweiligen Linknachbarn, Routen und Metriken anzeigt. Quelle: <https://github.com/pocki80/ER-wizard-0xFF-WSLE/releases/latest>

OLSRv2 Status								
Übersicht		Status		Kontakt		78.41.119.41 - router.wehr24.wien.funkfeuer.at		
OLSRd2 Version 0.15.1 v0.15.1-96-g8397c64e								
OLSRd2 Uptime	13:45:00	700	1	5.263M				
Originator	2a02:61:0:ff:46d9:e7ff:fe5f:6bee							
IPv6 Default-Route	hostname UNKNOWN	(2a02:61:0:ff:822a:a8ff:fe5f:7b05)						
IPv6 OLSR2-Links								
IF	Remote IPv6	Remote Hostname	Remote MAC	Metric-In	Metric-Out	routes	nodes	
br0	2a02:61:0:ff:822a:a8ff:fe5d:fa99	router2.wehr24.wien.funkfeuer.at	80:2a:a8:5d:fa:99	10.034Mbit/s	10.034Mbit/s	1	1	
br0	2a02:61:0:ff:822a:a8ff:fe5f:7b05	(unknown)	80:2a:a8:5f:7b:09	10.034Mbit/s	93.368Mbit/s	127	81	
br0	2a02:61:0:ff:46d9:e7ff:fe50:30dc	router.luxi122home.wien.funkfeuer.at	44:d9:e7:50:30:dc	10.034Mbit/s	10.034Mbit/s	2	1	
br0	2a02:61:a9e:105::20	rpi.wehr24.wien.funkfeuer.at	b8:27:eb:8e:ce:61	10.034Mbit/s	10.034Mbit/s	1	1	
br9	2a02:61:0:ff:5054:ff:fe:f3:4696	v33-mesh.ts2018.wien.funkfeuer.at	0a:fd:79:9b:35:85	1.02kbit/s	990.536kbit/s	0	0	
Trace to UPLINK								
#	IP Address	Hostname		Ping				
1	2a02:61:0:ff:822a:a8ff:fe5f:7b05	router.2355RIAG.wien.funkfeuer.at		1.678ms				
2	2a02:61:0:ff:feec:daff:fe4d:248f	router.2345faike.wien.funkfeuer.at		7.162ms				
3	2a02:61:b80:7	ge2-r01faike.falke-bb.wien.funkfeuer.at		7.017ms				
4	2a02:60:35:492:1:ffff:0:1	r01krypta.bb.funkfeuer.at		13.277ms				

Page generated in 2.93sec with Python.

Im CLI kann man aus einem telnet-socket Statusabfragen machen, z.B.:

```
echo /systeminfo version | nc 127.0.0.1 2009
echo /nhdpinfo head link_addr | nc 127.0.0.1 2009
echo /nhdpinfo link_addr | nc 127.0.0.1 2009
```

Im Browser kann man über das http-plugin Statusabfragen machen, z.B.:

```
http://<<router-ip>>:8000/telnet/systeminfo%20version
http://<<router-ip>>:8000/telnet/nhdpinfo%20json%20link_addr
```

OLSRv2-Link via OpenVPN-Tunnel

Besteht noch keine direkte Verbindung zur restlichen OLSRv2-Topologie, bietet auch der Tunnelserver via OpenVPN eine OLSR2-Anbindung. Das (eigene) OpenVPN-Interface muss dazu ggf. in die olsrd2.conf als Mesh-Interface hinzugefügt werden; eine zusätzliche IPv6-Adresse ist nicht erforderlich.

Details zur Einrichtung einer OpenVPN-Tunnelanbindung:

https://wiki.funkfeuer.at/wiki/OpenVPN_mit_EdgeRouter_X-SFP

Status Tunnelserver, OpenVPN-Ports, olsrd, olsrd6 und olsrd2:

<http://tunnel.wien.funkfeuer.at/>

Technische Details

Access-List für http:8000 und telnet:2009 plugins

Das Telnet-Interface auf Port tcp:2009 ist auf localhost gebunden, funktioniert also nur via 127.0.0.1 oder ::1

Das http-Plugin auf Port tcp:8000 kann nur lokal, aus RFC1918-LANs und von Funkfeuer-IP-Adressen angesprochen werden. Die Accesslist ist in olsrd2.conf festgelegt:

```
acl default_reject
acl first_reject
acl 127.0.0.1
acl 10.0.0.0/8
acl 172.16.0.0/12
acl 192.168.0.0/24
acl 78.41.112.0/21
acl 193.238.156.0/22
acl 185.194.20.0/22
acl ::1
acl 2a02:60::/29
```

Logfiles

/var/log/0xffffolsrd2.log enthält debug-infos aus dem Wizard (erstellt von wizard-run)

/var/log/olsrd2.log ist die Logfile des olsrv2 daemons

DNS und FQDNs für IPv6-Adressen

Derzeit gibt es keinen Name-Service von Funkfeuer-Wien, der ein Lookup von IPv6-Adressen von Freenet-Routern beinhaltet. Um eine Zuordnung eines OLSRv2-Routers und seiner IPv6-Adresse zu einem Knoten (und ev. Device-Namen) und somit zu einem Besitzer/Verwalter/Zuständigen zu ermöglichen, gibt es aktuell also nur folgende Anhaltspunkte:

- Der selber Router betreibt auch olsr(v1) mit einer Freenet-IPv4-Adresse (und diese olsr(v1)-IPv4-Adresse ist einem Knoten zugeordnet, der in der Map existiert)
- Der Router nutzt eine Originator-IPv6-Adresse aus dem User-Block (beinhaltet die NodeID, dessen Knoten in der Map existiert)
- Der Router kündigt einen Prefix aus dem User-Block als Attached-Network an (mit NodeID, dessen Knoten in der Map existiert)
- Der Router nutzt eine v642 4in6 Adresse, die auf der Wiki-Seite https://wiki.funkfeuer.at/wiki/Projekte/v642/IPv4_Liste einem Knotennamen zugewiesen ist (und dieser Knotennamen existiert in der Map)

Wer gerne ein (lokales) Namelookup nutzen möchte, kann die routereigene File `/etc/hosts` mit den bekannten Funkfeuer-IPv6-Adressen und -Hostnames befüllen. Ein entsprechendes Shellskript dazu gibt es für die erstmalige Befüllung sowie einen Update-Vorgang:

```
curl -sS https://gist.githubusercontent.com/pocki80/fc6ede395e04dde62ce2199a8281261f/raw/load_ip6_hostsfile.sh  
cat load_ip6_hostsfile.sh | sudo bash; rm load_ip6_hostsfile.sh
```

OLSRv2 Topologie und Wien-Map

...kann (derzeit, Stand November 2018) unter folgenden Links eingesehen werden (Voraussetzung: Login mit Redeemer-Credentials):

Topologie: <https://ff.cybercomm.at/monitor/olsr2.php>

Map: <https://ff.cybercomm.at/map2/>

Installation via CLI

Sofern der Router bereits aufs Internet zugreifen kann, ist mir folgendem Codesnippet eine Wizard-Installation auch in der Shell möglich (sudo erforderlich):

```
#install a wizard taken directly from github  
sudo -s  
name="OLSRd_V2"  
gitowner="pocki80"  
gitrepo="ER-wizard-OLSRd_V2"  
gitbranch="master"  
wizard="/config/wizard/feature/"$name  
url="https://codeload.github.com"  
curl -sS -o /tmp/ER-wizard.tar.gz --connect-timeout 2 --speed-time 2 $url/$gitowner/$gitrepo/legacy.tar.gz/$gitbranch  
tar -xvzf /tmp/ER-wizard.tar.gz -C /tmp >/tmp/result.txt  
rm -f /tmp/ER-wizard.tar.gz  
[ ! -d $wizard ] && mkdir $wizard  
while read LINE; do  
    set $LINE  
    if [ $(basename $LINE) == "wizard-run" ] || [ $(basename $LINE) == "wizard.html" ]  
    || [ $(basename $LINE) == "validator.json" ]; then W=1; else W=2; fi  
    [ -f "/tmp/"$LINE ] && [ $W == "1" ] && cp "/tmp/"$LINE $wizard"/$(basename $LINE)"  
done < /tmp/result.txt  
rm /tmp/result.txt  
rm -Rf /tmp/$gitowner-$gitrepo*  
chmod 755 $wizard"/wizard-run"  
chown www-data $wizard"/wizard-run" $wizard"/wizard.html" $wizard"/validator.json"  
$wizard"/wizard-run load  
exit 0
```

Mit diesem Code kann man auch andere Wizards von GitHub direkt laden: einfach die anfänglichen Variablen name, gitowner, gitrepo entsprechend anpassen.

Anmerkung: der Code fängt keine Fehler ab.

Known Bugs

- olsrd2: die Binary von Commit v0.15.1-96-g8397c64e liefert im systeminfo-plugin falsche Uptime-Zähler. Der Wert ist um den Faktor 333 zu hoch: also durch 333 dividieren
- UI: Beim Anwenden von LAN und LAN-Firewall für wlan-interfaces kann es in bestimmten Konstellationen vorkommen, dass anstatt des VLANs das native Interface aktiviert wird.

Uninstall/Removal

Es ist derzeit keine Uninstall/Cleaning Prozedure vorhanden. Eine Deinstallation muss "händisch" vorgenommen werden:

1. Entfernen des Wizards

Im UI: Wizard öffnen und ganz unten "Aus Liste entfernen" klicken. Zuvor eventuelle LAN- oder Firewall-settings bei den Interfaces entfernen und nochmals "Apply" klicken

2. Olsrd-paket deinstallieren

CLI: sudo kill \$(pgrep -x olsrd2); sudo dpkg -P 0xff-er-olsrd-v2

3. Persistent Files entfernen

CLI: sudo rm -r /config/scripts/post-config.d/startolsrd2.sh /config/olsrd2

4. Stoppen/dekonfig vom 4in6-Wrapper vornehmen (siehe oben), Startskript entfernen:

CLI: sudo rm /config/scripts/post-config.d/start-4in6.sh

IP-Adresse im Wiki "freigeben": https://wiki.funkfeuer.at/wiki/Projekte/v642/IPv4_Liste

5. Konfiguration von IPv6-Elementen und Firewallsettings entfernen (Interfaces und Adressen entsprechend einsetzen, Eingabe von "?" liefert mögliche Werte)

```
configure
delete interfaces loopback lo address 2a02:61:0:ff:xxxx:xxff:fexx:xxxx/128
delete system name-server 2001:4860:4860::8888
delete system name-server 2001:4860:4860::8844
delete interfaces bridge br0 firewall local ipv6-name wan_local0xff-6
delete interfaces ethernet eth1 firewall out ipv6-name lan_out0xff-6
delete firewall group port-group WebServices0xFF
delete firewall ipv6-name wan_local0xff-6
delete firewall ipv6-name lan_out0xff-6
delete interfaces ethernet eth1 ipv6 router-advert
delete interfaces ethernet eth1 address 2a02:61:nnnn:rrii::51de:b00b/64
commit;save;exit
```

Weiterführende Links

<https://wiki.funkfeuer.at/wiki/Projekte/v642>

<https://github.com/OLSR/OONF>