

Linux

Verschiedene Grundlagen zum Linux Kernel bzw. insbesondere zu diversen Linux Distributionen. SystemD, Autostart, Desktopumgebungen, SSH, ...

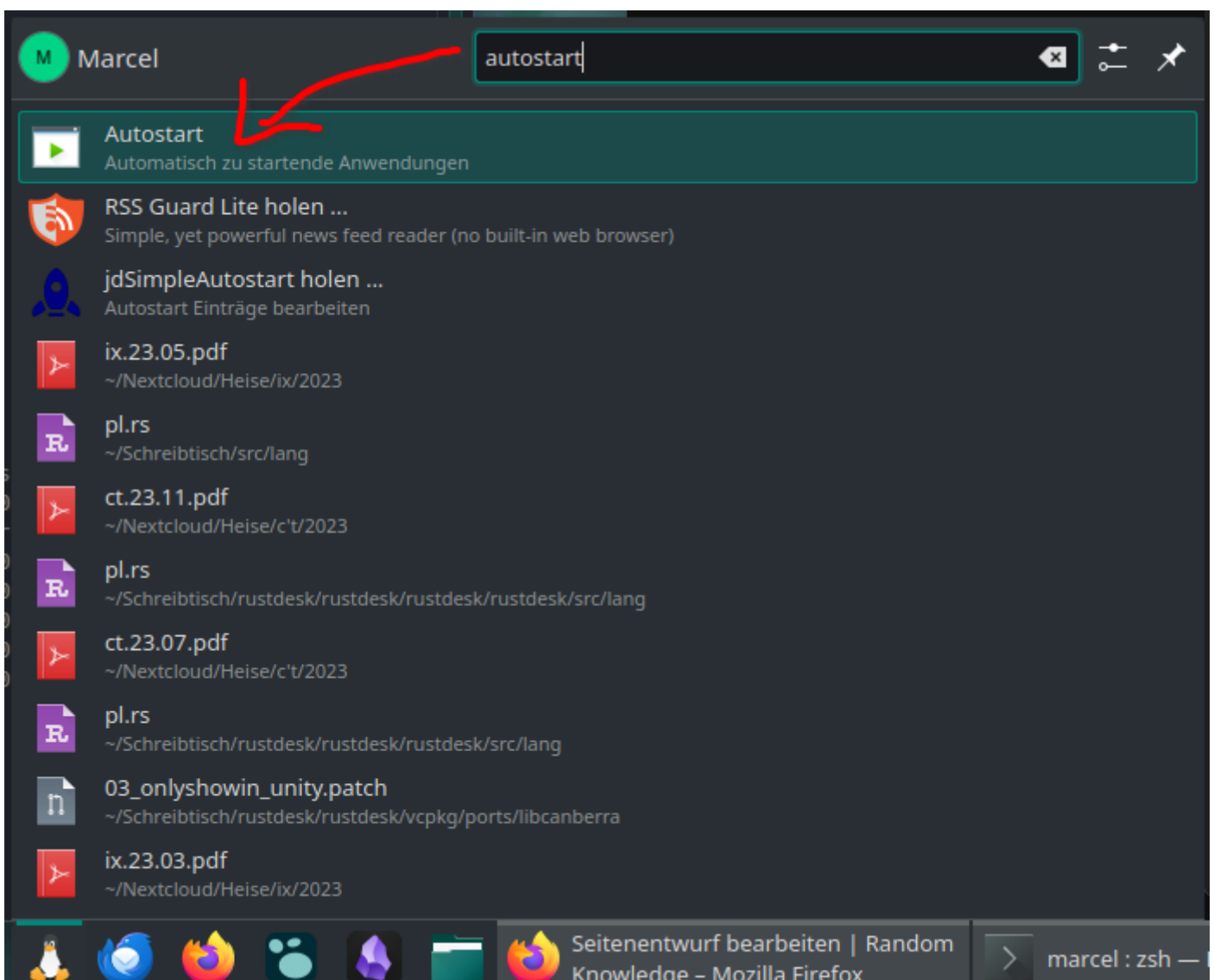
- [Autostart](#)
- [Dateien live einlesen](#)
- [Automatische Updates](#)

Autostart

Je nach Linux Distribution stehen verschiedene Möglichkeiten zur Auswahl, um Programme automatisch zu starten

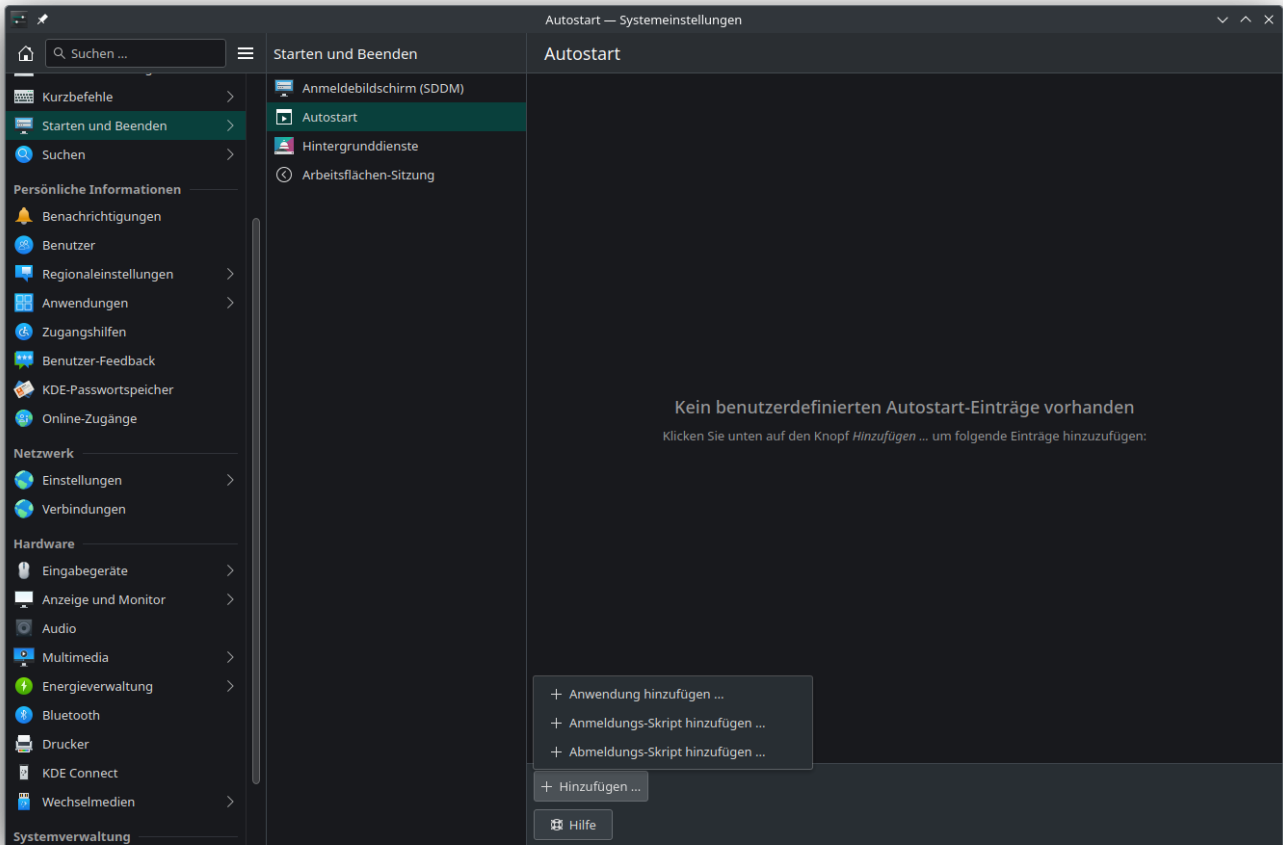
GUI (z. B. KDE, Gnome, Xfce)

Bei Distributionen mit einer grafischen Desktopumgebung wie z. B. Ubuntu (Gnome), Linux Mint (KDE, Xfce, Cinnamon, Mate) oder Debian (KDE) gibt es in der Regel ein Programm mit dem Namen Autostart.

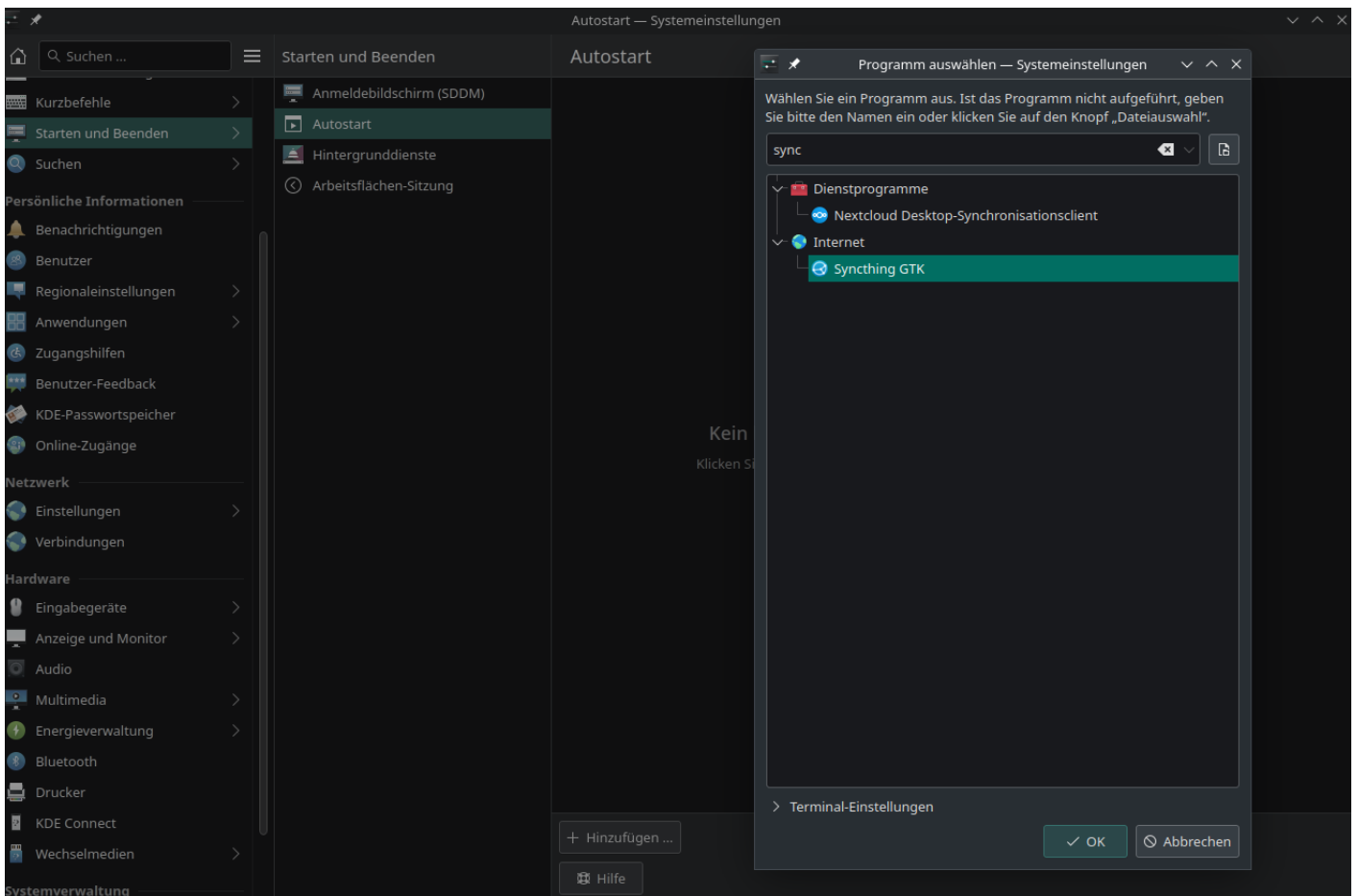


In dem Einstellungsfenster können nun für den aktuellen Benutzer Programme hinterlegt werden, welche beim Anmelden automatisch gestartet werden sollen. Dazu einfach auf das Pluszeichen oder *Hinzufügen* klicken. Es können nicht nur Programme ausgewählt werden, sondern auch An- und Abmelde Skripte, die ausgeführt werden sollen. Der folgende Screenshot zeigt das Menü

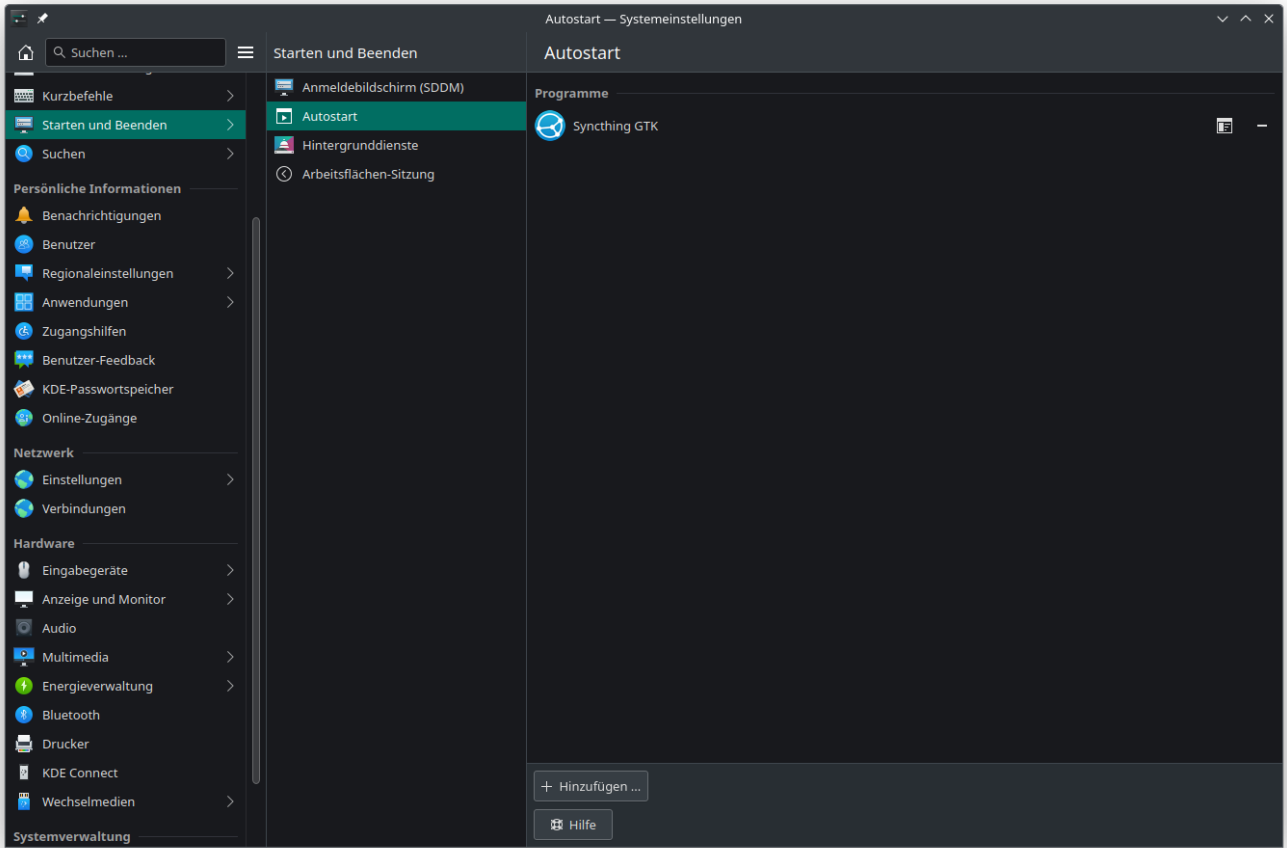
Autostart der Distribution Manjaro mit KDE als Desktopumgebung.



Als Beispiel wird nun *Anwendung hinzufügen* ausgewählt, was der häufigste Anwendungsfall sein dürfte. In dem Fall öffnet sich ein neues Fenster, in dem nun das Programm ausgewählt werden kann. Oben wird ein Suchfeld angezeigt, in das der Name des Programms oder ein Teil davon eingegeben werden kann, um die Liste zu durchsuchen.



Wird das Programm mit OK bestätigt, zeigt das Einstellungsfenster es nun in der Liste der Autostart Programme an. Dort kann es jederzeit wieder bearbeitet oder entfernt werden. Natürlich lassen sich noch weitere Programme oder Skripte zum Autostart hinzufügen.



Dateien live einlesen

Bei Logdateien ist es häufig hilfreich, wenn diese konstant eingelesen und dargestellt werden.

Hierfür bietet sich das Tool *tail* an. Angenommen wir haben eine Logdatei `/var/log/mysql/abfragen.log`. In diese Logdatei werden alle Abfragen eines MySQL-Servers geschrieben. Wenn nun live im Log gelesen werden soll, welche Abfragen gerade ausgeführt werden, so geht dies mit dem folgenden Befehl.

```
tail -f /var/log/mysql/abfragen.log
```

Wird der Befehl ausgeführt, wird die Datei geöffnet und ans Ende gesprungen. Ab dann werden alle neuen Einträge im Terminal angezeigt.

```
2024-03-26T06:31:34.749736Z 41 Query commit
2024-03-26T06:31:34.749954Z 41 Query begin
2024-03-26T06:31:34.750330Z 41 Query commit
2024-03-26T06:31:34.781827Z 25 Query select hostid,key_evaltype,formula,lifetime from items where itemid=102694
2024-03-26T06:31:34.782511Z 25 Query select item_conditionid,macro,value,operator from item_condition where itemid=102694
2024-03-26T06:31:34.783051Z 25 Query select lld_macro,path from lld_macro_path where itemid=102694
2024-03-26T06:31:34.783650Z 25 Query begin
2024-03-26T06:31:34.783988Z 25 Query select lld_overrideid,step,evaltype,formula,stop from lld_override where itemid=102694 order by lld_overrideid
2024-03-26T06:31:34.784581Z 25 Query commit
2024-03-26T06:31:34.785243Z 25 Query select i.itemid,i.name,i.key,i.type,i.value_type,i.delay,i.history,i.trends,i.status,i.trapper_hosts,i.units,i.formula,i.logtimefmt,i.valuemapid,i.params,i.ipmi_sensor,i.snmp_oid,i.authtype,i.username,i.password,i.publickey,i.privatekey,i.description,i.interfaceid,i.jmx_endpoint,i.master_itemid,i.timeout,i.url,i.query_fields,i.posts,i.status_codes,i.follow_redirects,i.post_type,i.http_proxy,i.headers,i.retrieve_mode,i.request_method,i.output_format,i.ssl_cert_file,i.ssl_key_file,i.ssl_key_password,i.verify_peer,i.verify_host,i.allow_traps,i.discover from items i,item_discovery id where i.itemid=id,itemid and id.parent_itemid=102694
2024-03-26T06:31:34.786156Z 25 Query select ip.itemid,ip.step,ip.type,ip.params,ip.error_handler,ip.error_handler_params from item_preproc ip,item_discovery id where ip.itemid=id,itemid and id.parent_itemid=102694
2024-03-26T06:31:34.786856Z 25 Query select ip.itemid,ip.name,ip.value from item_parameter ip,item_discovery id where ip.itemid=id,itemid and id.parent_itemid=102694
2024-03-26T06:31:34.787383Z 25 Query select it.itemid,it.tag,it.value from item_tag it,item_discovery id where it.itemid=id,itemid and id.parent_itemid=102694
2024-03-26T06:31:34.788120Z 25 Query select id.itemid,id.key_id,lastcheck,id.ts_delete,i.name,i.key,i.type,i.value_type,i.delay,i.history,i.trends,i.trapper_hosts,i.units,i.formula,i.logtimefmt,i.valuemapid,i.params,i.ipmi_sensor,i.snmp_oid,i.authtype,i.username,i.password,i.publickey,i.privatekey,i.description,i.interfaceid,i.jmx_endpoint,i.master_itemid,i.timeout,i.url,i.query_fields,i.posts,i.status_codes,i.follow_redirects,i.post_type,i.http_proxy,i.headers,i.retrieve_mode,i.request_method,i.output_format,i.ssl_cert_file,i.ssl_key_file,i.ssl_key_password,i.verify_peer,i.verify_host,i.allow_traps from item_discovery id join items i on id.itemid=i.itemid where id.parent_itemid in (102695,102696,102697,102698)
2024-03-26T06:31:34.788770Z 25 Query select ip.item_preprocid,ip.itemid,ip.step,ip.type,ip.params,ip.error_handler,ip.error_handler_params from item_discovery id join item_preproc ip on id.itemid=ip.itemid where id.parent_itemid in (102695,102696,102697,102698)
2024-03-26T06:31:34.789256Z 25 Query select ip.item_parameterid,ip.itemid,ip.name,ip.value from item_discovery id join item_parameter ip on id.itemid=ip.itemid where id.parent_itemid in (102695,102696,102697,102698)
2024-03-26T06:31:34.789677Z 25 Query select it.itemtagid,it.itemid,it.tag,it.value from item_discovery id join item_tag it on id.itemid=it.itemid where id.parent_itemid in (102695,102696,102697,102698)
2024-03-26T06:31:34.790164Z 25 Query begin
2024-03-26T06:31:34.790339Z 25 Query commit
2024-03-26T06:31:34.790488Z 25 Query begin
2024-03-26T06:31:34.790661Z 25 Query update item_discovery set lastcheck=1711434694 where itemid in (103947,103948,103949,103950)
2024-03-26T06:31:34.791299Z 25 Query commit
2024-03-26T06:31:34.794701Z 25 Query select t.triggerid,t.description,t.expression,t.status,t.type,t.priority,t.comments,t.url,t.recovery_expression,t.recovery_mode,t.correlation_mode,t.correlation_tag,t.manual_close,t.opdata,t.discover,t.event_name from triggers t where t.triggerid in (select distinct tg.triggerid from triggers tg,functions f,items i,item_discovery id where tg.triggerid=f.triggerid and f.itemid=i.itemid and i.itemid=id.itemid and id.parent_itemid=102694)
2024-03-26T06:31:34.795370Z 25 Query select distinct g.graphid,g.name,g.width,g.height,g.yaxismin,g.yaxismax,g.show_work_period,g.show_triggers,g.graphtype,g.show_legend,g.show_3d,g.percent_left,g.percent_right,g.ymin_type,g.ymin_itemid,g.ymax_type,g.ymax_itemid,g.discover from graphs g,graphs_items gi,items i,item_discovery id where g.graphid=gi.graphid and gi.itemid=i.itemid and i.itemid=id.itemid and id.parent_itemid=102694
2024-03-26T06:31:34.795916Z 25 Query select h.proxy_hostid,h.ipmi_authtype,h.ipmi_privilege,h.ipmi_username,h.ipmi_password,h.tls_connect,h.tls_accept,h.tls_issuer,h.tls_subject,h.tls_psk_identity,h.tls_psk from hosts h,items i where h.hostid=h.hostid and i.itemid=102694
2024-03-26T06:31:34.796335Z 25 Query select hi.interfaceid,hi.type,hi.main,hi.useip,hi.ip,hi.dns,hi.port,s.version,s.bulk,s.community,s.securityname,s.securitylevel,s.authpassphrase,s.privpassphrase,s.authprotocol,s.privprotocol,s.contextname from interface hi inner join items i on hi.hostid=i.hostid left join interface snmp s on hi.interfaceid=s.interfaceid where i.itemid=102694
2024-03-26T06:31:34.796800Z 25 Query select hm.macro,hm.value,hm.description,hm.type from hostmacro hm,items i where hm.hostid=i.hostid and i.itemid=102694
2024-03-26T06:31:34.797191Z 25 Query select h.hostid,h.host,h.name,h.status,h.discover,hi.inventory_mode,h.custom_interfaces from hosts h,host_discovery hd left join host_inventory hi on hd.hostid=hi.hostid where h.hostid=hd.hostid and hd.parent_itemid=102694
```

Automatische Updates

Um Linux Server automatisch auf dem neusten Stand zu halten und somit den besten Schutz gegen Angriffe bieten zu können, bieten sich die automatischen Updates an.

Automatische Sicherheitsupdates aktivieren

Hierfür steht das Paket `unattended-upgrades` bereit, welches ggf. mit folgenden Befehl nachinstalliert werden muss:

```
sudo apt -y install unattended-upgrades
```

Nach der Installation können die Auto-Updates wie folgt aktiviert werden:

```
sudo dpkg-reconfigure -plow unattended-upgrades
```

Zusätzlich sind noch einige Einstellungen vorzunehmen, damit auch wirklich alle Updates automatisch eingespielt werden.

Zuerst wird der automatische Neustart aktiviert, damit nach der Installation von Updates auch ein Neustart, sofern notwendig, durchgeführt wird. Hierfür ist die Konfigurationsdatei

`/etc/apt/apt.conf.d/50unattended-upgrades` anzupassen. Diese enthält diverse Einstellungen, also am besten nach `Automatic-Reboot` suchen, auskommentieren und auf `true` ändern, wie im folgenden Beispiel zu sehen.

```
// Automatically reboot *WITHOUT CONFIRMATION* if
// the file /var/run/reboot-required is found after the upgrade
Unattended-Upgrade::Automatic-Reboot "true";
```

Sicherheits- und Standard-Pakete aktualisieren

Die Datei noch nicht schließen, denn standardmäßig werden nur Sicherheitsupdates installiert. Also in der selben Konfigurationsdatei nach folgender Zeile suchen und die Kommentarzeichen entfernen.

```
"${distro_id}:${distro_codename}-updates";
```

Zeitpunkt der Suche/Installation anpassen

Nun können wir noch den Zeitpunkt anpassen, wann nach Updates gesucht und wann diese installiert werden sollen. Dazu dient der folgende Befehl, mit dessen Hilfe die Konfigurationsdatei bearbeitet wird.

```
sudo systemctl edit apt-daily.timer
```

Daraufhin öffnet sich die Konfiguration für die Update-Suche. Um die Updates z. B. jeden Tag um 01:00 Uhr automatisch suchen zu lassen, wird die Update-Suche wie folgt konfiguriert.

```
### Editing /etc/systemd/system/apt-daily.timer.d/override.conf
### Anything between here and the comment below will become the new contents of the file

[Timer]
OnCalendar=
OnCalendar=01:00
RandomizedDelaySec=0

### Lines below this comment will be discarded

### /lib/systemd/system/apt-daily.timer
# [Unit]
# Description=Daily apt download activities
#
# [Timer]
# OnCalendar=*-*-* 6,18:00
# RandomizedDelaySec=12h
# Persistent=true
#
# [Install]
# WantedBy=timers.target
```

Die Konfiguration für die Installation der Updates könnte z. B. wie folgt gestaltet werden, um diese eine halbe Stunde nach der Suche zu installieren.

```
sudo systemctl edit apt-daily-upgrade.timer
```

Und folgendes in die Konfigurationsdatei.

```
### Editing /etc/systemd/system/apt-daily-upgrade.timer.d/override.conf
### Anything between here and the comment below will become the new contents of the file
```

```
[Timer]
OnCalendar=
OnCalendar=01:30
RandomizedDelaySec=0

### Lines below this comment will be discarded

### /lib/systemd/system/apt-daily-upgrade.timer
# [Unit]
# Description=Daily apt upgrade and clean activities
# After=apt-daily.timer
#
# [Timer]
# OnCalendar=*-*-* 6:00
# RandomizedDelaySec=60m
# Persistent=true
#
# [Install]
# WantedBy=timers.target
```

Nun noch die Dienste neu starten und den Status prüfen.

```
sudo systemctl restart apt-daily.timer
sudo systemctl restart apt-daily-upgrade.timer
sudo systemctl status apt-daily.timer
sudo systemctl status apt-daily-upgrade.timer
```

Unter `/var/log/unattended-upgrades/` werden die verschiedenen Logs für Update-Suche, Installation und Neustart abgelegt. Das Log `/var/log/unattended-upgrades/unattended-upgrades.log` enthält z. B. die wichtigsten Informationen zu installierten Updates und Neustart.

Zusätzliche Pakete aktualisieren

Wenn auf einem System zusätzliche Paketquellen wie z. B. für Docker hinzugefügt wurden, dann müssen diese manuell mit aufgenommen werden.

Mit folgendem Befehl können alle Paketquellen des Systems eingelesen werden.

```
ls -l /var/lib/apt/lists/ | grep "Release"
# Optional kann auch der folgende Befehl genutzt werden, um verschiedene Standard-Paketquellen rauszufiltern
ls -l /var/lib/apt/lists/ | grep "Release" | grep -v "security.ubuntu.com\|asi-fs-n."
```

Die Ausgabe könnte z. B. wie folgt aussehen.

```
-rw-r--r-- 1 root root 108609 Mar 18 19:43 asi-fs-n.contabo.net_ubuntu_dists_jammy-backports_InRelease
-rw-r--r-- 1 root root 270087 Apr 21 2022 asi-fs-n.contabo.net_ubuntu_dists_jammy_InRelease
-rw-r--r-- 1 root root 118761 Mar 27 21:20 asi-fs-n.contabo.net_ubuntu_dists_jammy-updates_InRelease
-rw-r--r-- 1 root root 48847 Mar 22 14:50 download.docker.com_linux_ubuntu_dists_jammy_InRelease
```

Hier sehen wir nun oben die Standard-Pakete von Contabo unter Ubuntu und ganz unten eine Paketquelle von Docker, die zur Zeit nicht aktualisiert wird. Also ermitteln wir zuerst einige Informationen über diese Paketquelle mit folgendem Befehl.

```
cat /var/lib/apt/lists/download.docker.com_linux_ubuntu_dists_jammy_InRelease | grep
"Origin\|Suite"
```

In diesem Fall wird nun folgendes ausgegeben.

```
Origin: Docker
Suite: jammy
```

Jetzt können wir das Docker Paket hinzufügen, indem wir wieder die Datei `/etc/apt/apt.conf.d/50unattended-upgrades` bearbeiten. In dieser fügen wir die Zeilen in folgendem Abschnitt zwischen den geschweiften Klammern hinzu. Das Format für die Eingabe ist:

```
"<Origin>:<Suite>"; und somit in diesem Fall: "Docker:jammy";.
```

```
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but if
    // available, the policy for updates is such that unattended-upgrades
    // should also install from here by default.
    "${distro_id}ESMA:${distro_codename}-apps-security";
    "${distro_id}ESM:${distro_codename}-infra-security";
    "${distro_id}:${distro_codename}-updates";
    // "${distro_id}:${distro_codename}-proposed";
    // "${distro_id}:${distro_codename}-backports";
};
```

Danach sieht der Abschnitt wie folgt aus und wir speichern die Änderung.

```
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but if
    // available, the policy for updates is such that unattended-upgrades
    // should also install from here by default.
    "${distro_id}ESMApps:${distro_codename}-apps-security";
    "${distro_id}ESM:${distro_codename}-infra-security";
    "${distro_id}:${distro_codename}-updates";
//    "${distro_id}:${distro_codename}-proposed";
//    "${distro_id}:${distro_codename}-backports";
    "Docker:jammy";
};
```

Nun das ganze einmal mit folgendem Befehl testen.

```
sudo unattended-upgrade --dry-run --debug
```

Jetzt sollten auch die Docker Updates angewendet werden. Aufgrund des Parameters `--dry-run` wird der Vorgang nur simuliert und nimmt keine Änderungen am System vor.