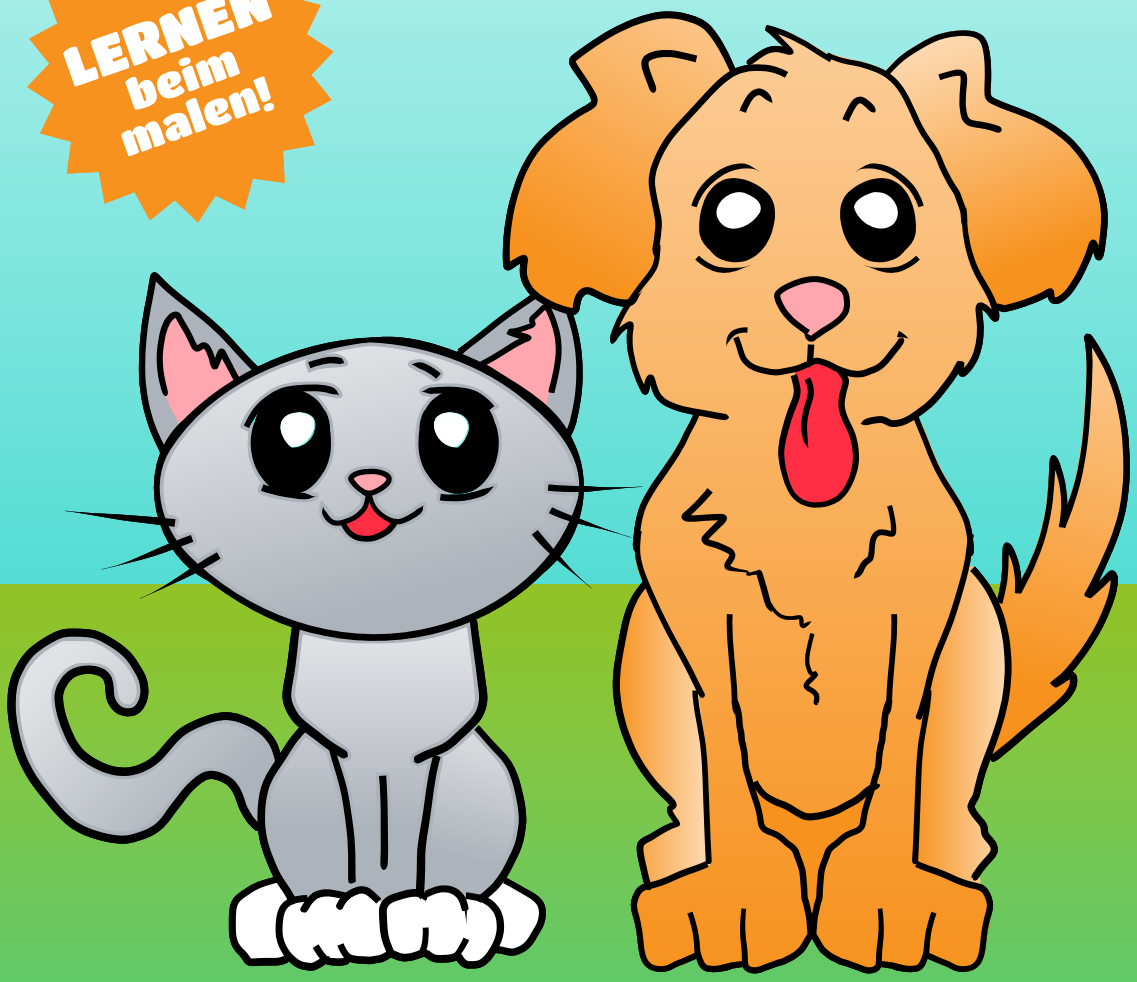
das

SELinux AUSMALBUCH

"Es regnet Katzen und Hunde!"

LERNEN beim malen!



Text von DAN WALSH

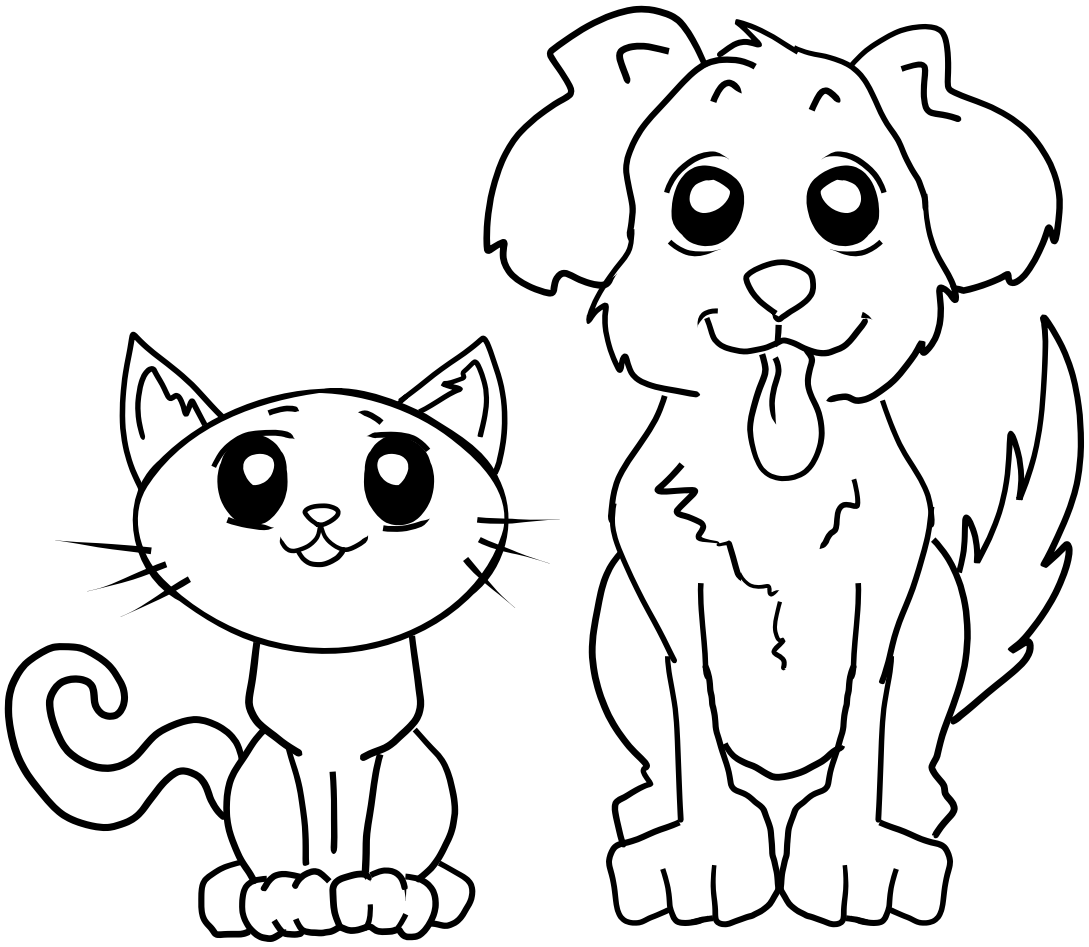
Illustrationen von MÁIRÍN DUFFY

Type Enforcement

PROCESS TYPES

Das grundlegende Prinzip der Zugriffskontrolle in SELinux heißt "Type Enforcement" (Typ Erzwingung). Das heißt eigentlich nur, dass wir das Label eines Prozesses und eines Dateisystem-Objekts auf Grundlage seines Typen definieren.

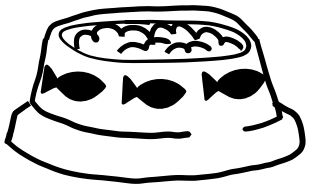
Stell dir ein System vor, in dem wir Typen für Objekte wie Katzen oder Hunde festlegen. Katzen und Hunde sind Prozesstypen (Process Types).



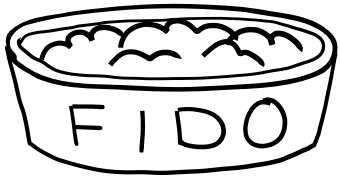
KATZE HUND

OBJECT TYPES

Wir haben eine Klasse von Objekten, genannt "Futter", mit der Hund und Katze interagieren wollen. Zudem wollen wir die Typen hund_futter und katze_futter zur Klasse "Nahrung" hinzufügen.



KATZE_FUTTER

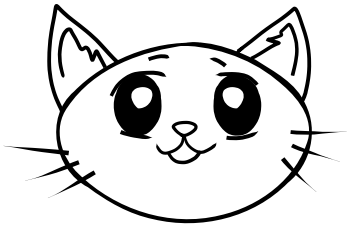


HUND_FUTTER

POLICY RULES

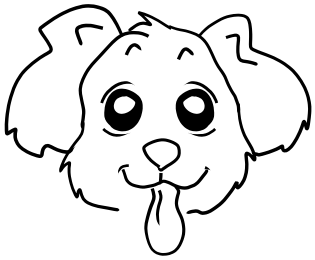
Ein Autor einer Policy Rule (Regel im Regelwerk) würde festlegen, dass ein Hund die Erlaubnis hat, hund_futter zu essen. Schreibe diese Erlaubnis als Policy Rule wie unten dargestellt.

ERLAUBEN



KATZE

ERLAUBEN

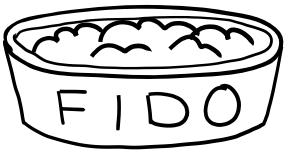


HUND

Eine Katze hat die Erlaubnis Nahrung vom Typ katze_futter zu fressen.
In SELinux würden wir folgende Regeln definieren:

KATZE_FUTTER:NAHRUNG

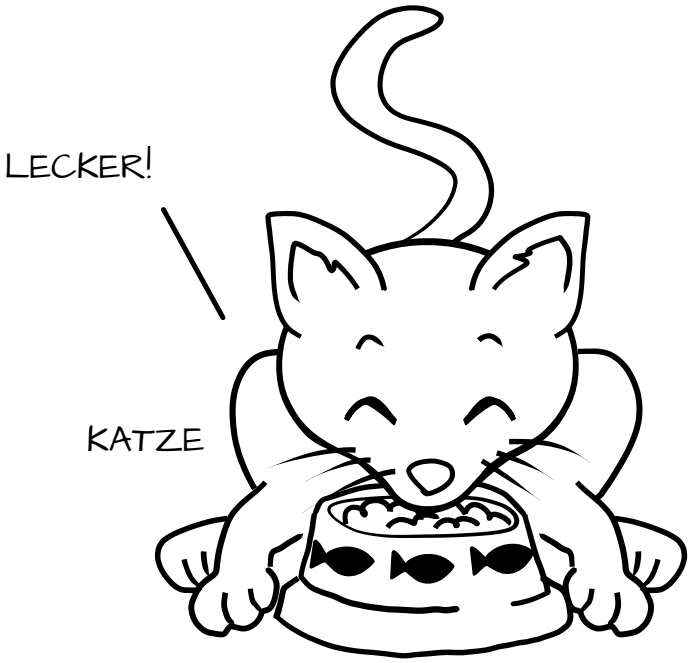
ESSEN



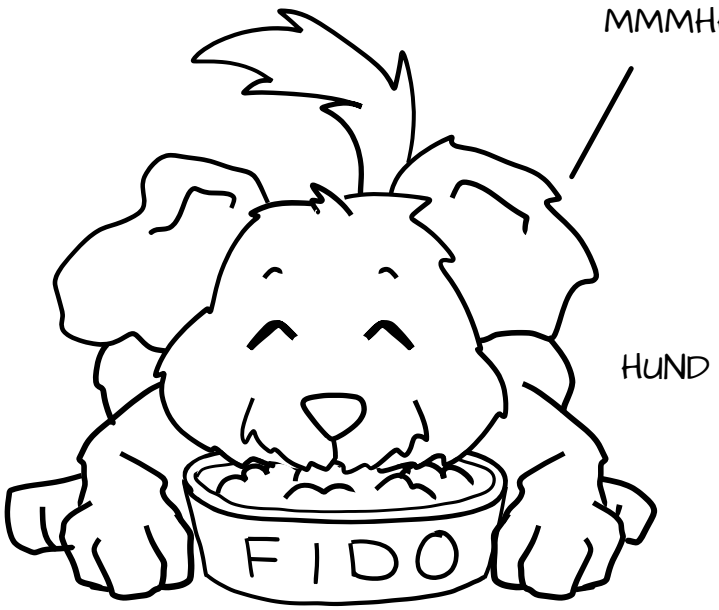
HUND_FUTTER:NAHRUNG

ESSEN

Mit diesen Regeln würde der Kernel dem Prozess "Katze" erlauben, Nahrung mit dem Label katze_futter zu fressen und dem Prozess "Hund", Nahrung mit dem Label hund_futter.

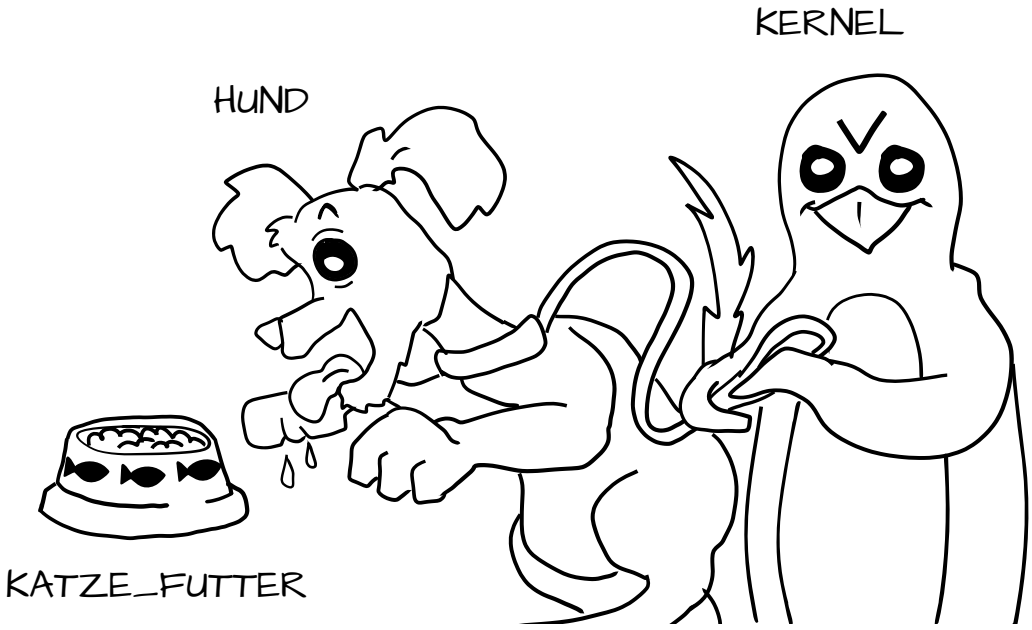


KATZE_FUTTER:NAHRUNG

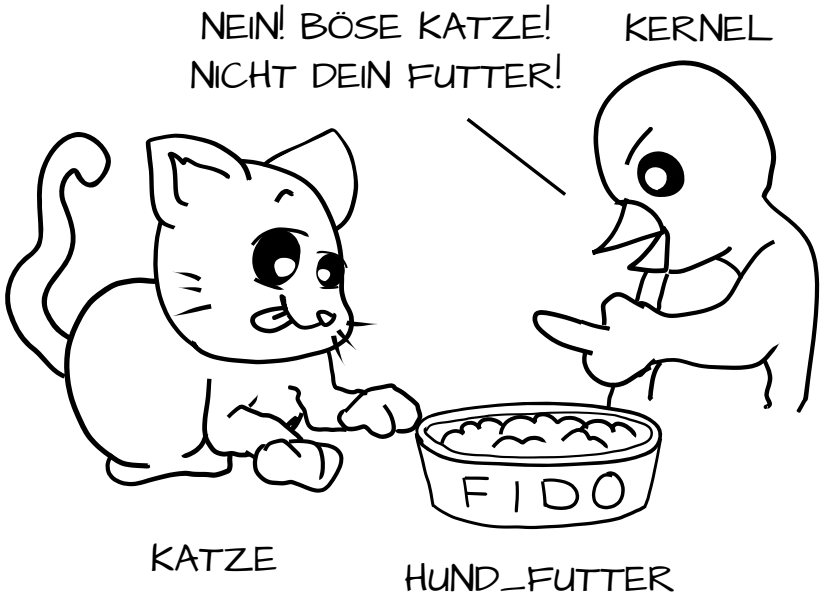


HUND_FUTTER:NAHRUNG

Aber in einem SELinux-System ist grundsätzlich ALLES verboten!
Das heißt, wenn der "Hund" Prozess versucht katze_futter zu fressen, würde ihn der Kernel davon abhalten.

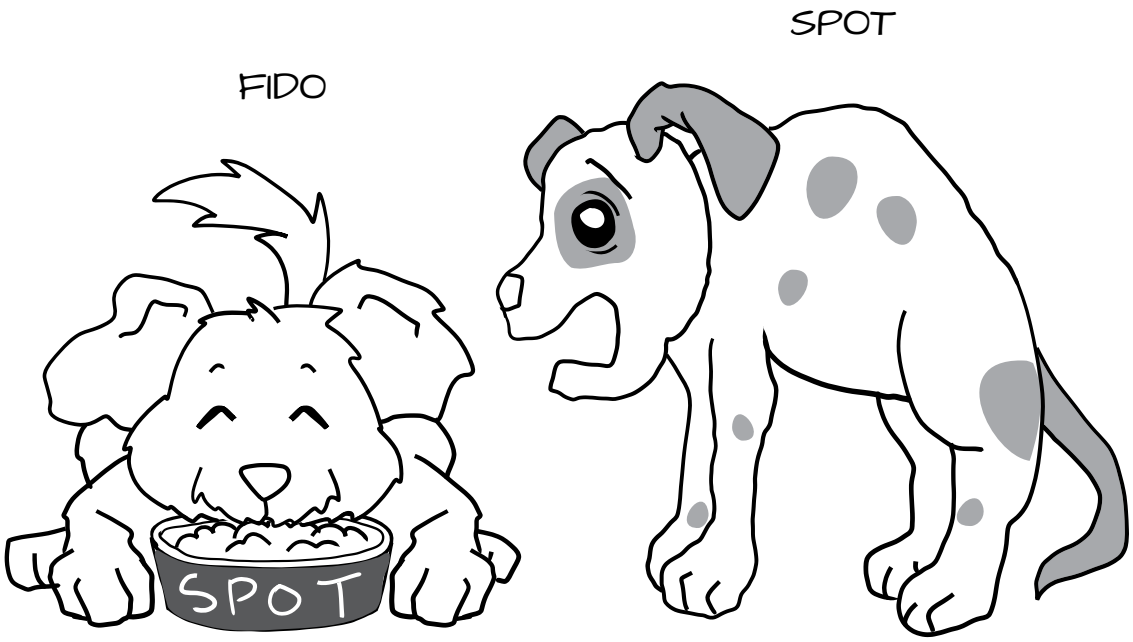


Genauso wäre es Katzen nicht erlaubt sich am Hundefutter zu bedienen.



MCS Enforcement

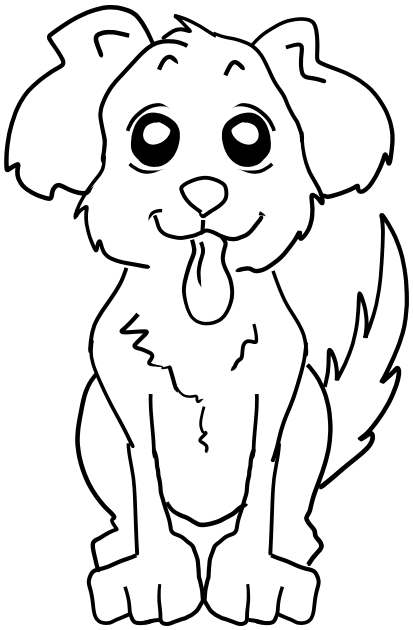
Wir haben nun die Prozesse "Hund" und "Katze" typisiert, aber was passiert wenn wir mehrere "Hund" Prozesse (Fido und Spot) haben? Fido soll davon abgehalten werden Spots Hundefutter zu fressen.



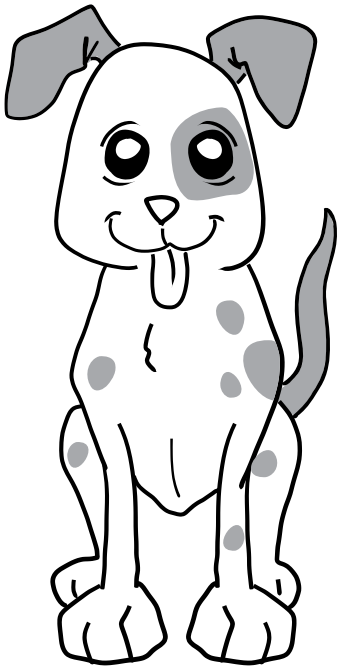
Ein mögliche Lösung wäre, viele neue Typen wie fido_hund und fido_hund_futter anzulegen. Das wäre auf Dauer allerdings unpraktisch, da die Berechtigungen für alle Hunde eigentlich gleich sind.

Um dieses Problem zu lösen, haben wir eine neue Form der Erzwingung (Enforcement) entwickelt, die wir Multi Category Security (MCS) nennen. MCS sieht vor, dass wir dem Label einen weitereren Abschnitt hinzufügen, den wir auf den "Hund" Prozess und auf die Nahrung hund_futter anwenden können. Nun vergeben wir die Label hund:zufall1 (Fido) und hund:zufall2 (Spot) für den "Hund" Prozess .

Das Hundefutter bekommt die Label hund_futter:zufall1 (Fido) und hund_futter:zufall2 (Spot).



HUND:ZUFALL1



HUND:ZUFALL2

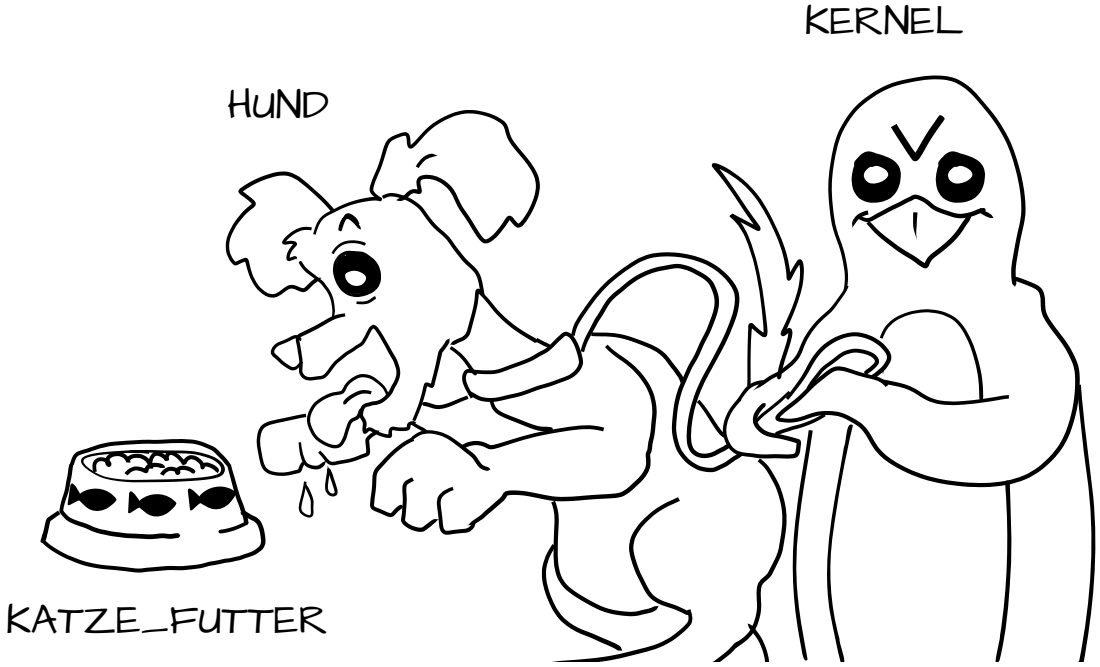
HUND_FUTTER:
ZUFALL1

HUND_FUTTER:
ZUFALL2

Das MCS Regelwerk besagt, dass wenn die Type Enforcement Regeln OK sind und die zufälligen MCS Label übereinstimmen, der Zugriff erlaubt (allowed) wird, ansonsten wird er verboten (denied).

TYPE ENFORCEMENT

Wenn Fido (hund:zufall1) versucht katze_futter:nahrung zu essen, wird dies per Type Enforcement verboten.



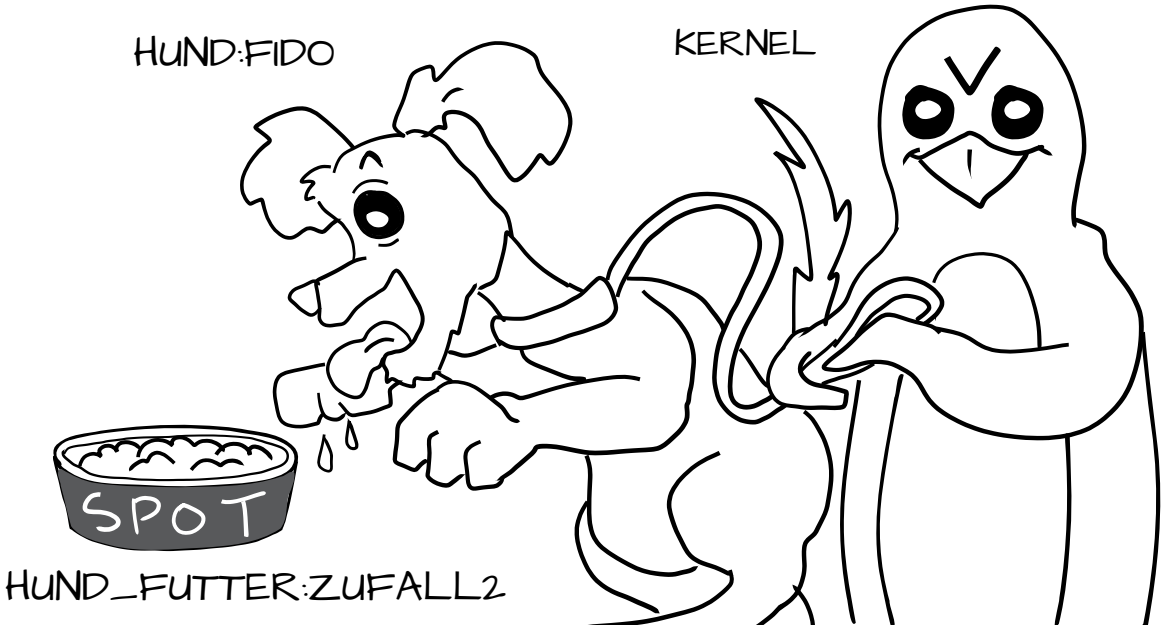
Es ist erlaubt, dass Fido (hund:zufall1) das Hundefutter hund_futter:zufall1 frisst.

HUND:ZUFALL1

HUND_FUTTER:ZUFALL1

MCS ENFORCEMENT

Fido (hund:zufall1) wird verboten, Spots Nahrung (hund_futter:zufall2) zu fressen.

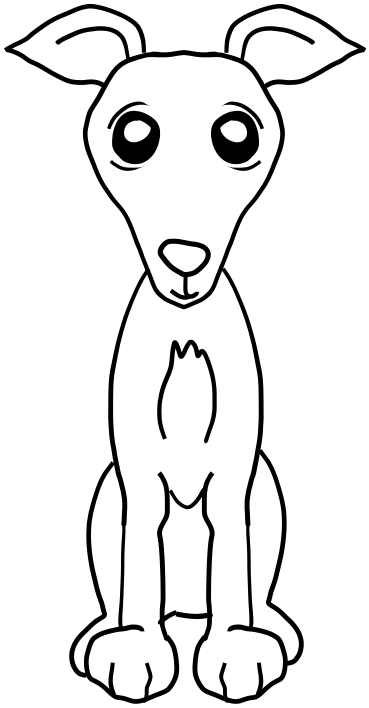


MLS Enforcement

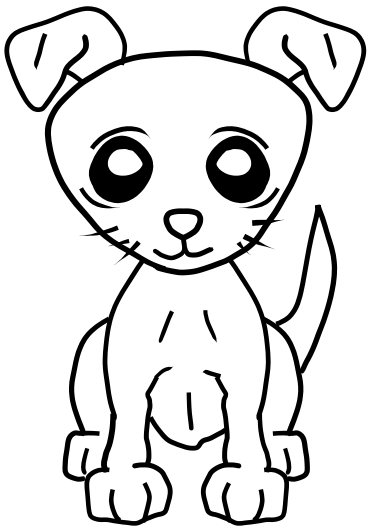
Eine weitere Form von SELinux Erzwingung (Enforcement), die weniger häufig benutzt wird, heißt Multi Level Security (MLS). MLS wurde in den 60er Jahren entwickelt und wird hauptsächlich in TOS (trusted operating systems) Betriebssystemen wie Trusted Solaris verwendet.

Der Grundgedanke dabei ist, Prozesse basierend auf Berechtigungsstufen der Daten die sie verwenden werden, zu kontrollieren. Ein "geheimer" Prozess kann keine "STRENG geheimen" Daten lesen.

Anstelle von verschiedenen Hunden zu sprechen, betrachten wir nun die verschiedenen Hunderassen. Wir gehen davon aus, dass wir einen Windhund und einen Chihuahua haben.



WINDHUND

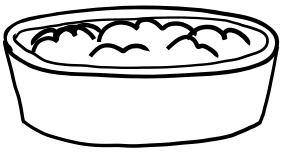


CHIHUAHUA

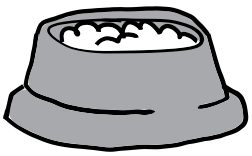
Wir möchten dem Windhund erlauben, jedes Hundefutter zu fressen. Der Chihuahua hingegen könnte sich am Windhund-Futter verschlucken.

Wir wollen für den Windhund die Label hund:windhund und für sein Futter hund_futter:windhund; für den Chihuahua hund:chihuahua und hund_futter:chihuahua vergeben.

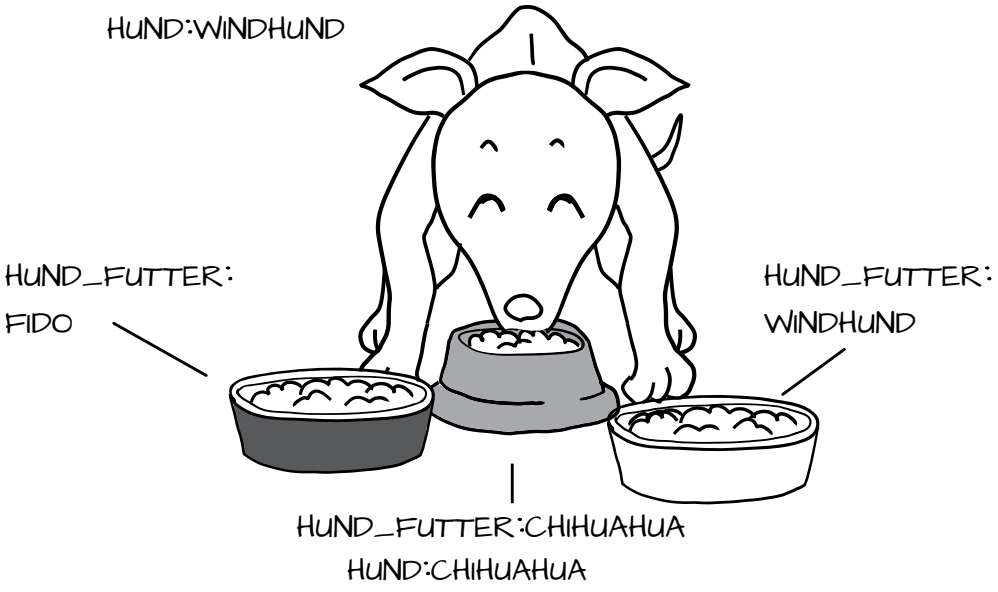
HUND_FUTTER:WINDHUND



HUND_FUTTER:CHIHUAHUA



Mit dem MLS Regelwerk, würde das MLS Windhund Label das Chihuahua Label stechen (dominate). Das heißt hund:windhund darf (is allowed) hund_futter:windhund und hund_futter:chihuahua fressen.

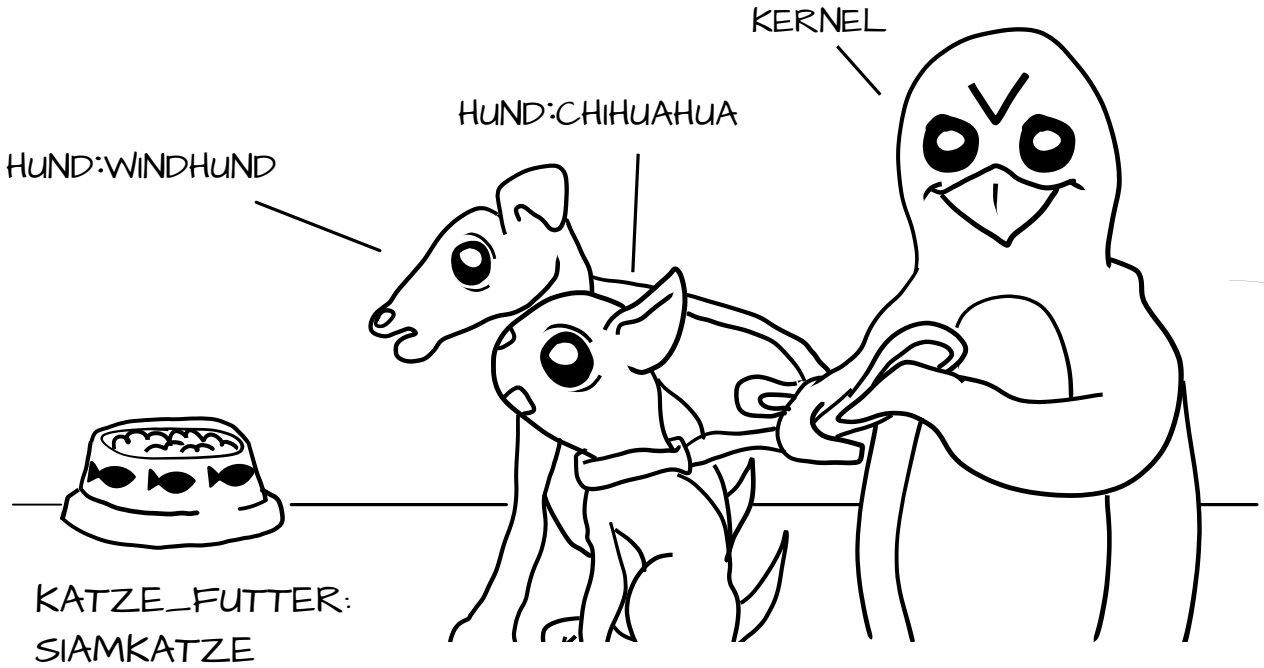


HUND:CHIHUAHUA

HUND_FUTTER:CHIHUAHUA

Aber hund:chihuahua darf kein hund_futter:windhund fressen.

Natürlich werden hund:windhund und hund:chihuahua weiterhin per Type Enforcement verboten katze_futter:siamkatze zu essen, selbst wenn der MLS-Typ Windhund den Typ Siamkatze sticht (dominates).



Erfahre mehr auf opensource.com:

http://ur1.ca/g12br