Network Bound Disk Encryption

In diesem Artikel gebe ich euch einen Überblick, was Network Bound Disk Encryption (NBDE) ist und beschreibe einen konkreten Anwendungsfall. Am Ende des Artikels führe ich einige Verweise auf, mit deren Hilfe ihr NBDE bei Interesse selbst implementieren könnt.

Linux Unified Key Setup (LUKS)

Bevor ich auf NBDE eingehe, möchte ich kurz ein paar Worte zu LUKS verlieren.

Bei LUKS handelt es sich um das Standardverfahren zur Festplattenverschlüsselung unter Linux [1)]. Dieses erweitert dm-crypt um einen Header, welcher Slots zum Speichern von bis zu acht Schlüsseln bietet [2)].

Ich benutze dieses Verfahren auf nahezu all meinen Rechnern. Ich möchte damit erreichen, dass meine Daten bei Diebstahl des Rechners bzw. der Festplatte möglichst lange vor unberechtigtem Zugriff geschützt sind.

Typischerweise wird zur Entschlüsselung einer Festplatte bzw. Partition während des Boot-Vorgangs die Eingabe eines Kennworts benötigt. Die Sicherheit des Verfahrens hängt dabei direkt von der Stärke des verwendeten Passworts ab.

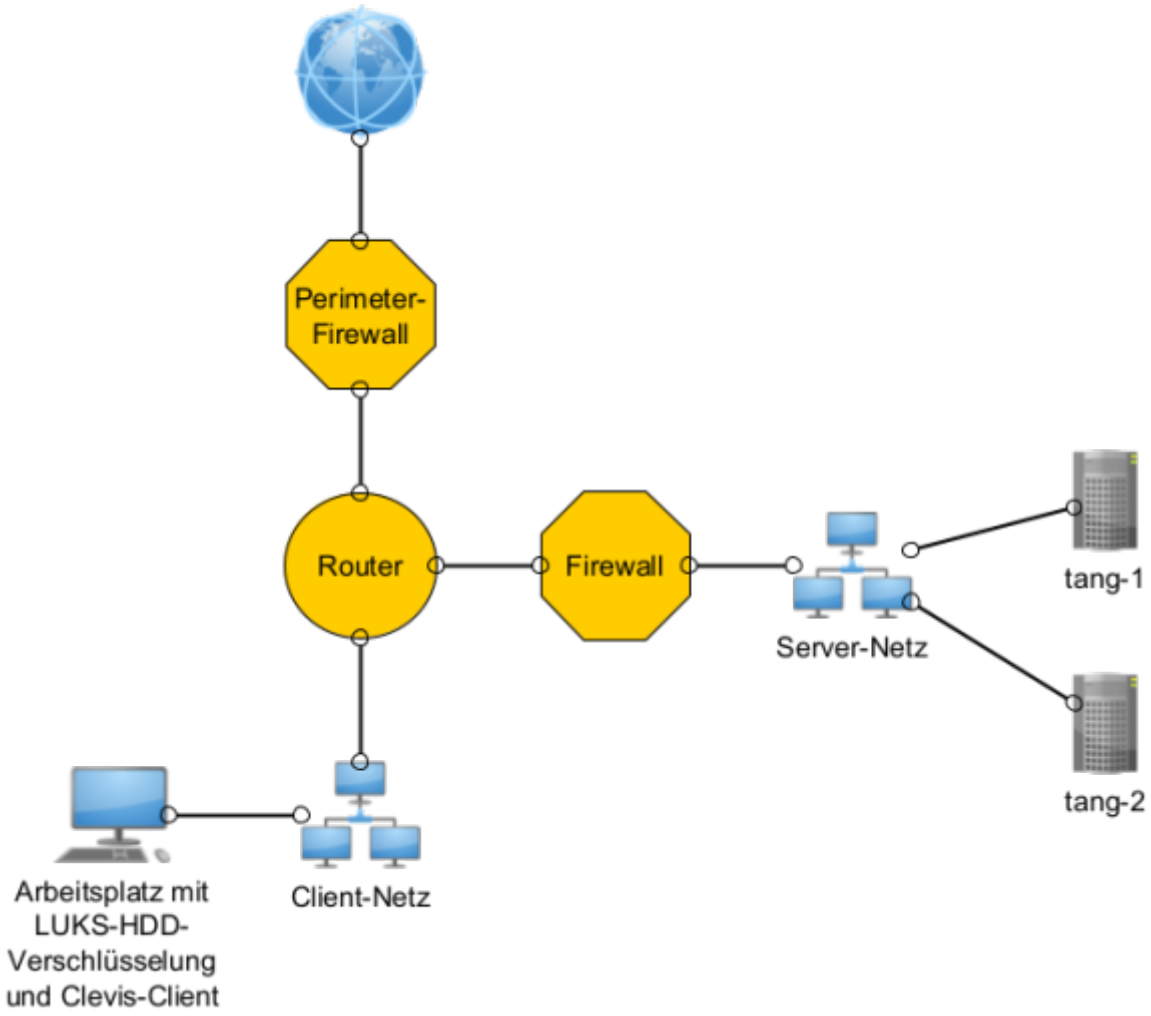
Steht der Rechner im Büro und man ist täglich vor Ort, ist es kein Problem, bei Neustart des Rechners das LUKS-Kennwort einzugeben. Wenn man allerdings im Homeoffice arbeitet und Zugriff auf seinen Büro-Rechner braucht, sieht die Sache anders aus.

Möchte man den entfernten Rechner neustarten, z.B. nach der Installation von Updates, muss man dafür extra ins Büro fahren. Alternativ kann man ein zweites Kennwort einrichten, dieses einem Kollegen mitteilen und diesen bitten, es vor Ort einzugeben. Beides ist nicht komfortabel. Und hier kommt NBDE ins Spiel.

LUKS an Netzwerkressource binden

Network Bound Disk Encryption heißt auf Deutsch in etwa netzwerkgebundene Festplattenverschlüsselung und bedeutet, dass die Verschlüsselung an eine oder mehrere Ressourcen im Netzwerk gebunden ist.

Das Prinzip ist ganz einfach. Wenn ein Rechner mit einer verschlüsselten Festplatte oder Partition startet, sucht er nach einer bestimmten Ressource im Netzwerk. Kann der Rechner diese Netzwerkressource erreichen, wird die Festplatte bzw. Partition entschlüsselt und der Startvorgang fortgesetzt. Folgende Abbildung soll dies veranschaulichen.



Im eigenen LAN werden zwei sogenannte Tang-Server positioniert. Diese stellen die Netzwerk-Ressource dar, welche erreichbar sein muss, damit ein an Tang gebundenes Gerät entschlüsselt werden kann. In diesem Beispiel werden zwei Tang-Server betrieben, um die Verfügbarkeit des Dienstes zu gewährleisten, wenn ein Server gewartet wird.

Auf dem Client kommt die Anwendung Clevis zum Einsatz, bei welcher es sich um die Client-Komponente zum Tang-Server handelt. Diese empfängt einen Schlüssel vom Tang-Server und verwendet diesen, um einen Token in einem LUKS-Slot zu speichern. Beim Start eines Rechners wird nun versucht, einen der Tang-Server zu erreichen, an die man sich gebunden hat.

Wird der Rechner bzw. seine Festplatte gestohlen, sind die Tang-Server nicht erreichbar und die Daten werden nicht automatisch entschlüsselt. Der Dieb muss nun die Verschlüsselung bzw. das verwendete Kennwort brechen.

Steht der Rechner jedoch an seinem Platz, kann er aus der Ferne neugestartet werden und den Startvorgang beenden, ohne dass jemand vor Ort ein LUKS-Kennwort eingeben muss.

Damit diese Konfiguration Sinn macht, dürfen die Tang-Server nicht weltweit erreichbar sein. Ihr Standort und die Netze, aus denen sie erreichbar sind, sind daher sorgfältig zu planen.

Zusammenfassung

Ohne NBDE muss an einem Rechner mit LUKS-Verschlüsselung bei jedem Startvorgang das LUKS-Kennwort eingegeben werden, was einen Neustart aus der Ferne enorm erschwert.

NBDE ist leicht zu implementieren und löst dieses Problem. Gleichzeitig bleiben die Daten im gleichen Maße bei einem Diebstahl des Rechners geschützt.

1)

LUKS im Wiki von Ubuntuusers.de

2)

https://de.wikipedia.org/wiki/Dm-crypt#LUKS