Qubes Windows Seamless Mode zurückbringen!

Keno Goertz

11. Oktober 2024

Sicherheit moderner Desktop-Betriebssysteme

Salopp ausgedrückt:

- Was möchte ich schützen? Meine Daten!
- Was schützen die meisten Betriebssysteme? Den Root-Account...

Sandboxing

- Traditioneller Ansatz: Ein ausgeführter Prozess hat Zugriff auf alle Dateien der ihn ausführenden Nutzer*in
- Sandboxing: Eine ausgeführte Anwendung hat nur Zugriff auf die Daten, die sie auch tatsächlich braucht

Sandboxing in der Praxis

- Windows :-(
  - Mit der Universal Windows Platform (UWP) waren sie auf nem guten Weg, aber Microsoft hat das Projekt ad acta gelegt
- Linux :-(
  - Flatpak-Anwendungen laufen in einer Sandbox, allerdings legen die Anwendungen selbst fest, auf welche Daten sie zugreifen können wollen...
- MacOS :-)
  - Apple hat richtig Gehirnschmalz in das Sandboxing von Anwendungen gesteckt

Vertrauen

- Trusted Computing Base (TCB): Alle Hardware-, Firmware- und Softwarekomponenten die kritisch für die Sicherheit des Systems sind
- „Trusted“ heißt, dass wir diesen Komponenten vertrauen *müssen*. Nicht, dass sie auch tatsächlich vertrauenswürdig sind!

Wie groß ist die TCB?

- Typische Betrachtungsweise: Kernel und einige System Utilities (z. B. `setuid`)
- Aber nochmal: Was möchte ich schützen? Meine Daten!
- TCB in Linux und Windows Desktops: Sämtliche Software inklusive aller von der Nutzer*in installierter Anwendungsprogramme?
- TCB in MacOS schon eher Kernel und einige System Utilities, aber ist das auch *vertrauenswürdig*?

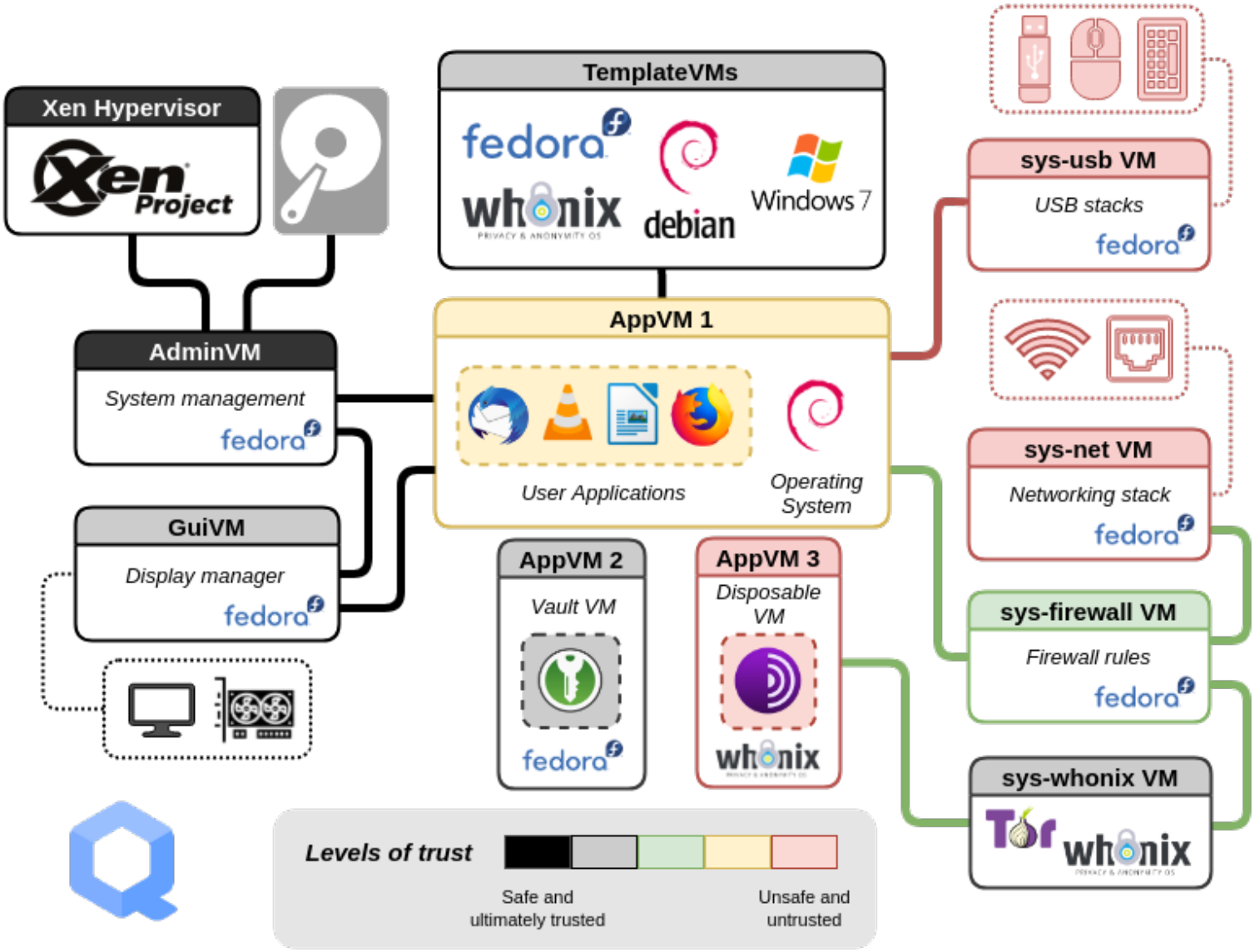
TCB minimieren

- Mehr Code führt zu mehr Sicherheitslücken. Daher wollen wir die TCB möglichst klein halten
- Wo wir nicht drum herumkommen: Der Hardware vertrauen!
- minimale TCB: Airgap (verschiedene physische Geräte für die unterschiedlichen Sicherheitskontexte)
  - Problem: Üblicherweise ist ein Gerät, auf das und von dem man überhaupt keine Daten übertragen kann, ziemlich nutzlos
  - Datenübertragung über USB? Über lokales Netzwerk? Dann sind wieder das Kernel und der ganze USB- und Networking-Stack Teil der TCB... Haben wir dann wirklich was erreicht?

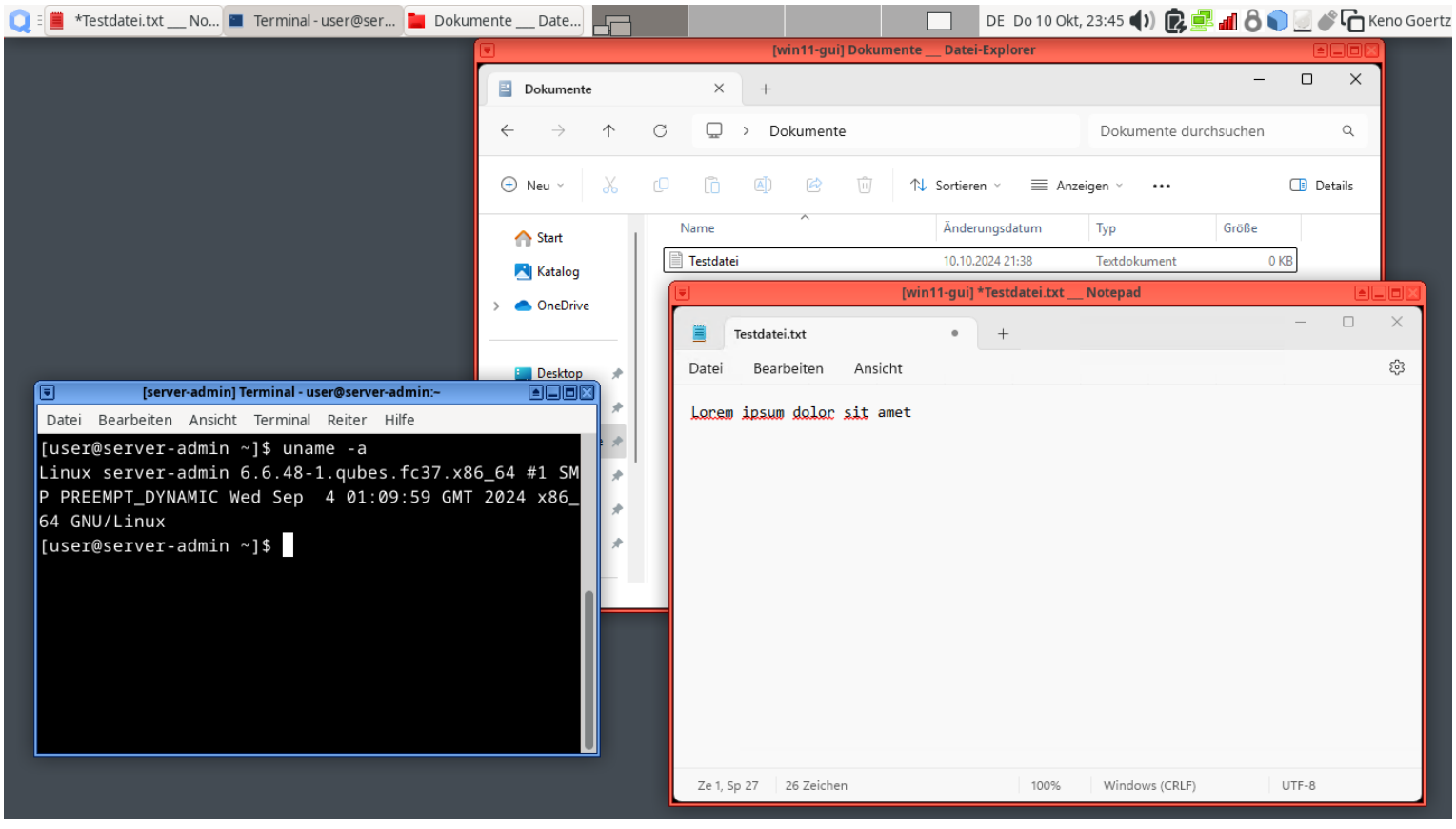
TCP minimieren: Zweiter Versuch

- Typ-1-Hypervisor: Verwaltet virtuelle Maschinen und läuft dafür selbst direkt auf der Hardware, ohne Hostbetriebssystem
- Vergleich Codezeilen zweier Open Source Projekte:
  - Linux Kernel: 35 Mio.
  - Xen Hypervisor: 70.000
- Qubes OS baut auf dem Xen Hypervisor auf. Weitere Teile der TCB:
  - vom Qubes-Team geschriebener Code zur Inter-VM-Kommunikation
  - RPM für Systemupdates
  - insgesamt Größenordnung $\approx 10^5$ Codezeilen. Typisch für andere Betriebssysteme ist eher $> 10^7$

Qubes OS Architektur



Demo!


*Testdatei.txt __ No...
Terminal - user@ser...
Dokumente __ Date...
DE Do 10 Okt, 23:45
Keno Goertz
[win11-gui] Dokumente __ Datei-Explorer
Dokumente
Dokumente durchsuchen
Neu
Sortieren
Anzeigen
Details
Start
Katalog
OneDrive
Desktop
Name
Änderungsdatum
Typ
Größe
Testdatei
10.10.2024 21:38
Textdokument
0 KB
[win11-gui] *Testdatei.txt __ Notepad
Testdatei.txt
Datei
Bearbeiten
Ansicht
Lorem ipsum dolor sit amet
Ze 1, Sp 27
26 Zeichen
100%
Windows (CRLF)
UTF-8
[server-admin] Terminal - user@server-admin:~
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
[user@server-admin ~]$ uname -a
Linux server-admin 6.6.48-1.qubes.fc37.x86_64 #1 SM
P PREEMPT_DYNAMIC Wed Sep 4 01:09:59 GMT 2024 x86_
64 GNU/Linux
[user@server-admin ~]$