

Heimadæmi 3

Arnar Sigurðsson

1. Vegna þess að Virtualbox og önnur samskonar forrit virka sem nokkurs konar sandkassi fyrir forritin/stýrikerfin sem eru inni í þeim. Skráin sem við keyrum í virtualbox „heldur“ að hún sé að sett upp sem venjulegt stýrikerfi á tölvu með venjulegan aðgang að öllu (disk, íhlutum, vélbúnaði og slíku) en í raun er Virtualbox að herma þessa hluti og þannig væri yfirleitt ekki mjög hættulegt að einhver slæmur kóði sé keyrður inni í kerfinu inni í Virtualbox, því hægt væri bara að slökkva á stýrikerfinu í Virtualboxinu og endurræsa eða hreinlega eyða og byrja upp á nýtt. Þessi slæmi kóði kemst ekki út úr þessu boxi nema hann viti að hann sé í sýndarvél og geti fundið samskiptaleiðina á milli stýrikerfisins og Virtualbox eða eitthvað slíkt en til þess þarf frekar fagaðan vírus.
2. Virtualbox processinn rýkur úr undir 1% örgjörvanotkun og fer í um það bil 15% á meðan að á keyrslu stendur.
3. Í host-vélinni fer ferlið Windows command processor upp í um 15% örgjörvanotkun á meðan að á keyrslu stendur en inni í guest-vélinni kemur ekki fram nein breyting.
4. Þetta er vegna þess að eins og kom fram í lið 1 er guest-vélin að keyra inni í hermun hjá Virtualbox þannig hún hefur ekki aðgang að upplýsingum sem eru fyrir utan Virtualbox nema Virtualbox deili þeim sérstaklega með vélinni. Eina samskiptarás frá guest-vélinni yfir í hvað er að gerast í host-vélinni er í gegnum Virtualbox.

Heimadæmi 4

Arnar Sigurðsson

Örgjörvin fer í gang og keyrir skipanir úr minnissvæði sem eru harðkóðaðar í örgjörvann. Þessar vélakóðaskipanir mynda það sem er kallað firmware en firmware er forrit sem er leyfir stjórn á „low-level“ hluta tölvunnar, eða vélbúnaðarins. Firmware inniheldur bootstrap-forrit sem getur bent á stað þar sem kjarninn er staðsettur, t.d. á hörðum disk eða SSD sem hefur verið útfærður þannig að hægt sé að „boota“ af þeim, eða hafa „boot-partition“.

Kjarnanum er svo hlaðið inn og honum afhent stjórnin. Þá er stýrikerfið í raun með stjórnina.

Ef við hugsum okkur að verið sé að nota kernel með hönnun byggða á einingum(modules) þá þarf einhvern veginn að ná í til dæmis file-system eininguna en kjarninn sjálfur getur ekki séð um það þegar ekki hefur verið náð í neina einingu. Linux leysir þetta vandamál með því að láta lítið tímabundið root-file-system bundið kjarnanum fylgja þegar kjarninn er ræstur. Þá er hægt að nota þetta tímabundna file-system til að hlaða hinu eiginlega kerfi. Þetta er kallað initial-ramdisk og eftir að hafa notað þessa aðferð til að hlaða inn nauðsynlegum einingum eins og file-systeminu og einnig driverum er hægt að hlaða inn restinni af einingunum. Þegar búið er að hlaða inn mikilvægu einingunum og driverum er tímabunduna file-systemið losað og minnið sem það notaði hreinsað. Eftir þetta er kjarninn tilbúinn og þá er cpu-schedulernum hleypt að til að taka við stjórn og þá er hægt að hlaða inn notendaumhverfinu.

Heimildir:

<https://web.archive.org/web/20080108195153/http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/s1-boot-init-shutdown-process.html>

<https://tecadmin.net/chicken-egg-problem-in-linux-and-initrd/>

<https://en.wikipedia.org/wiki/Booting>