Generiese Vaardighede: Linux Oefening

Abrie Greeff B.Sc Hons (Rekenaarwetenskap) Departement Rekenaarwetenskap Universiteit van Stellenbosch

17 April 2006

1 Inleiding

Die doel van hierdie oefening was om 'n persoon met 'n sterk kennis van linux en 'n persoon met 'n swak kennis saam te groepeer. Die ervare persoon moes deur middel van 'n verduideliking oor linux en 'n stel vrae die minder ervare persoon meer vertroud maak daarmee. Vir die oefening is ek gegroepeer saam met Mnr. W. Schulze. In die groep dien as ek die ervare persoon en hy as die minder ervare persoon. Saam het ons in [1] die volgende hoofstukke behandel: 1, 4, 5, 7-9, 25 en 44. Die tien vrae wat ek vir hom moes opstel het gekom uit: 8, 9, 25 en 44. In die volgende afdelings volg al die vrae en 'n verduideliking van hoekom ek hulle gekies het.

2 Vraag 1

Hoe sal jy kyk of 'n sekere ip adres op jou netwerk is?

Hierdie vraag is gekies, want dis belangrik dat enige persoon weet hoe om 'n ander rekenaar te *ping*. Dit laat jou toe om te agter te kom of hy jou kan sien en jy hom kan sien.

3 Vraag 2

Skryf 'n eenvoudige script wat 'n program as parameter kry en die uitvoer van die program na die tweede parameter ('n teks leer) toe skryf.

Dit is 'n baie eenvoudige voorbeeld om aan te toon hoe *piping* werk en ook op dieselfde tyd hoe *command line* parameters in 'n linux *script* werk. Dit is gekies omdat ek beide sien as 'n belangrike gedeelte van *bash scripts*.

4 Vraag 3

Doen die volgende.

Voer uit 'n eenvoudige program wat oneindig itereer.

Verander sy priority na +10.

Beeindig die program sonder om <Ctrl-C> te gebruik.

Wanneer enige een van ons 'n program uitvoer wat ons wil hê moet 'n groter of kleiner prioriteit hê as ander programme dan moet ons weet hoe om dit te verander. Ons wil ook partykeer weet hoe om 'n program te beëindig wat buite ons beheer geraak het. Hierdie oefening dek beide hierdie probleme.

5 Vraag 4

Stel die veranderlike X gelyk aan die PATH en die "current directory". Onderskei die twee waardes met 'n komma. Wys die waarde van X op die skerm

Linux bevat 'n hele aantal enviroment variables wat die verbruiker toelaat om meer van die sisteem te leer. Dit is veral handig wanneer dit in scripts gebruik word op rekenaars anders as die een waarop die script geskryf is. Deur te leer hoe om hierdie variables te gebruik kan scripts baie meer onafhanklik gemaak word. Hierdie oefening maak ook 'n nuwe veranderlike en leer ons hoe om dit op die skerm te vertoon.

6 Vraag 5

Wat is 'n "Denial of Service" aanval op 'n sisteem?

Dit is baie belangrik om te weet hoe jou rekenaar se sekuriteit gebreek kan word. Een voorbeeld van 'n aanval is die *Denial of Service* aanval. Deur bewus te raak van hierdie swak plekke in jou sisteem kan 'n persoon sy rekenaar beter leer verstaan.

7 Vraag 6

Hoe kan jy uitvind watter ports is oop op jou rekenaar? Waarom is dit belangrik om te weet?

Hierdie vraag sluit aan by die vorige vraag. Deur *netstat -nlp* te hardloop kan 'n persoon baie uitvind van moontlike oop poorte op sy/haar sisteem.

8 Vraag 7

Begin 'n program dadelik in die background van die shell. Bring die program na die shell en stop dit.

Dikwels is dit nodig dat 'n program baie lank moet uitvoer om bv. sy berekeninge te doen. Deur te weet hoe om 'n program in die agtergrond te hardloop kan so 'n program weg gesteek word en toegelaat word om te doen wat hy wil doen sonder om die gebruiker te pla.

9 Vraag 8

Vat alle c leers in 'n directory en skryf al die lyne wat die woord "printf" bevat na 'n teks leer "output".

Deur verskillende programme wat data tussen mekaar voer saam uit te voer kan 'n tydsame taak baie vereenvoudig word. Hierdie vraag soek so 'n moontlike antwoord. Dit is nog 'n voorbeeld van hoe *piping* gebruik kan word.

10 Vraag 9

Vat 'n c leer en met behulp van "sed" verwyder alle enkel lyn comments bv. "//ek is 'n comment" en skryf dit na 'n teks leer

Om teks in 'n groot leêr of selfs 'n aantal leêrs te vervang kan 'n lang proses wees. As hierdie proses geoutomatiseer word kan die probleem baie eenvoudiger gemaak word. 'n Program wat hierdie vir ons kan doen is sed. Die oefening is 'n voorbeeld van hoe dit gedoen kan word deur middel van regular expressions en sed.

11 Vraag 10

Wat is 'n buffer overflow attack?

Dit is belangrik om te weet hoe 'n program wat die gebruiker niks kennis het van nie sy sisteem kan aanval. 'n *Buffer overflow attack* is 'n baie algemene manier om dit te doen en dit is dus nodig om te weet hoe dit werk.

12 Verwysings

1. Rute