

## Lab 1.1 – BIV principes Game Ontwikkeling

### Doelen:

**Deel 1: In de ontwerpfase**

**Deel 2: Samenwerking met bedrijven**

**Deel 3: Opnemen van scènes in het buitenland**

### Achtergrond/Scenario

We gaan met dit Lab in op welke onderdelen van BIV aangeraakt worden, wanneer we ons verplaatsen in de onderstaande beknopte casus. Hierbij kijk je vanuit het perspectief van de gamedeveloper naar de aandachtspunten op het gebied van beschikbaarheid, integriteit en vertrouwelijkheid.

### Vereiste bronnen

#### Deel 1:

Een gamedeveloper heeft een game ontwikkeld voor de Apple Store. Ze hebben ook plannen voor een nieuw spel. Het nieuwe spel bevindt zich nog in de ontwerpfase. Er zijn al wat levels uitgetekend op papier.

*Welke BIV-principes zijn waar in bovenstaande tekst en waarom belangrijk?*

Noot vooraf: de vragen in dit specifieke Lab zijn niet in de laatste plaats om met elkaar het erover te hebben wat er bij een eenvoudige casus allemaal al kan spelen op BIV-gebied. Onderstaande uitwerking is zeker niet compleet of de enige waarheid, maar geeft een eerste aanzet waar je aan kan denken.

Qua *beschikbaarheid* is het (enkel) hebben van een papieren versie van de level-informatie riskant: als de papieren beschadigen of op een andere manier verloren gaan, gaat [een deel van] die creativiteit verloren.

Qua *vertrouwelijkheid* heb je dan wel meteen een basis: je moet fysiek naar de papieren toe om de gegevens in te zien. Er staat alleen nog niets beschreven over de bewaarplek van de ontwerpen zelf. Zo zou het gebruik van een kluis het oneigenlijk gebruik van de papieren bewerken.

## Deel 2:

De gamedeveloper heeft veel partijen waarmee zij samenwerkt. Zo werkt het bedrijf met echte acteurs die scenes spelen en met verschillende losse bedrijven die realistische 3d-modellen ontwikkelen. Met deze bedrijven wordt veel samengewerkt.

*Welke BIV-principes zijn waar en waarom belangrijk?*

Het samenwerken met derden heeft – natuurlijk – effect op de *vertrouwelijkheid*. Zo wil je als spellenmaker wellicht het plot van je verhaal niet vooraf richting de media gelekt hebben, dus is het handig om de acteurs een geheimhoudingsverklaring te laten tekenen.

Maar als je afzonderlijke acteurs puur informatie verstrekt op basis van ‘need-to-know’, kan het ook zijn dat ze de informatie anders interpreteren en hun acteerwerk niet helemaal goed in relatie staat tot jouw totaalplaatje. Wanneer ben je volledig (*integriteit*)?

## Deel 3:

De scenes worden in het buitenland opgenomen.

*Welke BIV-principes zijn hierbij en waarom belangrijk?*

De afgelopen jaren is er veel media-aandacht geweest rondom het plaatsen van datacentra, waarbij het lijntje tussen waar je hostpartij zich juridisch gevestigd heeft en waar de data opgeslagen is ook steeds minder duidelijk wordt. Zo vestigde Arjen Lubach met zijn item “Nederland als harde schijf” bijvoorbeeld de aandacht op grote partijen als Microsoft, Facebook en Google, waarbij werd verkondigd dat Facebook videodata van het Midden-Oosten in ons land zou opslaan.

Maar elk land (of samenwerkingsverband van landen) heeft weer eigen juridische kaders, en daaraan gekoppeld bijvoorbeeld ook een mate van vrijheid van meningsuiting (wat schurkt aan *integriteit*). Kun je er dan zomaar vanuit gaan dat de landen waarin je scènes opneemt geen enkele inzage willen hebben in jouw materiaal (dus ook *vertrouwelijkheid*)?

## Lab 1.2 – Checksum

### Doelen:

**Deel 1: Winhasher**

**Deel 2: Verifiëren van de checksum in Powershell en CMD**

**Deel 3: Checksum in macOS en Linux**

**Deel 4: De ISO aanpassen en opnieuw verifiëren van de checksum**

### Achtergrond/Scenario

Een checksum stelt je in staat om bepaalde fouten of informatieverlies te achterhalen. In het geval van een back-up kan het bijvoorbeeld voorkomen dat het opslagmedium defect aan het raken is. Door middel van een checksum weet je of je het oorspronkelijke bestand moet vervangen door de back-up, waarna je kan verifiëren of de back-up wel een identieke kopie is van het origineel.

### Vereiste bronnen

Winhasher, Windows 10 Powershell en Command Line Interface

### Deel 1: Winhasher (indien je met macOS of Linux werkt ga naar Deel 3)

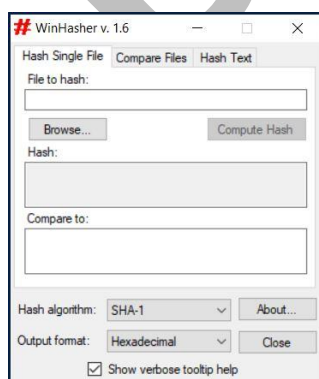
1. Download het programma WinHasher <https://github.com/gpfjeff/winhasher>

#### Download WinHasher

The following links point to the latest version of the WinHasher download files hosted on Google Drive. The GnuPG signature and SHA-1 hash for each file can also be found below. Jeff's current GnuPG signature can be found [here](#).

Download Type	GnuPG Signature	SHA-1 Hash	Size
<a href="#">Windows Installer</a>	<a href="#">Signature</a>	6620fae809b7fecb6dc18335065199d4a479cd88	389kb
<a href="#">Binaries w/o Installer</a>	<a href="#">Signature</a>	f5bc900380e3fb68f16f73e1dad4b867e3a34086	91kb
<a href="#">Source Archive</a>	<a href="#">Signature</a>	a8d7ff8f65415635900c418252f38191ae5da8b4	142kb

2. Installeer WinHasher en start daarna Winhasher op

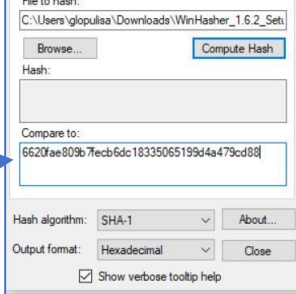


3. Kopieer de SHA1 Hash van de website waar je Winhasher hebt gedownload in het veld Compare to van Winhasher

### Download WinHasher

The following links point to the latest version of the WinHasher download files hosted on Google Drive. The GnuPG signature and SHA-1 hash for each file can also be found below. The GnuPG signature and SHA-1 hash for each file can also be found below. Jeff's current GnuPG signature can be found [here](#).

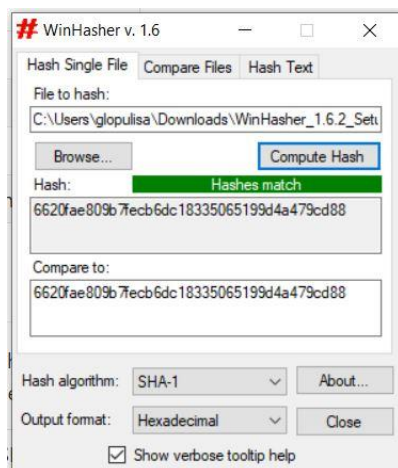
Download Type	GnuPG Signature	SHA-1 Hash	Size
<a href="#">Windows Installer</a>	<a href="#">Signature</a>	6620fae809b7fecb6dc18335065199d4a479cd88	389kb
<a href="#">Binaries w/o Installer</a>	<a href="#">Signature</a>	f5bc900380e3fb68f16f73e1dad4b867e3a34086	91kb
<a href="#">Source Archive</a>	<a href="#">Signature</a>	a8d7ff8f65415635900c418252f38191ae5da8b4	142kb



4. Klik op de button Browse en verwijst naar het setup bestand van Winhasher
5. Welk resultaat verwacht je als je de hash van het setup bestand gaat vergelijken met de hash die je net hebt gekopieerd?

De checksum zal hetzelfde moeten zijn

6. Klik op de button Compute Hash



7. Sluit Winhasher af

## Deel 2: Verifiëren van de checksum in Powershell en CMD

1. Start de Powershell op
2. Navigeer naar de folder waar het setup bestand van Winhasher in staat
3. Tik het volgende commando in:

```
Get-FileHash -Path <naam van het setup bestand> -Algorithm SHA1
```

Je commmando kan er ook zo uit zien:

```
Get-FileHash -Path .\Winhasher_1.6.2_Setup.exe -Algorithm SHA1
```

Bevind je je niet in de folder waar het setup bestand zich bevindt dan zal je commando er bijvoorbeeld zo uit kunnen zien:

```
Get-FileHash -Path C:\Users\Administrator\Downloads\Winhasher_1.6.2_Setup.exe -Algorithm SHA1
```

```
PS C:\Users\Downloads> Get-FileHash -Path .\WinHasher_1.6.2_Setup.exe -Algorithm SHA1
```

Algorithm	Hash	Path
SHA1	6620FAE809B7FECB6DC18335065199D4A479CD88	C:\Users\Downloads\...

4. Sluit de Powershell af door het commando Exit in te tikken
5. Open de CMD en navigeer naar de folder waar het setup bestand van Winhasher in staat
6. Tik het volgende commando in

```
certutil -hashfile <pad naar het bestand> SHA1
```

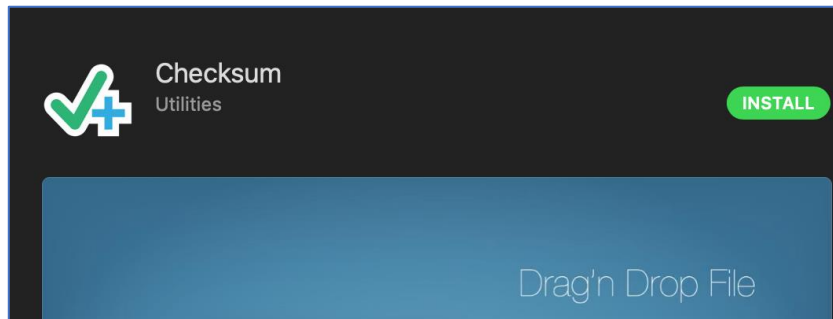
```
C:\Users\Downloads>certutil -hashfile WinHasher_1.6.2_Setup.exe SHA1
SHA1 hash of WinHasher_1.6.2_Setup.exe:
6620fae809b7fecb6dc18335065199d4a479cd88
CertUtil: -hashfile command completed successfully.
```



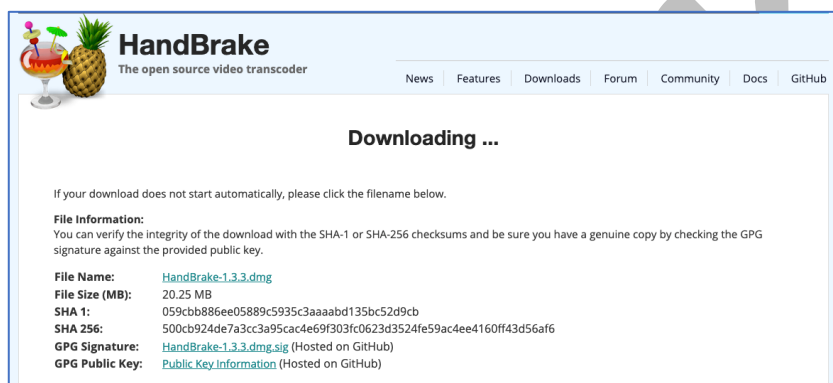
## Deel 3: Checksum in macOS en Linux

(Voor Linux ga naar stap 14)

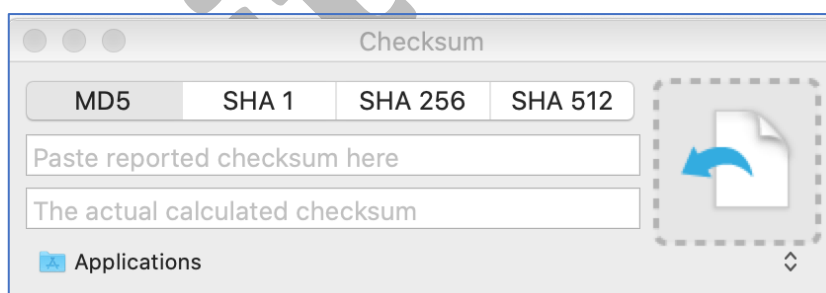
1. Download checksum uit de App store en installeer checksum



2. Download de App handbrake <https://handbrake.fr/rotation.php?file=HandBrake-1.3.3.dmg>

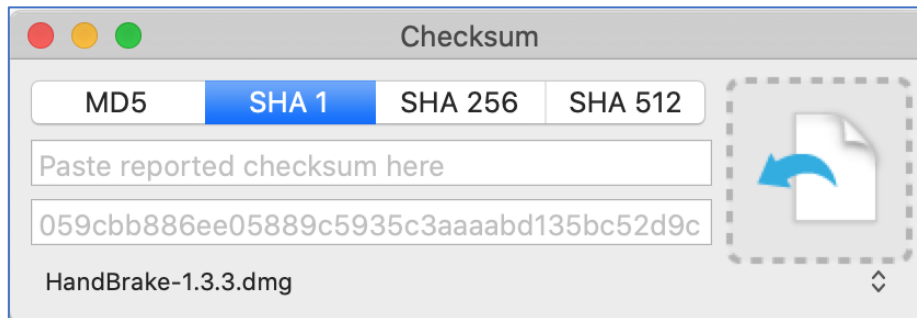
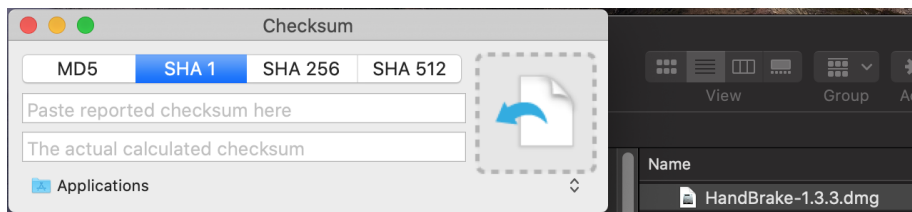


3. Start de app checksum op



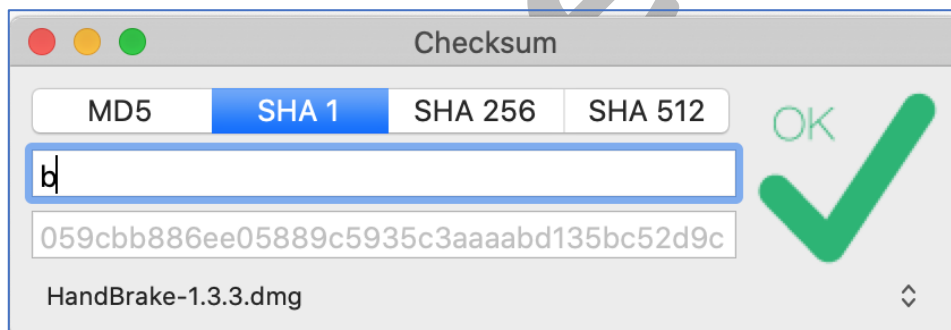
4. Selecteer de tab SHA 1

5. Sleep het bestand Handbrake-1.3.dmg naar checksum

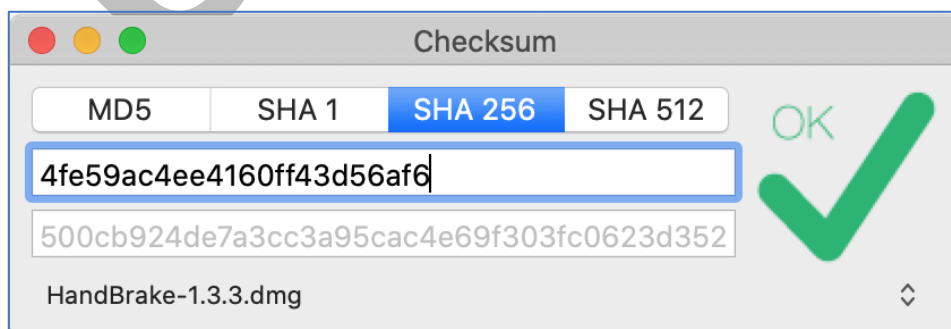


6. Kopieer nu van de website de SHA 1 checksum en plak dit in de app checksum

<b>File Name:</b>	<a href="#">HandBrake-1.3.3.dmg</a>
<b>File Size (MB):</b>	20.25 MB
<b>SHA 1:</b>	059cbb886ee05889c5935c3aaaabd135bc52d9cb
<b>SHA 256:</b>	500cb924de7a3cc3a95cac4e69f303fc0623d3524fe59ac4ee4160ff43d56af6

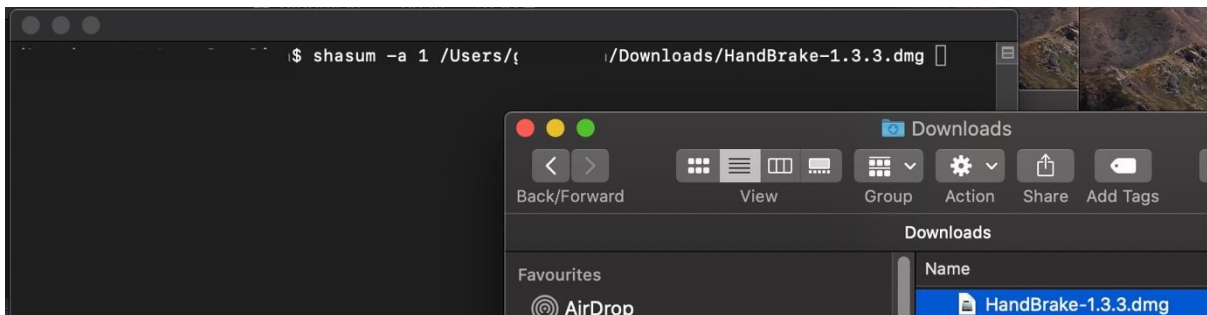


7. Doe nu hetzelfde voor de SHA 256 string



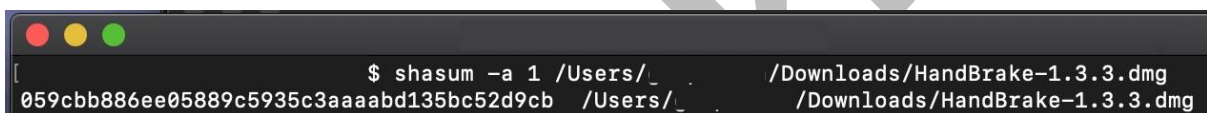
8. Sluit de app checksum af

9. Start nu de Terminal op
10. Tik nu het commando `shasum 1` in. Nu kan je het volledige pad intikken waar de dmg van Handbrake staat. Eenvoudiger is om een Findervenster te openen waar de dmg van Handbrake zich bevindt en de dmg naar de Terminal te slepen. Het padnaam wordt automatisch voor je ingevuld.



11. Welke resultaat verwacht je?

De checksum zal hetzelfde moeten zijn



12. Nu ga je de SHA256 hash controleren. Welk commando moet je gebruiken om de SHA256 hash te verifiëren?

`shasum -a 256 /padnaam/Handbrake-1.3.3.dmg`

13. Probeer het commando. Is de uitkomst wat je had verwacht?



<b>SHA 1:</b>	059cbb886ee05889c5935c3aaaabd135bc52d9cb
<b>SHA 256:</b>	500cb924de7a3cc3a95cac4e69f303fc0623d3524fe59ac4ee4160ff43d56af6

De checksum is hetzelfde

14. Voor deze oefening controleer je de MD5 en SHA1 hash van een Ubuntu minimal ISO. In dit voorbeeld wordt Ubuntu 14.04 LT "Trusty Tahr" gedownload.

#### 32-bit PC (i386, x86)

1. Ubuntu 18.04 "Bionic Beaver" 57MB (MD5: c7b21dea4d2ea037c3d97d5dac19af99, SHA1: a2a3b9c952ffa774ef77974e4e98ed5a9cdba2c8)
2. Ubuntu 16.04 LTS "Xenial Xerus" 48MB (MD5: 574fd244f5069f086065a23f7bdf604f, SHA1: 59211a88a125a7933c176365bb36b13197983ab7)
3. Ubuntu 14.04 LTS "Trusty Tahr" 31MB (MD5: a2502844750ecb6477d8fb4ff6b9aaf8, SHA1: d17c34ce716f13396040ccdc02d32482ed6b01a1)

15. Start de terminal van Linux op.





16. Gebruik je een Desktop editie van Linux dan kan je de link kopiëren en plakken in de Terminal. Gebruik je een Servereditie dan zal je de link handmatig moeten intikken

```
wget http://archive.ubuntu.com/ubuntu/dists/trusty/main/installer-i386/current/images/netboot/mini.iso
```

```
winuser@DESKTOP-LVOAFTI:~$ wget http://archive.ubuntu.com/ubuntu/dists/trusty/main/installer-i386/current/images/netboot/mini.iso
--2020-10-29 17:10:19-- http://archive.ubuntu.com/ubuntu/dists/trusty/main/installer-i386/current/images/netboot/mini.iso
Resolving archive.ubuntu.com (archive.ubuntu.com)... 2001:67c:1562::18, 2001:67c:1360:8001::23, 2001:67c:1562::15, ...
Connecting to archive.ubuntu.com (archive.ubuntu.com)|2001:67c:1562::18|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32505856 (31M) [application/x-iso9660-image]
Saving to: 'mini.iso'

mini.iso          9%[====>] 2.91M  576KB/s  eta 51s
```

17. Tik het commando `ls` in om te verifiëren dat het bestand is gedownload

```
winuser@DESKTOP-LVOAFTI:~$ ls
mini.iso
winuser@DESKTOP-LVOAFTI:~$
```

18. Tik het volgende commando in:

```
shasum mini.iso
```

```
winuser@DESKTOP-LVOAFTI:~$ shasum mini.iso
d17c34ce716f13396040ccdc02d32482ed6b01a1 mini.iso
```

19. Vergelijk de uitkomst met de checksum op de website

#### 32-bit PC (i386, x86)

1. **Ubuntu 18.04 "Bionic Beaver"** 57MB (MD5: c7b21dea4d2ea037c3d97d5dac19af99, SHA1: a2a3b9c952ffa774ef77974e4e98ed5a9cdba2c8)
2. **Ubuntu 16.04 LTS "Xenial Xerus"** 48MB (MD5: 574fd244f5069f086065a237bdf604f, SHA1: 59211a88a125a7933c176365bb36b13197983ab7)
3. **Ubuntu 14.04 LTS "Trusty Tahr"** 31MB (MD5: a2502844750ecb6477d8fb4ff6b9aaf8, SHA1: d17c34ce716f13396040ccdc02d32482ed6b01a1)

20. Tik nu het volgende commando in:

```
md5sum mini.iso
```

21. Vergelijk de uitkomst met de checksum op de website

```
winuser@DESKTOP-LVOAFTI:~$ md5sum mini.iso
a2502844750ecb6477d8fb4ff6b9aaf8 mini.iso
```



## Deel 4: De ISO aanpassen en opnieuw verifiëren van de checksum

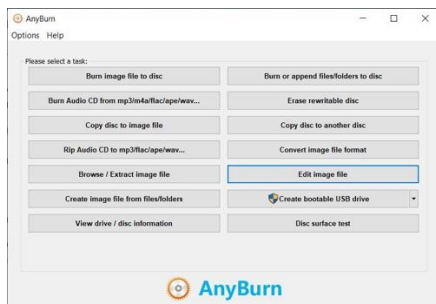
Werk voor deze opdracht in tweetallen waarvan een van jullie een Windows laptop heeft. De bedoeling is dat de ISO gedeeld m.b.v. bijvoorbeeld google drive.

1. Maak een tekstbestand aan en sla dit op.
2. Download de Ubuntu minimal ISO (Ubuntu 14.04 LTS "Trusty Tahr")

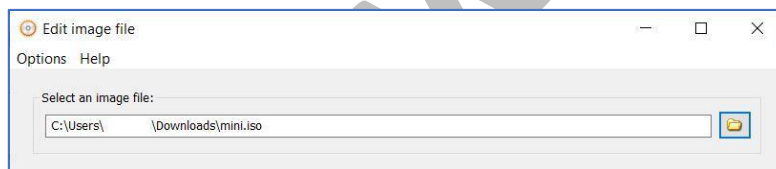
### 32-bit PC (i386, x86)

1. **Ubuntu 18.04 "Bionic Beaver"** 57MB (MD5: c7b21dea4d2ea037c3d97d5dac19af99, SHA1: a2a3b9c952ffa774ef77974e4e98ed5a9cd8a2c8)
2. **Ubuntu 16.04 LTS "Xenial Xerus"** 48MB (MD5: 574fd244f5069f086065a23f7bdf604f, SHA1: 59211a88a125a7933c176365bb36b13197983ab7)
3. **Ubuntu 14.04 LTS "Trusty Tahr"** 31MB (MD5: a2502844750ecb6477d8fb4ff6b9aaf8, SHA1: d17c34ce716f13396040ccdc02d32482ed6b01a1)

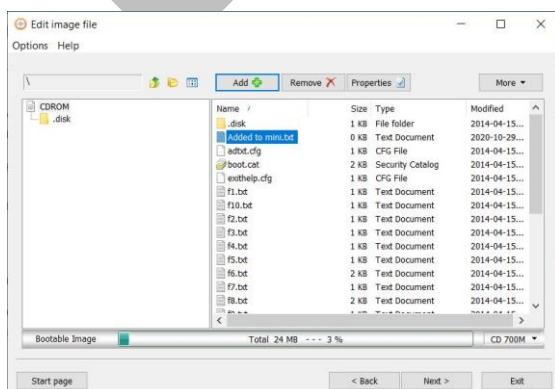
3. Download AnyBurn voor Windows 10 [http://www.anyburn.com/anyburn\\_setup\\_x64.exe](http://www.anyburn.com/anyburn_setup_x64.exe)
4. Installeer en start AnyBurn
5. Klik op de button Edit Image File



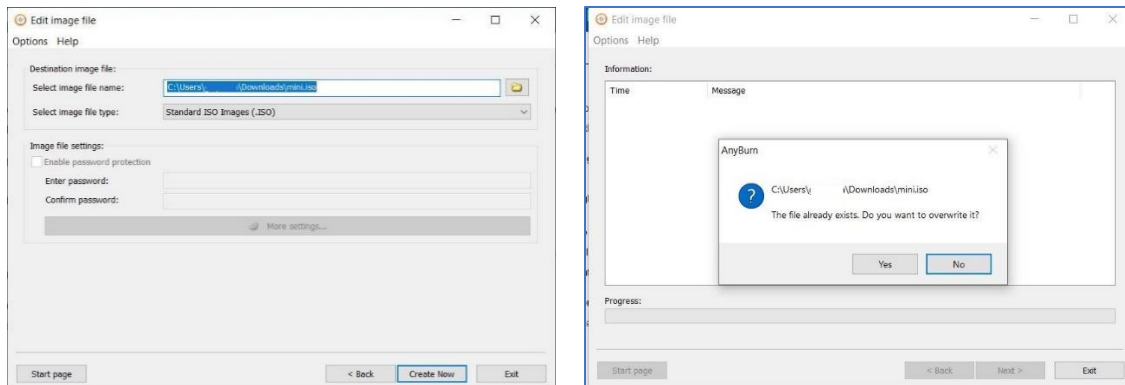
6. Kies in het volgende venster voor de mini.iso en klik op next



7. Klik in het volgende venster op de button Add en voeg het eerder gemaakte tekstbestand toe en klik op next



8. In het volgende venster klik je op de button Create Now. De iso wordt opnieuw gemaakt. Er verschijnt een melding dat het bestand reeds bestaat. Bevestig met Yes



9. De mini.iso is gemaakt, maar nu met een extra bestand!  
10. Wissel het bestand uit met je groepgenoot  
11. Controleer nu de checksum voor MD5 en SHA1 met een van de eerdere methoden  
12. Wat verwacht je?

De checksum is niet hetzelfde gebleven

13. Waar is m.b.t. deze ISO een aanval op uitgevoerd als je kijkt naar BIV en waarom is dat?

Op de Integrity. De inhoud van de iso is veranderd.

14. Stel dat een hacker kwaadaardige code in de iso heeft verstopt, waar zou je naast de checksum ook allemaal op kunnen letten?

De hacker zou ervoor moeten zorgen dat jij denkt dat je deze iso download vanaf de officiële website, maar in feite wordt de iso van de server van de hacker gedownload. Je zou kunnen controleren op de link van het bestand of deze niet verdacht is. Is de ISO eventueel van andere websites te downloaden? Je zou de bestandsgrootte kunnen vergelijken. Een hele goed hacker zou ervoor kunnen zorgen dat de officiële website gekaapt zou kunnen zijn. In hoeverre ben jij in staat om een gekaapte website te herkennen? Maakt de website gebruik van een certificaat? Indien het onveilige verbinding zou je dan wel of niet de iso downloaden?

## Lab 1.3 – Beschikbaarheid

### Doelen:

**Deel 1: Berekening beschikbaarheid**

**Deel 2: Berekening beschikbaarheid met redundantie**

### Achtergrond/Scenario

Een IT-infrastructuur is (ooit) gebouwd op basis van opgestelde requirements. Deze requirements zijn meestal opgesteld door één of meerdere stakeholders.

Stakeholders willen dat de IT-infrastructuur een bepaalde mate van beschikbaarheid moeten hebben. Data-exchange van, naar en binnen de IT-infrastructuur moet een bepaalde mate van integriteit en vertrouwelijkheid hebben.

Een hostingbedrijf wil een zo hoog mogelijke beschikbaarheid hebben in een datacenter. Een zeer ambitieus streven is een beschikbaarheid van 99,999%. Dit houdt in dat de uitval of downtime ongeveer 5 minuten op jaarbasis is. Om de beschikbaarheid zo hoog mogelijk te maken worden veel servers redundant uitgevoerd met allerlei fail-over technieken.

### Vereiste bronnen

Excel

## Deel 1: Berekening beschikbaarheid

Een leverancier biedt de volgende devices:

- A. Firewall 99,995% €3900,- per jaar
- B. Webserver WH18017 90% € 5000,- per jaar
- C. Databaseserver DB18017 81% €2600,- per jaar

Om de beschikbaarheid uit te rekenen van bijvoorbeeld de firewall en webserver WH18017 vermenigvuldig je de beide beschikbaarheden:

Beschikbaarheid A x beschikbaarheid B... x beschikbaarheid n

Voor dit voorbeeld is dat dus  $0,99995 * 0,9 = 0,899955$ , oftewel een beschikbaarheid van 89,9955% (zorg ervoor dat je in excel ongeveer 10 decimalen achter de komma laat staan)

$24 * 365 = 8760$  uren in een jaar

1% is 87,6 uur

Een systeem met een beschikbaarheid van 89,9955% mag dus ongepland 876 uur **onbeschikbaar** zijn. Dit zijn ongeveer 36,5 dagen per jaar!.....

Reken dit voorbeeld na, maar nu voor de firewall, webserver en databaseserver



89,9955% beschikbaarheid  
 876,438 uren ongepland onbeschikbaar p/j  
 36,51825 volle dagen p/j

(F) Firewall	99,99500%	€ 3.900,00
(W) Webserver WH18017	90,00000%	€ 5.000,00
(D) Databaseserver DB18017	81,00000%	€ 2.600,00

#### Deel 1: Berekening beschikbaarheid

(F) Firewall	99,99500%	€ 3.900,00	
(W) Webserver WH18017	90,00000%	€ 5.000,00	
Beschikbaarheid (F * W)	89,99550%	€ 8.900,00	Totaal

10,00450% Onbeschikbaar

24 * 365 dagen	8760	uren in een jaar
1%	87,600	uur
	7883,562	uren beschikbaar p/j
	876,438	uur niet beschikbaar p/j
	36,51825	dagen niet beschikbaar p/j

(F) Firewall	99,9950000000%	€ 3.900,00	
(W) Webserver WH18017	90,0000000000%	€ 5.000,00	
(D) Databaseserver DB18017	81,0000000000%	€ 2.600,00	
Beschikbaarheid (F * W * D)	72,8963550000%	€ 11.500,00	Totaal

27,10365% Onbeschikbaar

24 * 365 dagen	8760	uren in een jaar
1%	87,600	uur
	6385,720698	uren beschikbaar p/j
	2374,279302	uur niet beschikbaar p/j
	98,92830425	dagen niet beschikbaar p/j



## Deel 2: Berekening beschikbaarheid met redundantie

Bereken de beschikbaarheid in de volgende situatie (redundantie) en gebruik de gegevens uit lab 1.3.

- A. 2 x Firewall
- B. 2 x Webserver
- C. 2 x Databaseserver

De formule die je hiervoor nodig hebt:

$$(1 - (1 - \text{beschikbaarheid A})^{\text{aantal}}) * (1 - (1 - \text{beschikbaarheid B})^{\text{aantal}}) * (1 - (1 - \text{beschikbaarheid C})^{\text{aantal}})$$

Antwoord:

**95,4260997614347000%** beschikbaarheid

**400,6736609** uren ongepland onbeschikbaar p/j

**(16,69473587** volle dagen p/j)

Uitleg:

De firewall is van alle drie de componenten (firewall, webserver en databaseserver) veruit het meest betrouwbaar qua beschikbaarheid: 99,995% van de tijd is de firewall actief. Als je daar een tweede firewall 'naast' zet, dan is de *gecombineerde* beschikbaarheid hoger. Immers, als de eerste firewall eens onbeschikbaar is, dan is de kans dat de tweede firewall wel de taken op zich kan nemen natuurlijk levensgroot (ervan uitgaande, dat apparaten elkaar niet 'aansteken' met hun downtime). Anders gezegd: de niet-beschikbaarheid is kleiner wanneer je meerdere firewalls parallel actief hebt.

Bij één webserver is de niet-beschikbaarheid iets groter: namelijk 10% (want: 90% beschikbaar). Dan zorgt een tweede webserver weer voor een verhoging van de beschikbaarheid: wanneer de eerste webserver down is, is de kans dat de tweede webserver die downtime kan opvangen natuurlijk nog steeds 90%. De kans dat ze *beide tegelijkertijd* niet beschikbaar zijn op moment X, is  $10\% * 10\% = 0,1 * 0,1 = 0,01$  (oftewel: 1%).

*Als je helemaal duur gaat doen en je kiest voor 3 webserver die tegelijkertijd (parallel) worden ingezet, dan daalt het percentage niet-beschikbaar zelfs tot  $0,1 * 0,1 * 0,1 = 0,001$  (oftewel: 0,1%).*

Met  $(1 - \text{beschikbaarheid B})^{\text{aantal}}$  bereken je dus de verbeterde (=verlaagde) niet-beschikbaarheid van de webserverfunctie, wanneer je deze parallel een aantal keer uitvoert.

Dat is fijn, maar we willen eigenlijk weten hoeveel procent een functie WEL beschikbaar is, oftewel dat er (ten minste) één van de twee webserver op moment X beschikbaar is. Vandaar dat vooraan de gegeven formule (nogmaals) "1-" staat.

De beschikbaarheid met deze configuratie is  $1 - 0,01 = 0,99 \rightarrow 99\%$ .

Dit werkt hetzelfde voor de databaseserver, met dien verschil dat de aangegeven beschikbaarheid per server 'slechts' 81% is.

$$\begin{aligned} & (1 - (1 - 0,99995)^2) * (1 - (1 - 0,90)^2) * (1 - (1 - 0,81)^2) = \\ & (1 - (0,00005)^2) * (1 - (0,10)^2) * (1 - (0,19)^2) = \\ & (1 - 0,0000000025) * (1 - 0,01) * (1 - 0,0361) = \\ & 0,9999999975 * 0,99 * 0,9639 = 0,954260997... \\ & \hspace{15em} \text{(ongeveer 95,42\%)} \end{aligned}$$



Maar ja, het kost wel wat extra om van elk type een tweede neer te zetten..

(F) Firewall	99,99500%	€ 3.900,00
(W) Webserver WH18017	90,00000%	€ 5.000,00
(D) Databaseserver DB18017	81,00000%	€ 2.600,00

1	Beschikbaarheid	Per jaar	Aantal		Totaal
Firewall	0,9999500	€ 3.900,00	2	1,0000000	€ 7.800,00
Webserver WH18017	0,9000000	€ 5.000,00	2	0,9900000	€ 10.000,00
Databaseserver DB18017	0,8100000	€ 2.600,00	2	0,9639000	€ 5.200,00
		Totaal			€ 23.000,00
					95,4260997614347000%

95,4260997614347000

	4,57390%	Onbeschikbaar
24 * 365 dagen	8760	uren in een jaar
1%	87,600	uur
	8359,326339	uren beschikbaar p/j
	400,6736609	uur niet beschikbaar p/j
	16,69473587	dagen niet beschikbaar p/j



Welke conclusie moet je trekken als je een beschikbaarheid van 99,99(9) % zou willen halen?

Dat betekent dat je meerdere firewalls en/of meerdere webserver en/of databaseservers moet inzetten en ook moet gaan kijken wat de meest goedkope oplossing zou zijn. In dit voorbeeld zijn er 6 webserver en 7 databaseservers gekozen. De vraag is of je dit wel zo zou moeten doen. Misschien moeten er juist meer webserver worden gekozen dan de databaseservers, of meer firewalls en minder databaseservers. Is het voldoende om toch met twee firewalls verder te gaan? Eventueel een andere type webserver die veel meer verkeer requests aankan waardoor je toch minder webserver nodig hebt dan het aantal databaseservers.

(F) Firewall	99,99500%	€ 3.900,00
(W) Webserver WH18017	90,00000%	€ 5.000,00
(D) Databaseserver DB18017	81,00000%	€ 2.600,00

1

1	Beschikbaarheid	Per jaar	Aantal		Totaal
Firewall	0,9999500	€ 3.900,00	2	1,0000000	€ 7.800,00
Webserver WH18017	0,9000000	€ 5.000,00	6	0,9999990	€ 30.000,00
Databaseserver DB18017	0,8100000	€ 2.600,00	7	0,9999911	€ 18.200,00
		Totaal			€ 56.000,00
					99,9990058791574000%

99,9990058791574000

	0,00099%	Onbeschikbaar
24 * 365 dagen	8760	uren in een jaar
1%	87,600	uur
	8759,912915	uren beschikbaar p/j
	0,087084986	uur niet beschikbaar p/j
	0,003628541	dagen niet beschikbaar p/j

