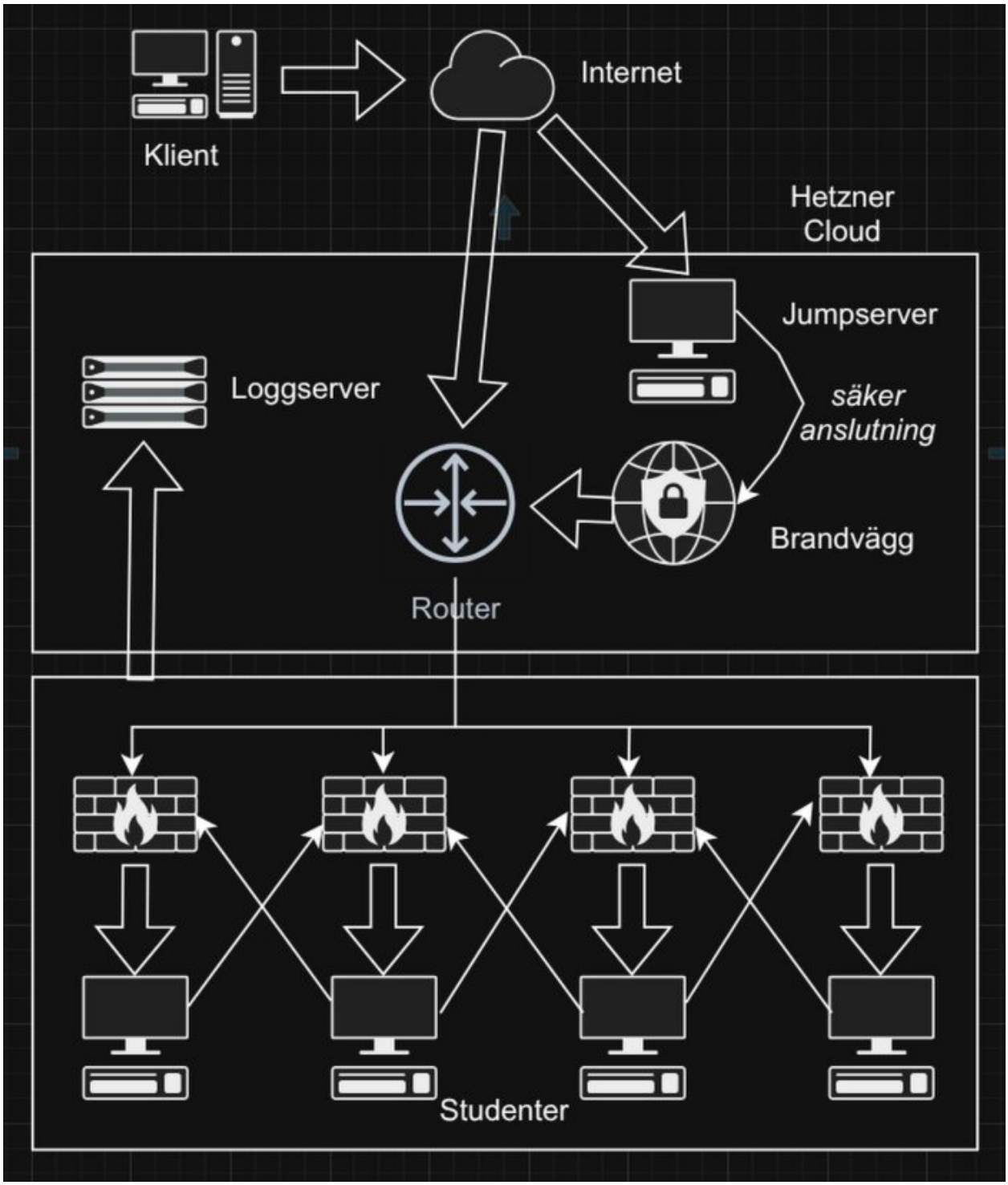
Uppgift inom netowkring och zero trust

Från nuläge till säker målarkitektur

Steg 1 - Nuläge



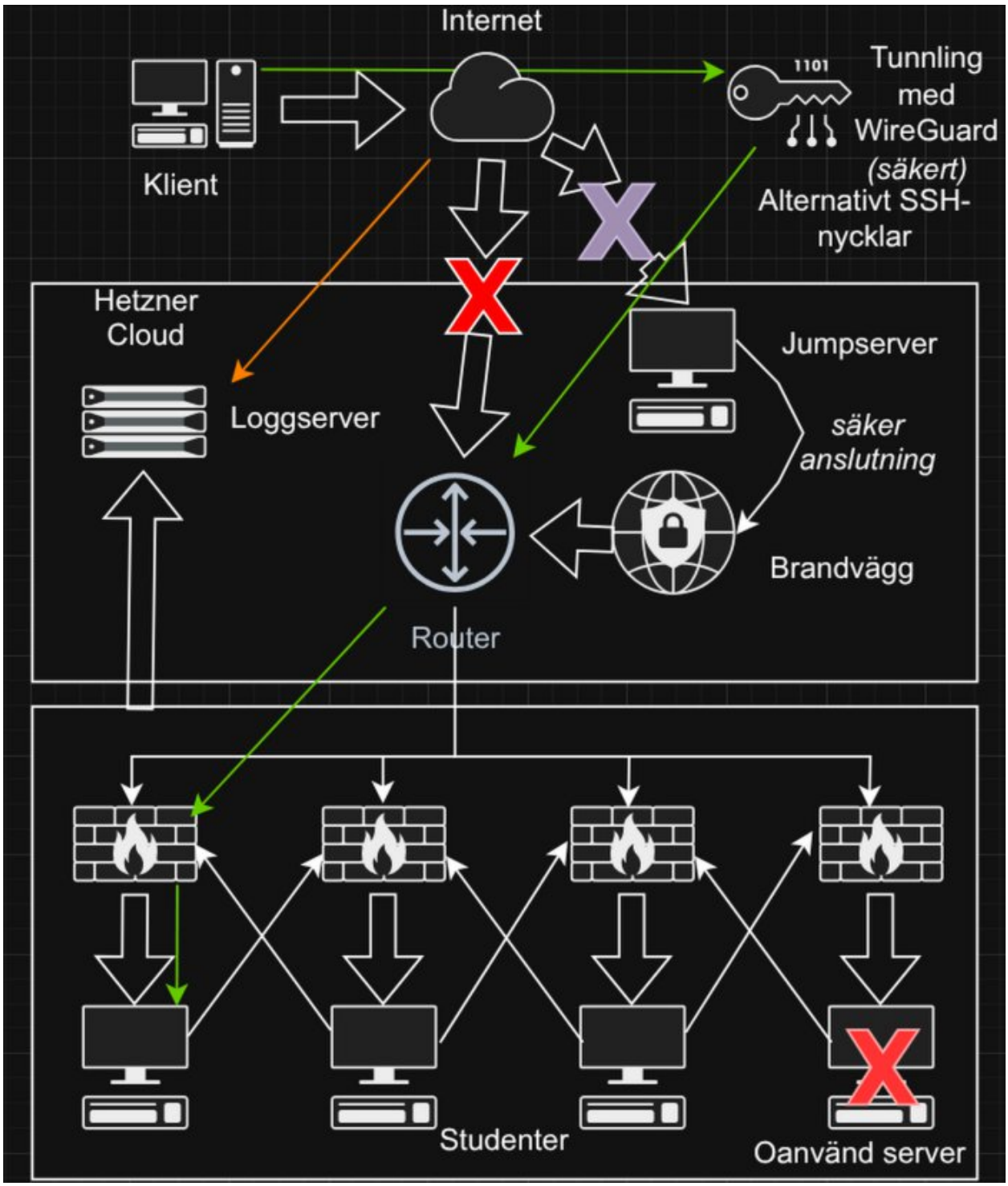
Beskrivning

När vi fick servrarna och innan vi började använda alla häftiga tjänster som OPNsense och WireGuard, så hade vi egentligen bara vår lilla server på Hetzners nät. Våra servrar hade inga brandväggar, och vi loggade in med SSH via port 22. Alla hade samma lösenord (har hört att inte alla studenter har bytt lösenord än för övrigt).
Vi fick själva ansvara för att aktivera en brandvägg vilket resulterade i att flera blev utelåsta när de aktiverade både UFW och firewalld.
Kommunikationen sker via SSH. Vi kunde ansluta direkt till våra servrar eller via en jumpserver.
Loggar skickades (och skickas kanske fortfarande för vissa) internt på nätverket till en loggserver. Jag är ganska säker på att vi hade ställt in det när vi fick den här uppgiften.
Vi kunde logga in på loggservern och läsa loggar i /var/log/remote/ men inte göra mycket annat på den servern.
Från våra studentservrar kunde vi kontakta andras servrar, men det troligaste var att vi stoppades ganska snabbt i brandväggen.
Några sårbarheter beror på vår okunskap. Man kan säkra upp SSH med nycklar och att byta SSH-port och stänga av möjligheten att logga in med lösenord och som root.
Brandväggarna är inte optimalt inställda. Flera har fortfarande port 80 öppen utan egentlig anledning till exempel.
Loggservern går att komma åt utifrån. Jag vet inte om det är nödvändigt, men den är exponerad mot internet.
Jumpservern är även den exponerad, och jag tror inte vi har behörigheter att härda den anslutningen.
Nätverket är, som uppgiften nämner, inte segmenterad ordentligt. Det kan vara krångligt att göra i det här skedet när vi har våra Boiler Room-grupper. Alla kan se andras servrar på nätverket.
Hetzner har en brandvägg, men det känns lite som att den inte gör så mycket för det mesta, och därför har jag ritat pilar mot routern snarare än brandväggen. Tekniskt sett går väl all kommunikation via brandväggen, men brandväggarna på våra servrar verkar göra mer.

Steg 2 - Förbättrad design

Vad fungerar bra idag?
Vi har lärt oss använda tunnlar som WireGuard. Våra brandväggsregler blir bättre hela tiden. Vi kan övervaka loggar.
Personligen har jag börjat använda SSH-nycklar vid vanlig inloggning, och har bytt SSH-port till en som skannas mycket mer sällan.
Vi har Fail2Ban som blockerar påstridiga intrångsförsök.

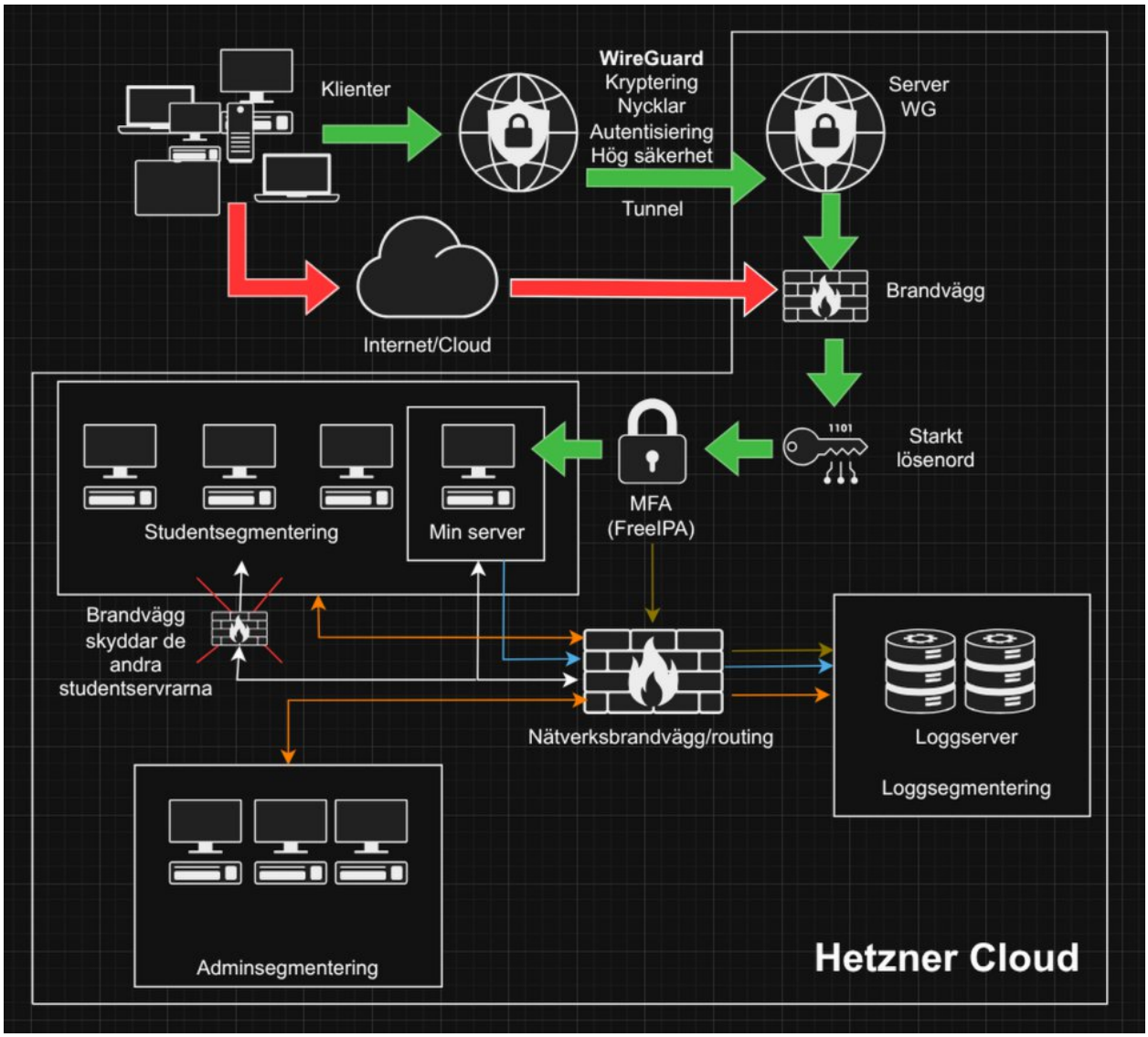
Vilka delar bör förbättras?
Några studenter hoppade av efter att de tilldelades studentservrar, men innan de bytte lösenord eller härdade servern. De servrarna ligger nu exponerade mot internet och det finns en reell risk att de kan utsättas för lyckade intrångsförsök varpå en angripare får tillgång till vårt interna nät.
Vi bör utbildas i att använda säkra SSH-nycklar vid anslutning. Jumpservrar och VPN-tunnlar i all ära, men SSH-nycklar är väldigt säkra för våra syften.
Om utbildarna anser att tunnling är det säkrare alternativet (nycklar används även där, men det är lite krångligare att ställa in på en ny maskin) så bör det pushas på att vi använder WireGuard för all inloggning.
Skillnaden mellan ”bara” SSH-nycklar och WireGuard är att nycklarna kan man spara på en säker plats online och ladda ner och använda smidigt. WireGuard måste man installera, ställa in alias för, generera eller skaffa gamla nycklar, och vara säker på att WireGuard körs på servern.
Å andra sidan använder WireGuard UDP, vilket är säkrare på det sättet att det inte går läsa av vad det är för kommunikation som sker. Det är en avvägning man får göra.



Beskrivning
Min tanke är att stänga av möjligheten att ansluta direkt till servern utan SSH-nycklar.
Min åsikt är att jumpservern inte ”behövs” när vi har möjligheten och kompetensen att säkra vår anslutning på andra sätt. Vi kan inte härda jumpservern och då är den sårbar.
Om det inte finns bra anledningar så bör man fundera på att stänga av möjligheten att ansluta till loggservern utifrån nätverket. Det är samma där som med jumpservern, vi studenter har inga möjligheter att härda den.
Servrar som har tilldelats men inte längre används bör stängas ner för att minska attackytan.

Steg 3 - Målarkitektur

Zero Trust:
MFA - FreeIPA, SSH-nycklar, lösenord
Least Privilege - Tillåt inte inlogg som root, ge inte mer behörigheter än som verkligen behövs
Network Segmentation - Dela upp nätverket i logiska delar
Strict Firewall - Så få öppna portar som möjligt
Logging - Logga allt på den lokala servern, skicka till en central loggserver
Monitoring - Varna vid upprepade misstänkta händelser (misslyckade inloggningar med mera)
Encryption - Endast anslutning via krypterade kanaler som SSH och med nycklar



Beskrivning

För att logga in på min server behöver jag i den här arkitekturen först tunnla mig via WireGuard. Min tunnel kommer till brandväggen (min servers i det här fallet, men den kanske går vid nätverkets brandvägg först) som släpper igenom mig då jag ansluter från en känd IP.
När brandväggen släppt igenom mig måste jag ange lösenord, följt av min MFA på FreeIPA.
I det här scenariot har jag tagit mig igenom fyra kontroller: WireGuard med rätt nycklar, brandvägg som kontrollerar vilken IP jag ansluter från, lösenord och MFA. Det borde vara oerhört säkert, om än en aning krångligt för användaren att ställa in WireGuard från första början och ange en extra kod varje gång hen vill logga in.

Servern är nu segmenterad. Man kan tänka sig att studenternas servrar är segmenterade ytterligare i våra Boiler Room-grupper. Loggservern är på sin egen segmentering. Administratörerna är på sin egen segmentering.

Alla segmenteringar har behörighet att skicka loggar till loggservern, men ingen har rättighet att redigera loggarna, utom möjligtvis administratörerna men jag ser inte några anledningar till att det ska behövas.

Loggarna övervakas och vid misstänkt aktivitet larmas administratörerna.

Administratörerna kan komma åt studentservrarna vid behov, men det finns givetvis policy och regler att följa. Anledningen är att vi fortfarande är väldigt bra på att låsa oss ute. När vi är mer kompetenta kommer behovet minska.

Studenterna bör uppmuntras att stänga portar som är öppna i onödan, byta SSH-port, och om möjligt dölja SSH-porten.

Det bör inte finnas något sätt för användare att komma åt servern direkt via SSH eller andra vägar som inte är WireGuard. På nätverket bör exempelvis tillgång till loggservern begränsas från utsidan. Vill man in och läsa loggar ska man göra det från sin egen server.

Kostnadsestimering

Den största kostnaden kommer vara lön, då tjänsterna och mjukvaran som används är billig eller till och med gratis. Om man ska uppskatta ungefär så skulle jag säga att fem personer skulle kunna klara jobbet på två veckor. Med en månadslön på 50000 för att det ska vara lätt att räkna och att det är fyra veckor på en månad så är beräkningen:

(50000 x 0.5) x 5 = 12500

Så kostnaden för att de fem personerna ska göra jobbet blir ungefär 125000 kronor. Men då vet jag inte hur lång tid arbetet egentligen tar, hur många som egentligen behöver sitta med det, eller vad mjukvaran lägger till det beloppet. Man får dessutom tänka på att jobbet kan dra ut på tiden, eller ta mindre tid än beräknat.

Det kommer dock krävas mycket träning för att användarna ska första varför man måste använda säkerhetslagren de får tillgång till. Många tänker sig inte för, och det kommer alltid finnas risk att rutinerna inte följs.