TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Mihkel Kiil 201748IVSB

Secure Offsite Data Backup Solution for a Small Business

Bachelor's thesis

Supervisor: Aleksei Talisainen
MSc

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Mihkel Kiil 201748IVSB

Turvaline andmevarunduslahendus väikeettevõttele

Bakalaureusetöö

Juhendaja: Aleksei Talisainen
Magister

Tallinn 2023

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Mihkel Kiil

10.05.2023

Abstract

The purpose of this thesis is to propose a solution to the problem which is the lack of offsite backup in a small business. Several solutions are available from commercial backup service providers; however, the prototype solution provided in this thesis would give more control and oversight to the business by using the business's own hardware. The solution was implemented in a test environment and is designed to be used in a business of up to roughly 50 employees. The solution was designed with information security at its core and delivers an overall improvement to availability, integrity, and confidentiality of data.

The backup system uses Duplicity software to compress and encrypt the data and maintain versioning information. The backup server is intended to be in a separate physical location from the main data storage location to ensure data integrity and availability even in the event of a disaster at the main location. The backup system stores multiple versions of data to allow integrity to be restored in case of tampering. To transfer the data offsite, an OpenVPN client was deployed on the server to connect to the main location. Confidentiality is provided by encryption of the backup copies at rest and in transit. The thesis contains a detailed guide on implementing the designed solution. The process of restoring the data from a backup was also successfully tested.

It was determined that the proposed solution is viable, given that the business has at least two locations which can house the servers, as well as one on-premises server already in use. However, the author has concluded that businesses who only have one location or no file server currently should opt for cloud-based solutions instead.

This thesis is written in English and is 45 pages long, including 7 chapters and 15 figures.

Annotatsioon

Käesoleva lõputöö eesmärk on pakkuda välja lahendus probleemile, milleks on füüsiliselt eraldiseisva varundussüsteemi puudumine väikeettevõttes, millel on kuni 50 töötajat. Antud probleemile on olemas mitmeid lahendusi teenusepakkujate poolt, kuid töös välja toodud lahendus annab ettevõttele rohkem kontrolli selle toimimise üle. Lahendust saab kasutada ettevõtte olemasoleva arvutiriistvaraga ja selleks vajalik tarkvara on tasuta ning avatud lähtekoodiga. Välja pakutud lahenduse oluliseks osaks on infoturve.

Loodud varunduslahendus kasutab Duplicity tarkvara varukoopiate haldamiseks. Server, kus varukoopiaid talletatakse, on mõeldud paiknema eraldi asukohas ettevõte esmasest serverist, et tagada andmete terviklikkus ja kättesaadavus ka õnnetusjuhtumi kohal esmase serveri asukohas. Konfidentsiaalsust aitab tagada varukoopiate krüpteerimine. Lõputöö sisaldab juhendit lahenduse paigaldamiseks. Autor on edukalt läbi viinud andmete taastamiskatse loodud lahendusega.

Töös on järeldatud, et antud lahendus sobib ettevõtetele, millel juba on olemas vähemalt üks failiserver ning millel on mitu kontorit vm asukohta, kuhu servereid majutada. Ettevõttetele, millel neid võimalusi ei ole, soovitab autor pilvepõhiseid varunduslahendusi.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 45 leheküljel, 7 peatükki ja 15 joonist.

List of abbreviations and terms

BaaS	Backup as a service
FTP	File transfer protocol
FTPS	File transfer protocol secure
GB	Gigabyte
GPG	GNU Privacy Guard
GUI	Graphical user interface
HDD	Hard disk drive
IKE	Internet key exchange
IP	Internet protocol
IPSec	Internet protocol security
ISP	Internet service provider
IT	Information technology
L2TP	Layer 2 tunnelling protocol
Mb/s	Megabits per second
NAS	Network attached storage
NAT	Network address translation
NIC	Network interface card
On-premises	Within the complete set of locations that are considered business premises
Onsite	Within a single location or branch of a business
OS	Operating system
PC	Personal computer
RAID	Redundant array of independent disks
RHEL	Red Hat Enterprise Linux
SCP	Secure copy
SFTP	Secure file transfer protocol
SSD	Solid state drive
SSH	Secure shell
TB	Terabyte

VPN	Virtual private network

Table of contents

List of figures

1 Introduction

1.1 Definitions

A backup copy, or backup, is a secondary copy of data taken and stored elsewhere to be used to recover from a data loss. The process of creating a backup is known as “backing up”. As the use of computers and other electronic devices in businesses has grown, so has the need to store and back up business data securely. Solutions to back up this data are provided by many companies.

It should be noted that backing up does not mean only storing it in a separate location, and a backup solution should also be capable of restoring a file from a previously saved state. Therefore, cloud storage solutions such as Google Drive and Microsoft OneDrive that do not offer this functionality when used alone are not considered in this thesis.

Another important note is the “3-2-1 rule”. This rule states that there should be 3 copies of all data, on 2 different media, 1 of them in a separate physical location [1]. Therefore, a simple copy of a workstation’s data on a server does not comply with the 3-2-1 rule. To comply, an offsite backup is required. This can be either in the form of a backup provider or a business may choose to use their own second premises and their own hardware for the purpose.

1.2 Problem statement

The problem this thesis aims to solve is the absence of offsite backup in an arbitrary small business of up to 50 employees. As commercial backup providers offer their own solutions and there are a multitude of ways of setting it up a solution using one’s own hardware, there is no one way to solve this problem. IT infrastructure is a crucial part of operations for almost any business and a secure offsite backup is a vital part of it.

The thesis will detail how this problem can be solved using on-premises hardware and free, open-source software; delivered and maintained by the IT staff of the business using the solution. The thesis explores the advantages and disadvantages of this approach and

evaluates whether and under which circumstances would this approach be viable. The thesis does not address the issue of one particular business, the resulting offsite backup solution can be implemented to an arbitrary organisation using chapter 5 as a guide. The scope of the thesis is additionally limited to addressing the aspects of secure data transfer and configuration of required software and does not address aspects such as redundant failsafe storage, because the solution works regardless of the storage media used. The solution was designed and implemented in a test environment using the author's existing hardware.

1.3 Security

As with any IT solution, security is of upmost importance also regarding data backups. This thesis considers the three traditional pillars of information security, which are availability, integrity, and confidentiality [2].

An offsite backup solution directly improves availability by providing a second copy of the data, which is available even in the event of a disaster at the main location, including but not limited to theft, fire or structural damage to the building. Availability can be increased further by increasing the amount of backup copies that are stored. The decision of how many backup sites to use is up to each individual business, but the 3-2-1 rule is recommended to be followed at a minimum. Costs are associated with the addition of each additional backup location and chapter 2.2.1 explains the viability in more detail, however the solution proposed in this thesis is flexible regarding the number of backup sites.

Additionally, an offsite backup improves integrity by ensuring that the data can be restored if it's been tampered with in cases such as ransomware infections, business sabotage or accidental deletion. Integrity of the backup solution itself was also considered and there is software that ensures the consistency of the backups by signing them with a digital signature. The implementation of RAID is recommended for a server storing backups to ensure that backups aren't lost due to a hard drive failure, however it is not a requirement for the proposed solution to function. Chapter 4.2 explains the software choice and 5.3.2 includes information on the verification of the integrity of the backup copies.

In contrast to availability and integrity, backup copies on their own may reduce confidentiality, as each additional copy of the data is potential attack vector. However, measures can be put in place to mitigate this risk. For example, if the source data has granular permissions set on directories and files to only grant access to those that need it, when this data is backed up, the backup archive itself does not have these permissions. Therefore, the backup archive should be encrypted to ensure that unauthorised users cannot read the data inside. In addition to encryption at rest, data is to be encrypted in transit as well, to prevent man-in-the-middle and other interception attacks. The backup archive itself should have file permissions set per the principle of least privilege, to ensure that it can only be accessed by user accounts associated with the backup process (for instance, administrator and service accounts) and to prevent unauthorised tampering or deletion [3]. The thesis explains the measures mentioned above in more detail in chapters 3.2, 5.3.1 and 5.4.1.

2 Overview of data backup solutions

Several commercial providers offer backup solutions to protect business data. The thesis proposes a solution as an alternative to these. Subchapter 2.1 provides an overview of cloud-based solutions and 2.2 elaborates the essential components of an on-premises solution, which this thesis proposes.

2.1 Cloud-based solutions

Large technology corporations such as Google and Microsoft offer solutions to store data in their datacentre infrastructure, also referred to as "the cloud". For instance, Google offers Google Workspace [4] and Microsoft offers Microsoft 365 for Business [5], both also include a suite of productivity tools. These cloud storage solutions however do not provide protection against accidental deletion, ransomware infection, or any other case where unauthorised tampering of data had taken place.

For that, there are advanced backup solutions. One such solution is Microsoft 365 Backup and Recovery by Veeam, which is a BaaS solution to back up user data stored in Microsoft 365, such as OneDrive, Exchange Online mailboxes, or SharePoint. This would allow data to be backed up to an on-premises server (discussed in chapter 2.2), or to another cloud service provider. The backup software is provided for a monthly fee and is billed on a 1–5-year plan, however pricing for BaaS solutions which include storage is only available through sales partners. For example, Telia, a Veeam Gold partner in Estonia, lists a monthly licence fee of €1.86 plus €0.12 for each GB backed up. [6] [7]

Another business-oriented backup solution is Backblaze Business Backup. Backblaze hosts their own storage, and it is included in the cost of their service. Software is provided for backing up workstations and servers and integration with Veeam is also provided. The service is billed per terabyte or per workstation. [8]

2.2 On-premises solutions

A business may opt to purchase their own storage media and use that for storing backups. The media may be a hard drive, optical disc, or a NAS device. The benefit of this is greater control and privacy since the end user can configure and monitor their chosen solution as they intend without relying on a cloud provider. The disadvantage is greater upfront cost since the business would have to purchase their own media. However, a business may already have decommissioned server or workstation available, which can be reused as a backup server. This approach significantly reduces upfront costs and environmental impact and this type of solution is the focus of the thesis.

If the backup media is stored in the same physical location as other copies of the data, then this backup does not follow the 3-2-1 rule. A backup copy that is in a separate physical location is defined as an offsite backup and is required for following the 3-2-1 rule and for ensuring data availability and integrity in the event of a disaster at the main location, including but not limited to fire, building collapse or theft. [1] As the data will be stored offsite, it would have to be transferred over a network, therefore the backup hardware would require a network connection. Thus, an external hard drive for example on its own is not suitable for an offsite backup, however a server with the required storage space would be sufficient. The details how the components for this solution are chosen are described in chapters 3 and 4.

2.2.1 Viability

As on-premises server solutions require physical space for servers along with electricity and networking, they are best suited for business who already have such facilities available. As the aim is to create an offsite backup, the organisation would also need a second location to store the offsite backup server. Therefore, the solution described in this thesis is most optimal for a small business with at least 2 locations, such as offices, with appropriate accommodations available for servers. The solution assumes the presence of at least one on-premises server already, which would the store the data which is to be backed up.

A business with no office or whose office lacks accommodations for servers should consider cloud storage solutions instead, supplemented with a backup solution such as Veeam Microsoft 365 Backup. A business with a single office and an existing server but

without a second location may use a commercial server backup solution which includes storage in its pricing, such as Backblaze.

3 Backup system parameters

Several parameters are to be taken into consideration when designing a backup solution. These parameters will form the selection criteria for the software and other components needed for the backup solution. Each subchapter describes the formation of the selection criteria and making the appropriate decisions for the backup system.

3.1 Capacity

The capacity of this backup system is for the business to choose based on the size of their data. However, to mitigate against data corruption and ensure integrity and availability of data, enough capacity should be provided to hold three full backups of data concurrently, as well as incremental backups. A full backup contains a copy of all data which must be backed up, whereas an incremental backup contains only the changes made since the last full backup. Should a backup copy become corrupted, all incremental backups based on it become unusable as well. The author as the creator of the backup system recommends two full backups to be always stored as this decreases the chances of a restore process failing due to corrupt data: if one full backup is corrupted, data can be restored from another one. Additionally, a new full backup would be performed before an old one is deleted, so until the old full backup is deleted, there would have to be storage for three full backups.

The test environment contains approximately 180 GB of data to be backed up, therefore at least 540 GB of storage space is needed for three full backups. To account for incremental backups and the addition of more data, a 1 TB drive was deemed most suitable.

3.2 Security

As stated in chapter 1.3, the offsite data backup system itself must be designed so that its availability, integrity, and confidentiality are ensured. For confidentiality, measures are put in place to ensure that backup copies are not accessible to unauthorized users.

Therefore, the backup copies are to be encrypted and the chosen backup software must provide such functionality. Encryption of the backup is necessary even if the original data is not, because file permissions contained in the source data do not apply to the backup copy, which generally takes the form of a single directory. Without encryption, any user with access to this directory can restore the files and bypass the permissions on the original files. In addition to that, file system permissions on the backup copy itself will be set up so that only the user accounts necessary for the system to function can access the backup data, using the principle of least privilege [3]. This is to prevent unauthorised or accidental deletion or modification. Encryption keys for the backup copies are to be stored separately in a key vault. Additionally, when planning the physical accommodations for the backup servers, the same or equivalent security measures should be in place as the main server. For example, if the main server is in a secure server room with access granted only to administrators, the offsite server should have the same level of physical access restrictions.

Integrity of backup copies is ensured by signing the backup copy digitally, storing multiple copies of the backup and if deemed necessary checking the files for corruption. Should corruption be detected, a full backup is to be made. The details of how corruption is detected, and encryption is set up are described in chapter 5.3.2. Furthermore, to ensure integrity and availability in case of a hard drive failure in the backup server, it is assumed that RAID is used, however the exact RAID configuration is out of scope of the thesis as its configuration and hardware may vary depending on available hardware and business needs. To ensure availability, the process of restoring data from a backup copy is to be tested at least once every 6 months [9].

An important note regarding encryption keys is that they should be stored in a separate location and the backup system itself cannot be relied on to keep a copy of the keys. If the first copy of the key were to be lost, and the only other copy is stored in the same backup, then it becomes impossible to restore both the key and the rest of the data, because the key needed to decrypt the backup containing the key is stored in the encrypted backup itself. This key should also not be stored in plaintext and should ideally be stored in a hardware security module or other key storage vault. Furthermore, the key storage medium should grant access to the key only to those who need it, such as administrators responsible for the backup. There are several open-source solutions that provide the needed functionality, such as Bitwarden, KeePass and Passbolt. The author already uses

Bitwarden and this was therefore chosen to store the encryption keys in the test environment. When deployed to a business which already uses a password manager or other encryption key vault, it may be used to store the necessary keys [10].

3.3 Scalability

A backup system needs to be able to accommodate more data as business needs increase. However, the capacity of the backup system is limited by the amount of storage available on the backup server. On-premises IT solutions in general have inferior scalability than cloud-based solutions because capacity is limited by physical hard disk space available, whereas with a cloud-based solution storage can easily be increased [11]. Additionally, for on-premises storage with a single hard disk, the data would have to be manually migrated to a new hard disk if more storage is needed. This problem can be solved with certain types of RAID, where additional disks can be added without needing to migrate data. In addition to that, the backup system is flexible with regards to the number of destination servers: additional servers can be added as needed to store the data.

In the test environment, 1 TB was initially deemed sufficient even when accounting for growth in the amount of data. This was a limitation of the hardware available to the author. Once this becomes insufficient, additional physical hard drives will have to be purchased and the backup copies need to be migrated. However, TalTech IT College was able to provide an additional server with 4 TB of storage as a backup destination, which features hardware RAID capabilities.

3.4 Connectivity

A core component of the backup solution is the transfer of data offsite to a separate physical location. Both sites therefore need a reliable internet connection. Additionally, to ensure that data remains secure during transit, encrypted connections would be used. This means that for example FTP on its own is unsuitable, however FTPS is suitable. Even if the data were transmitted over an encrypted VPN, the data transfer protocol itself should be encrypted regardless, to protect against threat actors inside the network and in addition to maintain another layer of security in the absence of a VPN, should it fail.

Additionally, both sites have a network firewall in place which denies all inbound traffic by default and the servers are behind a layer of NAT. As inbound traffic is necessary for the backup system to function, one possibility of facilitating connectivity between two sites is to open the ports on the network firewall at either site that are necessary for the protocol of choice to function. This poses a security risk: an out of date or improperly configured service listening on that port can be exploited by a threat actor [12]. Access control lists would restrict access to needed ports only to those IP addresses that are needed, but this requires changes every time that the IP address at either site changes, therefore this is unsuitable in the test environment where external IP addresses are dynamic. The most suitable solution to this problem is to use a VPN, which connects the offsite network directly to the main network, allowing data to be transferred through between the two.

3.4.1 VPN

The two main types of VPNs are remote access and site-to-site. A remote access VPN connects a single computer to a network, whereas a site-to-site VPN connects two networks together in their entirety. A site-to-site VPN provides a more seamless connection, since the routing would be done at both sites by dedicated VPN gateways, however a remote access VPN is more flexible, because the client software can run on the offsite backup server itself. While it is possible to use the servers themselves to also host VPN gateways for a site-to-site VPN, this raises complications. The author has tested this configuration previously using the open-source VPN server software provided by SoftEther and discovered that due to a limitation of Linux and other UNIX OS-s, VPN clients cannot access resources on the server hosting the VPN. A workaround involves using a separate NIC. If additional hardware was available for a VPN gateway, a site-to-site VPN would be preferred. Should an organisation have a site-to-site VPN already in place, this can be used for transferring the data. [13] [14]

The test environment has the hardware for a dedicated VPN gateway only at the main network, whereas a business may have the required hardware at both. Therefore, a remote access VPN was set up, with the VPN gateway being hosted on the router at the main network and the offsite server would host the VPN client. Alternative measures of providing connectivity may be implemented for when the VPN fails, depending on the

business needs. For instance, in the test environment, the SSH port can be forwarded on the router via management tools provided by the ISP.

3.5 Manageability

After initial setup has been completed, the backup system should function automatically with minimal intervention by an administrator. For administration tasks, SSH will be used to access the storage server and make configuration changes. The router will generally be managed via its web GUI. SSH password authentication is to be disabled to reduce the risk of brute-force attacks. The private keys used for authentication are only generated for administrator(s). In the test environment, this only includes the author. The SSH server will listen on a non-default port, which will be closed on the firewalls of both sites. Alerting was set up for the VPN connection that backups complete without errors using scripts written by the author, shown in appendix 3 and explained in chapter 5.6 Any alerts will be sent to an administrator by email.

3.6 Performance

The performance of this backup system varies greatly depending on the hardware and network infrastructure in place at the business where it is deployed and therefore the author cannot make a reliable general assessment of it. However, the author has made observations in the test environment and noted that compute performance of the servers plays a significant role. While the network bandwidth in the test environment was 300 Mbit/s, the time needed to compress and encrypt the backup on the main server reduced overall throughput greatly. Additionally, the remote access VPN adds additional overhead because the VPN clients on the offsite servers decrypt all incoming data from the main server. When the author deployed the second offsite server, which had significantly more processing power than the first, the time to complete a backup shrunk by nearly half.

4 Selection of software

Based on parameters set out in chapter 3, software was chosen to perform the tasks needed in the backup system. As set out in chapter 1.2, all software is to be free and open source. Each subchapter compares the available options for each role and selects the most suitable option.

4.1 Server operating system

Both storage servers would run a Linux distribution since these are the most widely available open-source operating systems. Linux distributions designed for server use include Debian, Ubuntu (based on the former), as well as RHEL and its derivatives CentOS and Fedora. RHEL is paid software, but CentOS and Fedora are not. Fedora implements changes at a rapid pace and new releases are made roughly every 6 months [15]. CentOS is described as a midstream release tracking ahead of RHEL on a rolling release model [16]. Debian has an infrequent release schedule but is considered more stable [17]. Ubuntu is based on the "unstable" version of Debian currently in development, but releases versions with long term support every 2 years based on "testing" builds of Debian [18].

Ubuntu was chosen for deployment in the test environment because of its long-term support while also having later versions of software packages available from its repositories than Debian does. In March 2022, when the backup system was deployed, the latest release with long term support was 20.04, released in April 2020 and supported until April 2025 [19]. When additional hardware became available from IT College, the latest release with long term support was 22.04, but the author decided to install 22.10 on it instead, which was only supported for 18 months from its release. This decision was made to ensure that no changes had been made to Ubuntu or the Linux kernel which would render the VPN client unsuitable, and since the server was in use only for a few months in 2023 while this thesis was being written, end of support wasn't an issue. The author has also tested the VPN client in Ubuntu 22.04.

4.2 Backup software

The criteria for the software used for the backup solution is that it must be free and open source, support multiple versions of backups, full and incremental backup functionality, digital signatures of backups and encryption of backup files. There are several such tools available from commercial providers like Acronis or Veeam, but they are not open source [20] [21]. There are two tools that fit the criteria: Duplicity and Borg [22] [23].

Both Borg and Duplicity support versioning and encryption of backups, however there are differences. Borg only maintains one full backup and all further backups are incremental. To make another full backup, the old backup must first be pruned, or deleted. As the backup system is intended to store two full backups concurrently, Duplicity is much more suitable, because it allows the user to specify whether a full or incremental backup is required. However, Borg has advantages as well, such as the ability to mount a backup copy as a file system and browse it, without having to perform the full restore process. Borg's other advantage is space efficiency, using a deduplication method to reduce the amount of storage needed. Ultimately, in this backup system, integrity of data was deemed more important than space efficiency and Duplicity is better suited for this [22] [23].

While Borg only supports backups to remote hosts over SSH, Duplicity provides a variety of options for underlying data transfer protocols; the protocols relevant to the thesis are FTPS, SFTP and SCP. FTPS is the same protocol as the commonly used FTP but with encryption, which is necessary as specified in chapter 3.2 [24]. SFTP and SCP use SSH to transfer the data and therefore use the same encryption and authentication methods. For FTPS, just as with FTP, server software would have to be installed and configured on the offsite server, whereas SFTP and SCP can use the existing SSH server on the offsite server [25]. SFTP also provides interactivity and more functionality as part of that, however SCP has slightly reduced latency [25]. Both SCP and SFTP were tested, and it was found that SCP in fact takes several more hours to complete a backup than SFTP, so SFTP was chosen. Previously the author had seen that FTPS can be marginally faster, but as it requires separate server software along with user authentication to be configured and maintained, it was not deemed necessary and SFTP was expected to provide a balance between performance and simplicity.

4.3 Router operating system

As the router deployed in the backup system will use standard PC hardware, it also requires an operating system. A Linux distribution would work, but it would require significantly more configuration and be harder to manage than an OS designed for router use. Three examples of such operating systems are OpenWrt, pfSense and OPNSense. OpenWrt was mainly designed as a custom firmware replacement for common home internet gateway devices, but it can run on standard PC hardware as well [26]. However, configuring a VPN Server on OpenWrt requires the installation of a software package and configuring it via command line, and the author decided to use an OS that allows for VPN configuration through a GUI to reduce complexity.

pfSense and OPNSense include more features than OpenWrt, such as a built-in OpenVPN Server, and share a similar code base, as OPNSense was forked from pfSense. OPNSense features a newer user interface and tends to add features before pfSense, but pfSense has existed for longer and therefore has more documentation and community support available. pfSense was chosen because of better support, but the author would consider OPNSense viable as well. [27]

4.4 VPN software

pfSense includes the software for 3 types of VPN servers: OpenVPN, L2TP and IPSec. L2TP provides no encryption on its own, therefore L2TP alone is not suitable for this backup system, because data encryption while in transit is required [28]. IPSec and OpenVPN are both considered cryptographically secure and pfSense includes a vast number of options for configuring both [29]. The author decided in favour of a remote access VPN in chapter 3.4.2, and this was taken into consideration as well.

Since the offsite server is in a separate physical location, it needs to connect to the main network without any intervention by an administrator each time the OS boots. Therefore, the VPN client software would need to have this feature available. OpenVPN includes this functionality in its OpenVPN3 client which can be configured with an autoload file [30]. The feature had been tested by other users of OpenVPN and solutions to the problems I encountered were already available. IPSec did not have any documentation of

such feature and since it provides the same functionality as OpenVPN, there was no advantage to IPSec over OpenVPN.

5 Implementation

The backup system was implemented in a test environment, which uses the author's own hardware, the software selected in chapter 4 and was tested with the author's personal data. This chapter is intended to serve as a guide for a system administrator in a small business to implement the author's solution.

5.1 Hardware

The solution has been implemented using three physical servers. One server (the main or primary server) is in the author's home, the other in a secondary location in another city. The third server was provided later by IT College and is within their premises.

The main server has an Intel i7 4th generation processor, a 256 GB SSD boot drive for an operating system and a 3 TB hard disk drive for storage. Using separate drives provides flexibility in the case of a hardware failure: the data drive can be replaced without reinstalling the OS, and a failed OS drive can be replaced without having to restore data from a backup.

The hardware at the first offsite server is similar: an Intel i3 1st generation processor, 128GB SSD boot drive and a 1TB HDD for storing the backups itself.

While the thesis was being written, hardware was made available from the IT College for the purpose of storing additional backups. This server is a Fujitsu Primergy RX2530 M2 and features an Intel Xeon E5 v4 series processor and 4TB of HDD storage. The storage was in the form of 2 HDDs, 4 TB each, arranged in RAID1.

The author acquired the 256 GB SSD and the 3 TB HDD for the purpose of this project. The author already owned all other hardware.

5.2 Network

All sites feature an Ethernet network to provide internet connectivity to servers. In general, specific configuration is only needed for the main network and its router to

provide the VPN server. Because the VPN in the test environment is of a remote access type, the address spaces in the networks may overlap, however for the purposes of simplicity they do not.

5.2.1 Offsite network

The network at the first offsite location in Tartu has a dynamic public IP, and the private network uses the 192.168.1.0/24 address range. The server's private IPv4 address is static 192.168.1.2. The network in IT College, where the second backup server is located, uses the 192.168.161.0/24 address range and the address 192.168.161.127 was reserved for the backup server.

5.2.2 Main network

The network at the primary server is more complex. As specified in chapter 3.4.2, it was decided to host the VPN server on the pfSense router set up for this network.

The pfSense router uses an AMD A10 7850K processor and a 500 GB HDD boot drive. For networking an Intel network interface card was bought with 2 Gigabit Ethernet ports: one of which would connect to the internet and the other to the local network. The local network used an address range of 172.20.0.0/24.

5.2.3 VPN Server

pfSense includes OpenVPN server software. The server was configured using the wizard provided by pfSense and following their documentation [31]. User authentication was set up requiring both a username and password combination as well as a client certificate. The usernames and passwords would be authenticated via the local user database on the router and for client certificates, a new certificate authority was created and from that a new server certificate and client certificates for all users. OpenVPN was configured to operate in "tun" mode, meaning that clients would connect to a separate tunnel network rather than directly to the local network. The "tun" mode is recommended by pfSense because it is more compatible with different VPN clients and when the author tested the alternative "tap" mode, which would connect the client directly without a tunnel network, issues arose regarding IP address assignment and the clients could not establish connectivity. The tunnel network used an address range of 172.29.0.0/24, this can be any available address space. For clients to access the backup server, a route was added to the

local network from the tunnel network using the IPv4 local network setting. The required tunnel settings are shown in figure 1 and the resulting network in figure 2. Finally, to prevent the pfSense firewall from blocking traffic, a rule was created to allow traffic from any source to connect to port 1194. Other settings not mentioned in this paragraph were left default.

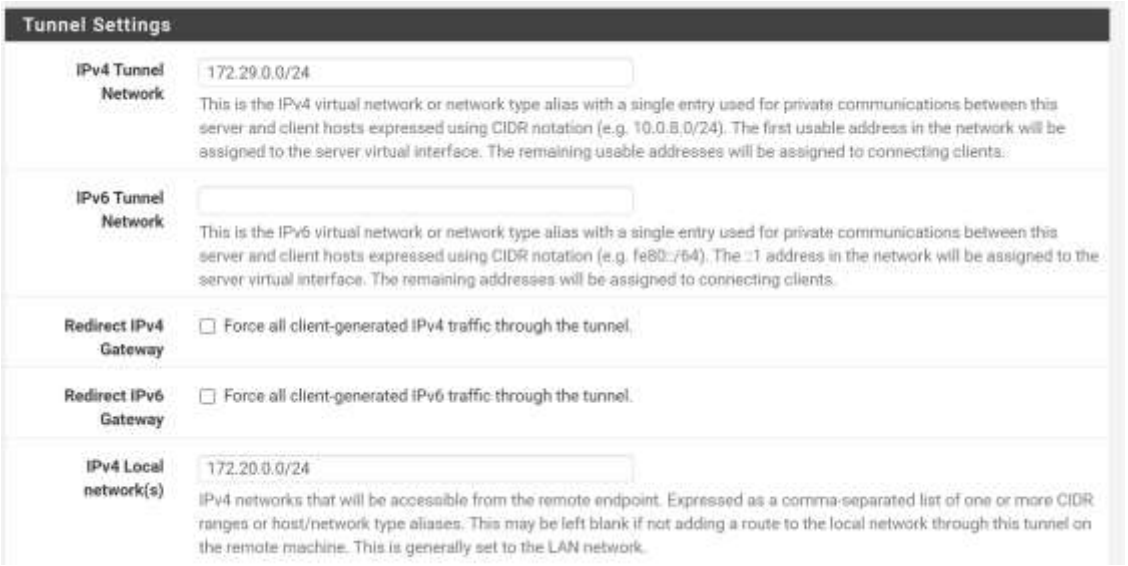

Figure 1. OpenVPN tunnel settings

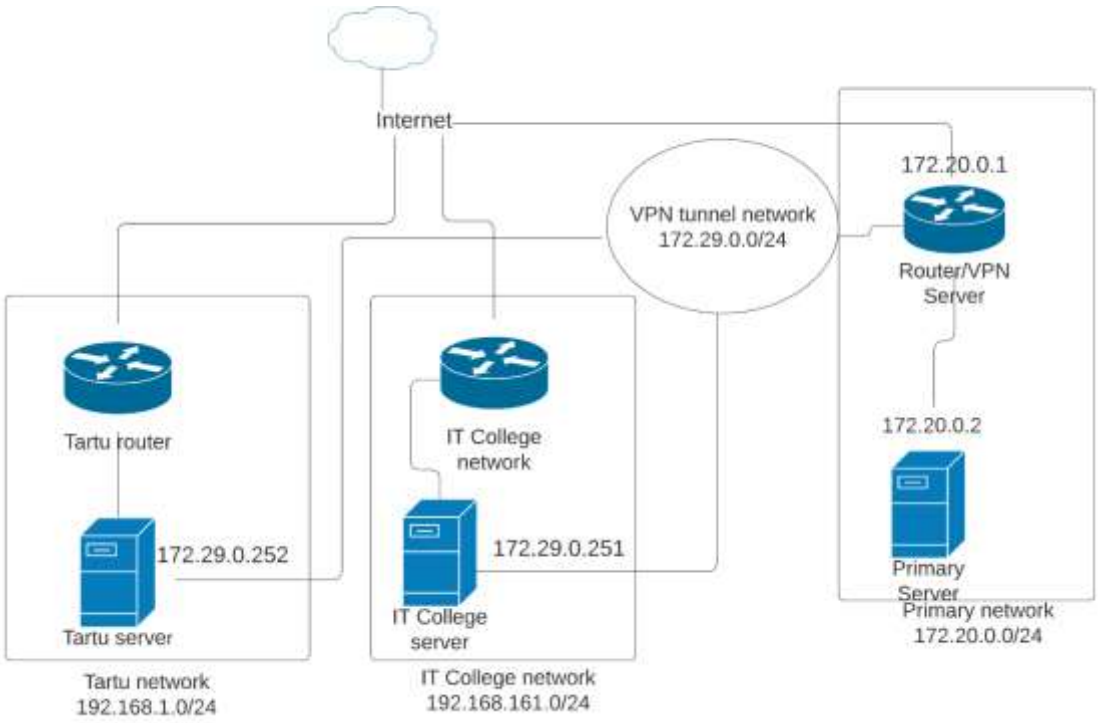

Figure 2. Network diagram including main and offsite networks.

5.3 Main server

The main server is in the author's home and was in use before the thesis was composed. The author has used this server for the storage of personal data and until the implementation of the backup solution this thesis proposes, there was no offsite backup of this data. This chapter explains the necessary steps to configure the main server, although the author had done most of these beforehand.

5.3.1 OS installation and initial configuration

Ubuntu Server 20.04 was installed on the SSD on the main server and patched with the latest updates to software packages and the Linux kernel. Initially configuration was done directly with a keyboard and mouse connected to the machine, later an SSH connection was used from a remote machine. A static IP address of 172.20.0.2 was configured on the server during the install process. For the SSH connection, public key authentication was used: a private key was generated on the author's computer and the public key copied to the list of authorized keys on the server, at ~/.ssh/authorized_keys. Additionally, password authentication was disabled, and the listening port was changed. This was achieved by editing the SSH configuration file at /etc/ssh/sshd_config as shown in figure 3. The hard disk was formatted to ext4 and mounted at /mnt/storage using an entry in /etc/fstab [32]. Two directories were created on the disk: media which would contain user data and archive, which would contain files to be backed up. Two user accounts were created: one during the initial setup process which belongs to the sudoers group, another without these permissions which would perform the backup. In the test environment, these usernames were sandy-bridge and backup_script. Both users were added to a new security group named media.

```
PasswordAuthentication no
Port 14503
```

Figure 3. SSH Configuration lines to disable password authentication and set the port.

Permissions for the relevant directories were assigned following the principle of least privilege. The media directory was owned by the sandy-bridge user account with full read, write and execute permissions, and the archive directory was owned by backup_script. The media group had read and execute access to the directories and the group identity permissions bit was applied as well, which applied the group ownership to

newly created contents of the directories. Other accounts had no access to either directory. The commands to achieve this result are shown in figure 4.

```
chown -R sandy-bridge:media /mnt/storage/media
chown backup_script:media /mnt/storage/archive
chmod 750 /mnt/storage/media
chmod 750 /mnt/storage/archive
chmod g+s /mnt/storage/media
chmod g+s /mnt/storage/archive
```

Figure 4. Commands to set permissions on the directories on the main server.

The media directory was mainly accessed by a network share provided by Samba, so to ensure that files added through Samba had the correct permissions, masks were set in the /etc/samba/smb.conf configuration file. The network share was defined as shown in figure 5. Newly created files can be read and written by the owner, read my members of the media group and other users have no permissions. Directories have the execute permission added as well, as this allows the owner and the group to enter the directories.

```
[Media]
  comment = Media share
  path = /mnt/storage/media
  read only = no
  guest ok = no
  create mask = 0640
  directory mask = 0750
  browsable = yes
```

Figure 5. Samba network share definition

5.3.2 Installation and testing of Duplicity

Duplicity was installed from Ubuntu's package repository. The backup process was initially tested by backing up the data to the server's local storage. The source directory would be /mnt/storage/archive, which would contain files themselves as well as symbolic links to directories to be backed up. Symbolic links can be added or removed from the directory as needed, depending on what data needs to be backed up. Initially, the backup process was tested without encryption as shown in figure 6.

```
duplicity --no-encryption /mnt/storage/archive
sftp://backup_script@172.29.0.252:14503//mnt/storage/tallinn-backup --
copy-links
```

Figure 6. Command to run a single Duplicity backup without encryption

When the initial backups completed successfully, GPG encryption was configured for Duplicity. For this, a GPG key was generated and secured with a passphrase, which was

stored in the author's key storage vault. The key must be generated by the user account which would run the backup. The generated key ID was then used for Duplicity to encrypt the backups. The command to generate a key and run an encrypted backup are shown in figures 7 and 8. An additional command to verify integrity of the backup is shown in figure 9.

```
gpg –-gen-key
```

Figure 7. Command for generating a GPG encryption key

```
PASSPHRASE=”key password” duplicity --encrypt-sign-key="$GPG_KEY"
/mnt/storage/archive sftp://172.29.0.252:14503//mnt/storage/tallinn-
backup --copy-links
```

Figure 8. Command for running an encrypted backup.

```
PASSPHRASE=”key password” duplicity verify --encrypt-sign-key="$GPG_KEY"
sftp://172.29.0.252:14503//mnt/storage/tallinn-backup /mnt/storage/archive
```

Figure 9. Command to verify integrity.

5.4 Offsite server

Two offsite servers are used for the test environment. Minimal configuration is required on these servers compared to the main server, most of which relates to the VPN client.

5.4.1 File permissions

While the main server and network are at the author's home in Tallinn, the first offsite server is in Tartu. While the thesis was being written, hardware became available from TalTech IT College for use in the thesis. The initial steps were the same for the offsite servers as described in chapter 5.3.1, with a few differences. The offsite network in Tartu uses the 192.168.1.0/24 address range, so the address was set to 192.168.1.2. In IT College, the assigned address was 192.168.161.127 in the 192.168.161.0/24 subnet. Backups were stored in /mnt/storage/tallinn-backup, permissions were set as shown in figure 10.

```
chown -R backup_script:media /mnt/storage/tallinn-backup
chmod 750 /mnt/storage/tallinn-backup
chmod g+s /mnt/storage/tallinn-backup
```

Figure 10. Commands to set permissions on the directories on the offsite server.

As files would be transferred over SFTP, permissions had to be set for newly created files as well. For this, the umask was set for the SFTP subsystem in SSH. The umask defines a value which is subtracted from the default octal value for file permissions. For example,

the default in Linux for files is 666 and directories 777. The umask was set to 027, so that newly created files have permissions set to 640 and directories to 750. To set the umask value for SFTP, it was defined in the SSH configuration file at /etc/ssh/sshd_config as shown in figure 8. Note that only one line starting with Subsystem SFTP may exist in the file – if one already exists, it must be modified as shown in figure 11 [33].

```
Subsystem sftp /usr/lib/openssh/sftp-server -u 027
```

Figure 11. SFTP subsystem definition with the correct umask value

For the main server to access the offsite server, SSH access was needed for both file transfer and administration tasks, so an additional SSH private key was generated for the main server and the public key added to the list of authorised keys on the offsite server's user accounts. As Duplicity would transfer data over SFTP, this same key pair would also be used for authenticating and encrypting the data transfer process.

5.4.2 VPN client

The OpenVPN3 client was installed as per the documentation by OpenVPN [34]. As the VPN client had to launch and connect automatically during system startup, the connection was configured using the autoload feature. OpenVPN3 can establish a connection without superuser privileges, so the backup_script user was used for starting the autoload service.

For this, a directory was created in the user's home folder, which stored the standard OpenVPN configuration profile as well as the autoload configuration file The directory would be .openvpn3/autoload and be readable and writable only by the backup_script account. The profile was exported from the VPN server on the main network router and the autoload configuration was made using documentation from OpenVPN, an example of the latter is shown in figure 12. [35]. The files must be named client.conf and client.autoload respectively. Finally, a systemd daemon file was created, which was used to launch OpenVPN3 on boot. An example is already included with the OpenVPN3 package in /usr/lib/system/system/openvpn3-autoload.service, this was configured to run using the backup_script user and to use the configuration stored in its home directory, an example is in appendix 2. It should especially be noted that the default directory in the autoload daemon - /etc/openvpn3/autoload – should not be used for configuration files, as this directory was once deleted during the OpenVPN3 package update process in the test environment. To ensure that the client receives the same IP address each time, it must be set using client specific overrides on the VPN server. For the offsite server in Tartu to

have an IP address of 172.29.0.252/24, the line in figure 13 was added to the Advanced section of client specific overrides for the VPN user account. For the server in IT College, the steps were repeated with a new VPN user account and an IP address of 172.29.0.251/24.

```
{
  "autostart": true,
  "user-auth" : {
    "autologin": true,
    "username": "taltechvpn",
    "password": "password redacted"
  },
  "tunnel": {
    "persist": true
  }
}
```

Figure 12. OpenVPN3 client autoload file

```
ifconfig-push 172.29.0.252 255.255.255.0;
```

Figure 13. Client specific override to set a static IP address.

5.5 Scripting and automation

After the completion of Duplicity initial testing, the next step was to write a shell script that would perform the backup as well as supporting scripts to monitor the process and send alerts. Details on the supporting scripts are provided in chapter 5.6. The backup script had several purposes: first to make a copy of the files to be backed up in the correct location, second to initiate Duplicity's backup process and third to clean up (delete) old backups. The script to run a backup is shown in appendix 3 and its configuration file in appendix 5.

As the main server also hosts a WordPress web server, its media and database are also included in the backup, but the configuration of this is out of scope for the thesis. The backup script performs a full backup to both servers if the latest full backup is older than 7 days, otherwise it performs an incremental backup. If the backup fails or returns any error, it is written to a log file and a copy is sent to the administrator by email. Finally, old backups are cleaned up, keeping 2 full backups in place. This script runs daily at 01:00, configured using cron.

The scripts read several parameters from a configuration file, shown in appendix 5, such as the source and destination directories, addresses of the backup servers, and the e-mail address of the administrator. The GPG key ID and passphrase are stored in a separate file, .backup.keys, which is only readable and modifiable by backup_script.

5.6 Alerting

To ensure that an administrator is aware of downtime and failures, basic alerting was configured for the VPN connection and the backup process. The VPN was monitored using a script to ping the offsite servers every 5 minutes, with an email alert sent to the administrator should all 4 out of 4 consecutive pings fail. For sending mail, the mailutils package is required. The script is shown in appendix 6. The email is only sent the first time that the attempt fails, for subsequent failures no email is sent. A notification is sent if the server comes back online.

The script in appendix 3 writes errors to a log file. An additional script, shown in appendix 4, runs hourly and checks these log files for errors. It retries the backup and clears the errors if the backup succeeds. If after 1 hour, the backup still fails, an email is sent to the administrator. The script keeps track of the emails it has sent and only sends one email per host during failure.

5.7 Test of the restoration process

The restore process was tested on the 13th of April 2023, starting at 16:48. The /mnt/storage/media directory was deleted, as well as the root directory of the WordPress web server container. For the restore process, an account in the sudoers group was used. The commands in figure 14 executed the restoration. To improve performance, the data was restored from the server in IT College, since backups to that server had completed in a shorter amount of time.

```
. /home/sandy-bridge/.backup.keys
duplicity --encrypt-sign-key $GPG_KEY
sftp://172.29.0.251//mnt/storage/backup
/mnt/storage/restore
```

Figure 14. Commands used during the tested restore process.

The process was started at 16:56. The exact completion time was not recorded, because the console window showed hundreds of lines of errors by the time the process had

completed. However, the errors were a minor issue and the files had successfully been restored. Because the process was run from a non-root user account, Duplicity was unable to set permissions on the files which had been restored. The solution is to run Duplicity as root using sudo, passing the SSH private key for authentication as an argument, since the root user does not have its own private key.

Finally, to complete the restoration process, the directories were moved from /mnt/storage/restore to an empty, recreated /mnt/storage/media. Not all directories in the original media directory were backed up, but everything that was backed up was successfully restored. The permissions on the directory were set manually as per chapter 5.3.1. The WordPress web server was also restored, by creating a directory with the path of /opt/wordpress/html and copying the restored wp-content directory there. Then, the containers for both the web server and the database were recreated, however the configuration file used by Docker was not included in the backup. When the containers were running, the database was restored from a dump file, following a guide [36]. To complete the process, execute permissions were set on the /opt/wordpress/html directory and all its contents for the www-data user, this is necessary for the web server to serve these files. By 20:01, the data had been restored.

Overall, the test of the restoration process was successful. The only two issues were that the configuration file for Docker wasn't backed up and that permissions had to be set manually. The author had beforehand intended for configuration to be documented separately, but as that was absent at the time of writing, it should have been included in the backup and will be in future backups. Permissions will not be an issue in future restorations, because Duplicity includes them in the backup, and they will be restored if the restore process is run as root. All data that was backed up was available after a restore. For future data restorations, the commands should be used as shown in figure 15.

```
. /home/sandy-bridge/.backup.keys
sudo duplicity --encrypt-sign-key $GPG_KEY
sftp://backup_script@172.29.0.251//mnt/storage/backup
/mnt/storage/restore --ssh-options="-
oIdentityFile='/home/backup_script/.ssh/id_ecdsa'"
```

Figure 15. Improved commands for restoring from a backup.

6 Summary

The proposed solution described in this thesis provides an offsite backup to a small business of up to 50 employees and therefore improves the integrity and security of their data. The prototype solution was implemented and tested successfully in a test environment.

The analytical part of the thesis (chapters 3 and 4) establishes the required parameters for the system and compares software that can meet these parameters. The software required for the backup system to function is free and open source. Furthermore, availability, integrity, and confidentiality were considered during the process to ensure security, which is of paramount importance.

Chapter 5 of the thesis serves as a guide on how to deploy the system in an organisation and the system is flexible to be useful in a variety of organisations. Furthermore, as decommissioned hardware can be repurposed for this backup system, which the author has done in the test environment, it reduces cost and environmental impact for the business.

The restore process was tested successfully. All data which was backed up was restored. The backup system is automated and administrator intervention is not required under normal operation. An administrator is notified in case of errors during the backup process. As the backup is stored offsite, it assists in complying with the 3-2-1 rule. The proposed solution improves information security of an organisation using it by providing a copy of business data in a separate physical location, thereby ensuring availability and integrity in case of a disaster. The author recommends implementation of this backup solution in a small business with at least 2 locations and at least one on-premises file server.

7 References

[1] U.S. Chamber of Commerce, “What Is the 3-2-1 Backup Rule?,” 16 October 2021. [Online]. Available: https://www.uschamber.com/co/run/technology/3-2-1-backup-rule#:~:text=Here's%20what%20the%203%2D2,least%20one%20backup%20file%20offsite.. [Accessed 14 February 2023].

[2] W. Chai, “What is the CIA triad (confidentiality, integrity and availability)?,” 1 February 2023. [Online]. Available: https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA. [Accessed 5 May 2023].

[3] National Institute of Standards and Technology, “Least Privilege,” in *Security and Privacy Controls for Information Systems and Organizations*, Gaithesburg, National Institute of Standards and Technology, 2020, p. 36.

[4] Google, “Google Workspace,” 5 February 2023. [Online]. Available: https://workspace.google.com/. [Accessed 5 February 2023].

[5] Microsoft, “Microsoft 365 for Business,” 5 February 2023. [Online]. Available: https://www.microsoft.com/en-ww/microsoft-365/business. [Accessed 5 February 2023].

[6] Veeam Software, “Backup as a Service for Microsoft 365,” 17 January 2023. [Online]. Available: https://www.veeam.com/backup-as-a-service-for-microsoft-365.html?ad=menu-solutions. [Accessed 17 January 2023].

[7] Telia Eesti, „Väljavõte Telia Eesti AS lõppkasutajate hinnakirjast,“ 22 December 2022. [Võrgumaterjal]. Available: väljavõte Telia Eesti AS lõppkasutajate hinnakirjast. [Kasutatud 31 January 2023].

[8] Backblaze, “Business Backup,” 17 January 2023. [Online]. Available: https://www.backblaze.com/business-backup.html. [Accessed 17 January 2023].

[9] P. Kirvan, “What is a good backup test frequency?,” 22 April 2019. [Online]. Available: https://www.techtarget.com/searchdatabackup/answer/What-is-a-good-backup-test-frequency. [Accessed 14 April 2023].

[10] OWASP, “Key Management Cheat Sheet,” 17 April 2023. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html. [Accessed 17 April 2023].

[11] P. Diamond, “Cloud storage vs. on-premises servers: 9 things to keep in mind,” 25 September 2020. [Online]. Available: https://www.microsoft.com/en-ww/microsoft-365/business-insights-ideas/resources/cloud-storage-vs-on-premises-servers. [Accessed 23 March 2023].

[12] D. Schrader, “Open Port Vulnerabilities List,” 4 August 2022. [Online]. Available: https://blog.netwrix.com/2022/08/04/open-port-vulnerabilities-list/. [Accessed 28 February 2023].

[13] MKS075, “Difference between site to site VPN and remote access VPN,” 21 February 2023. [Online]. Available: https://www.geeksforgeeks.org/difference-between-site-to-site-vpn-and-remote-access-vpn/. [Accessed 1 March 2023].

[14] Softether Project, “Local Bridges,” 1 March 2023. [Online]. Available: https://www.softether.org/4-docs/1-manual/3._SoftEther_VPN_Server_Manual/3.6_Local_Bridges. [Accessed 1 March 2023].

[15] Fedora Engineering Steering Committee, “Fedora Project Shedules,” 1 March 2023. [Online]. Available: https://fedorapeople.org/groups/schedule/. [Accessed 1 March 2023].

[16] The CentOS Project, “CentOS Stream,” 1 March 2023. [Online]. Available: https://www.centos.org/centos-stream/. [Accessed 1 March 2023].

[17] Debian, “Reasons to use Debian,” 11 February 2023. [Online]. Available: https://www.debian.org/intro/why_debian. [Accessed 1 March 2023].

[18] thomasrutter, “Is Ubuntu LTS based on Debian Unstable or Testing?,” 23 November 2015. [Online]. Available: https://askubuntu.com/questions/701345/is-ubuntu-lts-based-on-debian-unstable-or-testing. [Accessed 1 March 2023].

[19] Canonical, “Releases,” 12 January 2023. [Online]. Available: https://wiki.ubuntu.com/Releases. [Accessed 1 March 2023].

[20] Acronis, “Acronis Cyber Protect Home Office,” 2 March 2023. [Online]. Available: https://www.acronis.com/en-us/products/true-image/. [Accessed 2 March 2023].

[21] Veeam, “Veeam Backup & Replication Community Edition,” 2 March 2023. [Online]. Available: https://www.veeam.com/virtual-machine-backup-solution-free.html?ad=menu-products. [Accessed 2 March 2023].

[22] Duplicity, “Duplicity,” 7 February 2023. [Online]. Available: https://duplicity.us/. [Accessed 2 March 2023].

[23] The Borg Collective, “Borg Documentation,” 23 March 2023. [Online]. Available: https://borgbackup.readthedocs.io/en/stable/. [Accessed 23 March 2023].

[24] P. Ford-Hutchinson, “Securing FTP with TLS,” RFC Editor, 2005.

[25] H. Jehva, “SFTP vs SCP – What’s the Difference for Secure File Sharing?,” 23 May 2022. [Online]. Available: https://cloudinfrastructureservices.co.uk/sftp-vs-scp-whats-the-difference-for-secure-file-sharing/. [Accessed 10 March 2023].

[26] OpenWrt, “OpenWrt on x86 hardware (PC / VM / server),” 4 January 2023. [Online]. Available: https://openwrt.org/docs/guide-user/installation/openwrt_x86. [Accessed 3 March 2023].

[27] TekLager, “Which Operating System should I have on my router?,” 30 December 2022. [Online]. Available: https://teklager.se/en/knowledge-base/choosing-router-operating-system-pfsense-vs-opnsense-vs-openwrt/. [Accessed 3 March 2023].

[28] pfSense, “L2TP VPN,” 6 July 2022. [Online]. Available: https://docs.netgate.com/pfsense/en/latest/vpn/l2tp/index.html. [Accessed 5 March 2023].

[29] pfSense, “Choosing a VPN solution,” 2 August 2022. [Online]. Available: https://docs.netgate.com/pfsense/en/latest/vpn/selection.html. [Accessed 5 March 2023].

[30] OpenVPN, “OpenVPN 3 Autoload feature,” 12 November 2021. [Online]. Available: https://github.com/OpenVPN/openvpn3-linux/blob/master/docs/openvpn3-autoload.md. [Accessed 5 March 2023].

[31] Netgate, “OpenVPN Remote Access Configuration Example,” 10 June 2022. [Online]. Available: https://docs.netgate.com/pfsense/en/latest/recipes/openvpn-ra.html. [Accessed 4 April 2023].

[32] Ubuntu, “Introduction to fstab,” 21 August 2017. [Online]. Available: https://help.ubuntu.com/community/Fstab. [Accessed 14 April 2023].

[33] Linux manual page, “umask,” 22 March 2021. [Online]. Available: https://man7.org/linux/man-pages/man2/umask.2.html. [Accessed 14 April 2023].

[34] OpenVPN, “OpenVPN 3 Client for Linux,” 1 April 2023. [Online]. Available: https://openvpn.net/cloud-docs/openvpn-3-client-for-linux/. [Accessed 1 April 2023].

[35] OpenVPN, “OpenVPN 3 Autoload feature,” 20 February 2023. [Online]. Available: https://github.com/OpenVPN/openvpn3-linux/blob/master/docs/openvpn3-autoload.md. [Accessed 2 April 2023].

[36] D. Jansen, “How to dump & restore a MariaDB/MySQL database from a docker container,” 27 December 2019. [Online]. Available: https://davejansen.com/how-to-dump-and-restore-a-mariadb-mysql-database-from-a-docker-container/. [Accessed 13 April 2023].

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Mihkel Kiil

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Secure offsite data backup solution for a small business”, supervised by Aleksei Talisainen
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

10.05.2023

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – OpenVPN client autoload daemon

```
[Unit]
Description=OpenVPN 3 Linux configuration auto loader and starter
After=network.target dbus.service

[Service]
Type=oneshot
User=backup_script
ExecStart=/usr/sbin/openvpn3-autoload --directory
/home/backup_script/.openvpn3/autoload
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Appendix 3 – Regular script to perform a backup (backup.sh)

```
. "/home/backup_script/.backup-keys"
. "/home/backup_script/backup.config"
docker exec mariadb mysqldump --user=wordpress --password="$WP_PASSWORD"
wordpress > /mnt/storage/archive/Server/wordpress.sql
rsync -a -O --no-perms --delete /opt/wordpress/html/wp-content/
/mnt/storage/archive/Server/wp-content/
for host in "${HOSTS[@]}"
do
    /usr/bin/duplicity  --encrypt-sign-key="$GPG_KEY" --full-if-older-than 7D
$BACKUP_SOURCE sftp://$host/$BACKUP_DESTINATION --copy-links
2>/home/backup_script/backup-$host.log
    if [ -s /home/backup_script/backup-$host.log ]; then
        cat /home/backup_script/backup-$host.log | mail -s "Errors during
backup to $host" $ADMIN_EMAIL
    fi
    /usr/bin/duplicity remove-all-but-n-full 2 --force  --encrypt-sign-
key="$GPG_KEY" sftp://$host/$BACKUP_DESTINATION
done
unset PASSPHRASE
unset GPG_KEY
```

Appendix 4 – Backup monitoring script (backup-retry.sh)

```
. "/home/backup_script/.backup.keys"
. "/home/backup_script/backup.config"
host_success = 0
runtime="1 hour"
endtime=$(date -ud "$runtime" +%s)
while [[ $(date -u +%s) -le $endtime ]]
do
    for host in "${HOSTS[@]}"
    do
        if [ -s /home/backup_script/backup-$host.log ]; then
            /usr/bin/duplicity  --encrypt-sign-key="$GPG_KEY" --full-if-
older-than 7D $BACKUP_SOURCE sftp://$host/$BACKUP_DESTINATION --copy-links
2>/home/backup_script/backup-$host.log
        fi
        if ![ -s /home/backup_script/backup-$host.log ]; then
            host_success++
            if [ -f /home/backup_script/$host-email.sent ]; then
                rm /home/backup_script/$host-email.sent
            fi
        fi
        if host_success == ${#HOSTS[@]}; then
            break 2
        fi
    done
done
unset PASSPHRASE
unset GPG_KEY
for host in "${HOSTS[@]}"
do
    if [ -s /home/backup_script/backup-$host.log ] && ![ -f
/home/backup_script/$host-email.sent ]; then
        echo "Backup for $host in retry status for 1 hour" | mail -s "Backup
in retry status" $ADMIN_EMAIL
    fi
done
```

.backup.keys: stores the passphrase and key ID for GPG

```
PASSPHRASE=”redacted”
GPG_KEY=”redacted”
```

Appendix 5 – Backup configuration file

```
HOSTS=("172.29.0.252:14503" "172.29.0.251")
ADMIN_EMAIL="sandy@sandybridge.xyz"
BACKUP_SOURCE="/mnt/storage/archive"
BACKUP_DESTINATION="/mnt/storage/tallinn-backup"
```

Appendix 6 – VPN connection monitoring script

```
#!/bin/bash
. "/home/backup_script/backup.config"
COUNT=4
for myHost in ${HOSTS[@]}
do
host="${myHost%%:*}"
count=$(ping -c $COUNT $host | grep 'received' | awk -F',' '{ print $2 }' |
awk '{ print $1 }')
touch /home/backup_script/hosts_down
if [ $count -eq 0 ]; then
# 100% failed
if ! grep -q $host /home/backup_script/hosts_down; then
echo "Server $host failed at $(date)" | mail -s "Backup Server Down"
$ADMIN_EMAIL
echo "Host : $host is down (ping failed) at $(date)"
echo "$myHost" >> /home/backup_script/hosts_down
fi
elif grep -q $host /home/backup_script/hosts_down; then
echo "Server $host back online at $(date)" | mail -s "Backup Server Up"
$ADMIN_EMAIL
echo "Host : $host is up (ping success) at $(date)"
sed -i "/$host/d" /home/backup_script/hosts_down
fi
done
```