|  |
| --- |
| Sea Level Operations LLC |
| Zero Trust Remote Access & Administration |
|  |



**Building A High Security Remote Access / VDI Platform For**

**RDP, SSH, VNC & Kubernetes Clients**

**Security Framework Alignment:**

**ISO27001**: A.9.2.3, 9.4.3

**NIST SP 800-171**: 3.1.3, 3.1.14, 3.1.5, 3.5.2, 3.5.3, 3.13.3, 3.13.15

**NIST SP 800-53**: AC-17(2), AC-17(3), SC-5(3)

**CIS V7** 4.3, 4.5, 4.6, 16.3

**CIS V8** 6.3, 6.4, 6.5, 6.7, 16.7

**CIS Hardening Benchmark**: Level 2 Server – Defence In Depth

**ACSC Essential 8**: Maturity Levels 2 & 3

Contents

[1 Lab Overview 4](#_Toc103072284)

[2 Lab Prerequisites 5](#_Toc103072285)

[3 What Is Linux? 7](#_Toc103072286)

[3.1 Getting started with Linux 8](#_Toc103072287)

[4 Sizing Guacamole Jump Server Resources 9](#_Toc103072288)

[5 Ubuntu 20.4.04 Installation 10](#_Toc103072289)

[6 Securing Linux Administrative Access With Public/Private Keys 15](#_Toc103072290)

[6.1 SSH Key Creation 15](#_Toc103072291)

[6.2 Other key formats 16](#_Toc103072292)

[6.3 Enabling SSH Key Authentication 17](#_Toc103072293)

[6.4 Testing SSH Key Authentication 17](#_Toc103072294)

[7 Linux Filesystem Access Via WinSCP 18](#_Toc103072295)

[8 Installing Guacamole, MySQL & Tomcat 19](#_Toc103072296)

[9 Adding Nginx Reverse Proxy & SSL Frontend 20](#_Toc103072297)

[9.1 Additional Nginx Config Tweaks 20](#_Toc103072298)

[10 Hardening Rationale Against Brute Force, DDOS & Other Malicious Exploits 21](#_Toc103072299)

[10.1 Hardening Guacamole Against Internal Bad Actors 21](#_Toc103072300)

[10.1.1 To setup SSL encryption between Guacd and the Guacamole client: 22](#_Toc103072301)

[10.2 Hardening Nginx Against External Threats 24](#_Toc103072302)

[10.3 Dynamic IP Backlisting With Linux 27](#_Toc103072303)

[10.3.1 Installing And Configuring Fail2Ban 27](#_Toc103072304)

[10.3.2 Testing Dynamic Blacklisting 28](#_Toc103072305)

[11 Branding the Guacamole Login Page (optional) 29](#_Toc103072306)

[12 Integrating Guacamole with Active Directory (optional) 30](#_Toc103072307)

[13 Hardening Ubuntu OS 32](#_Toc103072308)

[13.1 Disabling SSH Password Authentication 32](#_Toc103072309)

[13.2 Linux OS Two Factor Local Authentication (optional) 33](#_Toc103072310)

[13.3 Firewall Configuration 34](#_Toc103072311)

[13.4 Host Based Intrusion Detection & Prevention (HIPS/HIDS) 35](#_Toc103072312)

[13.5 Applying The CIS Level 2 Security Compliance Benchmark Configuration 35](#_Toc103072313)

[13.6 Setup Automatic Updates For Ubuntu 36](#_Toc103072314)

[14 Security Auditing With The Ubuntu Security Guide 36](#_Toc103072315)

[15 Operating Guacamole 37](#_Toc103072316)

[15.1 Logging on to Guacamole 37](#_Toc103072317)

[15.2 Creating Remote Connection Profiles With Guacamole 38](#_Toc103072318)

[15.3 Viewing Recorded Guacamole Sessions 39](#_Toc103072319)

[15.4 Managing Open Session Restrictions 40](#_Toc103072320)

[15.5 Administering Guacamole Users & Groups 40](#_Toc103072321)

[15.6 RBAC Group Access Strategy 41](#_Toc103072322)

[15.7 Organisational Strategy For Managing Larger Numbers of Connection Profiles 42](#_Toc103072323)

[15.8 Sharing Of Connection Profiles 43](#_Toc103072324)

[15.9 Remote Clipboard Access 44](#_Toc103072325)

[15.10 Remote File Transfer (Upload) 44](#_Toc103072326)

[15.11 Remote File Transfer (Download) 45](#_Toc103072327)

[15.12 MYSQL Database Backup & Restore 45](#_Toc103072328)

[15.12.1 Manual MySQL database backup: 45](#_Toc103072329)

[15.12.2 Scripting MySQL database backup: 45](#_Toc103072330)

[15.12.3 Automating MySQL Database Backup With Cron: 47](#_Toc103072331)

[15.12.4 Copying Linux MySQL Backups To Another location 49](#_Toc103072332)

[15.12.5 Manually restoring or migrating the Guacamole MySQL database 49](#_Toc103072333)

[16 Root Access After OS Hardening 49](#_Toc103072334)

[17 Ubuntu 20.04.4 Bug Fixes 51](#_Toc103072335)

[17.1 VMware Log Flood Bug Fix 51](#_Toc103072336)

[17.2 Noisy Syslog in VMware Fix 52](#_Toc103072337)

**Document Change Log**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Notes / Revisions |
| 21/4/2022 | 1 | David Harrop | Initial Draft for Ubuntu 20.04.4 |
|  |  |  |  |

# Lab Overview

This lab is a recipe for a scratch build of a Secure Remote Administration/VDI Server (AKA Jump Server, AKA Jump Box) Based upon Guacamole 1.40 , Ubuntu 20.04.4, Apache Tomcat, Nginx & Let’s Encrypt SSL.

Key tasks within this lab are set in four sequenced stages. The order these are stages are completed is key to the successful application of security hardening and baselining tasks. Performing tasks out of sequence will lead to security restrictions impeding the configuration works:

**Stage One – Server Build**

* Gaining familiarity with Linux
* Ubuntu installation & Updating
* Establishing encrypted SSH authentication
* Installation and configuration Guacamole, MySQL and Apache Tomcat
* Installing and configuration of Nginx as a front-end reverse proxy
* Configuring Let’s Encrypt for HTTPS & SSL Connectivity

**Stage Two – Security Hardening**

* Hardening against internal RDP exploits
* Hardening Nginx front end against external DDOS attacks
* DDOS performance validation testing
* Hardening Linux SSH against brute force exploits
* Configuring automated email alerting
* Inspecting and interpreting Ubuntu security log files
* Ubuntu Firewall Configuration
* Hardening Ubuntu to CIS security benchmarks
* Administering a hardened system
* Security Audit reporting

**Stage Three – Optional Use Case Specific Configurations**

* Corporate Branding
* Active Directory Authentication Integration
* 2FA for the Linux OS

**Stage Four – Daily Operations**

* Creating & Managing Remote Connections
* Logging And Recording Remote Connections
* Administering Users, Groups & Connection Permissions
* Backup and Restore Operations

# Lab Prerequisites

This lab is intended for an audience of moderate technical background with 2 years + experience building or supporting a range of back-office, cloud infrastructure or wide area networks.

**Skills Required**

* Good familiarity with installing operating systems, ISO and USB boot media and method
* Familiarity with “Putty” or similar terminal applications
* General TCP/IP and networking knowledge
* Familiarity with password authentication vs PKI concepts
* Good familiarity with at least one hypervisor type
* Basic understanding of Linux command line, and the use of Linux text editors
* Ability to administer public DNS
* Ability to administer DHCP
* Optional: simple web design / CSS / json knowledge for branding tasks

**Software Resources Needed:**

* **Puttygen**

<https://www.puttygen.com> (Easily generates SSH key pairs in windows)

* **Winscp**

<https://winscp.net/eng/index.php> (Essential file transfer utility when working with Linux platforms)

* **Putty**

<https://www.putty.org/> (essential terminal utility)

<https://www.solarwinds.com/free-tools/solar-putty> (Solar Putty is a user-friendly alternative)

* **Notepad++**

<https://notepad-plus-plus.org/downloads> (text editor that won’t break Unix style CR line ends)

* **Ubuntu 20.04.4 (Server) installation media**

<https://ubuntu.com/download/server>

* **A Kali Linux VM (Kali is the industry standard pen-testing platform )**

https://www.kali.org/get-kali/#kali-virtual-machines

* **A suitable local or privately accessible hypervisor,** or
* **A cloud provider supplied Ubuntu 20.04.4 image.** (Cloud provider images may be altered from the standard Ubuntu install therefore elements of this guide may differ in your environment)

**Networking Pre-Requisites**

* **An available public IP & FQDN DNS** name established for this Jump Server i.e. jumpbox.yourcompany.com. The above FQDN entry in your public DNS should be created prior to starting this build as DNS may take some time to replicate. Let’s Encrypt certificate installation will fail without this step.
* **Ensure correct firewall/forwarding rules** to the allocated public address allowing both TCP ports 80 & 443 inbound are established, and that no other content restrictions or geo-blocking is enforced for the intended public IP. Geo-blocking settings can be tightened later after install. These rules should be toggled on and off at various build stages
* **No Proxy**: Ensure the desired IP address is allowed open outgoing internet access without the need for a proxy
* **Restricted DMZ/VLAN**: It is strongly suggested that any private network in which the Jump Server resides be a separate and restricted DMZ/VLAN, and that only necessary remote connection protocols i.e. TCP 3389 & 22 be allowed into any production networks from this Jump Server IP. Internet facing devices or services should never be placed in an open network that is shared with many other production workloads.
* **Static DHCP Reservation and VM Static MAC Address**: It is advisable to create a 12 hour static DHCP reservation against the MAC address for the new Jump Server. This step will keep lab operations consistent and avoid potential networking issues later should VM images be rolled back by snapshot, or by their movement between testing and production infrastructure, or in and out of VM backups etc. This step also requires that a static MAC be set on your VM’s config (e.g. VMware defaults to dynamic MAC allocation)

# What Is Linux?

Linux is a free computer operating system authored by a massive community of open-source developers. Unlike Windows, Linux is a crowd-sourced project that anyone can alter and adapt to their exact and very specific use case. This level of customisation available in Linux is something the commercial software business model cannot inherently achieve.

While other operating systems protect their code from outside interference, Linux is open for all forms of collaboration. This transparency has earned Linux a reputation as a breeding ground for much innovation and is well recognised as the more robust and secure platform to house critical applications, as is evidenced by Linux’s dominance in the datacentre. Because the datacentre is so underpinned by Linux i.e. bare metal virtualisation, enterprise databases, router/switch firmware and so much more, most datacentre roles will require some level of Linux familiarity.

Beyond the datacentre, Linux is literally everywhere and actually shaped the backbone of the internet era. It runs more than just our phones or the email and web servers that underpin ecommerce, Facebook or Google. Linux is also in games consoles, smart TVs, our cars. Even Mac OS and Chromebooks are Linux!

Many proprietary software products or services contain elements drawn from features or code that at some point originated in Linux. Elon Musk’s Starlink is one such example, with satellite ground stations being designed around a free embedded variety of Linux. Application containerisation is another example of a feature first available on Linux.

Because of its infinite configurability, in a smaller I.T. dept Linux also presents as an extremely useful custom solution platform, particularly for use cases where a vendor shrink-wrap product does not exist, or the economics of one licensing a proprietary product may not be justifiable.

With just a little bit of practice and familiarity, particularly in the smaller operation vertical, a confident approach to Linux can make security far more accessible.

In the realm of cybersecurity activities, Linux is an absolutely mandatory staple because most tools for vulnerability analysis or penetration testing require very granular control over the OS to be able to function.

There are two main Linux server variants from two major Linux branches: Debian and RedHat. Ubuntu is a variant of Debian, Fedora is a variant of RedHat. Variants are typically differentiated by their underlying software package management systems and their total available software set (collectively known as a distribution).

RedHat Enterprise Linux (RHEL) is solely enterprise focused, (as is CentOS – which is the same distribution as RHEL but without a support contract.) RedHat support is a paid subscription, therefore public support forms and resources tend to be less numerous with RHEL than with Debian. Debian also spawns a wider range of variants including several desktop products and so many IOT devices. Importantly, most how-to support articles tend to represent or discuss a solution assuming Ubuntu, making this the most suitable place to start a Linux journey.

## Getting started with Linux

Linux Servers have no GUI, therefore all Linux administration relies on the command line or manipulating text configuration files with a text editor in an SSH session. This lab will walk the user through all the required steps and configurations needed. This lab also aims to showcase the key skills for working with Linux and build confidence though the achievement of a fairly advanced outcome.

All command line or paste code in in this document is shown in **this font and highlight.** Always remember that all Linux commands are case sensitive.

**Some first commands to familiarise with when using the Linux “Nano” text editor are:**

**sudo nano** opens a Linux text editor as the sudo (root user) at your current filesystem location

**To copy text:** Highlighting text by dragging and highlighting with the left mouse button, whatever is highlighted is immediately captured to the clipboard.

**To paste text:** Right clicking inside the text editor window pastes whatever is on your clipboard

**Other useful tips**

If you make an error, cancelling without saving with **ctrl +x** is your friend!

To save a configuration file: **ctrl +x | y enter**

**ctrl +c** cancels any running script.

To exit an SSH shell, type **exit**

**cd ~** takes the user to their home directory.

**sudo** typed in front of any command is akin to “run as Administrator” in Windows.

To access the root account shell, **su -**

**exit** returns to the initial user shell from su root elevation.

**chmod +x <scriptpath>/<scriptname>** gives a text script file executable permission.

**sudo ./<scriptpath/<scriptname>** executes a script as root.

**cat <filepath/filename>** displays the contents of a text file.

Below is also extremely useful when appending multi-line strings into a configuration file where sudo permissions are required to do so. ( -a = append. Subtract the -a to overwrite):

**Creating custom scripts to automate commands:**

All custom Linux scripts must start with top left a nominated command interpreter and path to process the script contents. The most common interpreter to start with is bash: **#!/bin/bash**

There are many variations and techniques to achieve the same outcome when editing a Linux configuration. This guide will deliberately introduce some differing ways Linux can be administered to assist in building familiarity with this versatile operating system.

Remember to take regular Offline VM snapshots as you go

# Sizing Guacamole Jump Server Resources

Nginx & Apache are marquee internet applications capable of handling hundreds of thousands of connections. Guacamole is also a mature and enterprise capable product, as evidenced by government & enterprise security management providers such as Cyber Ark embedding Guacamole into their suite of security management and enforcement products.

Guacamole is quite low in its resource needs. As a general starting point, size your initial single VM at **2 CPU core & 4GB RAM for Ubuntu + 1 extra CPU core and 2GB RAM for every 25 concurrent users.** Storage capacity will depend on the volume and size of file transfers between client and remote device as a Jump Server must cache and such transfers. A good housekeeping and secure strategy is to use the Linux /tmp directory as the file transfer cache as this area is cleared after every reboot. For a test lab 50GB of disk will be adequate, but in production much more space may be needed to suit individual requirements

For larger deployments supporting remote worker type scenarios, Jump Servers can be load balanced in exactly the same ways as any other web application. In most cases, when used as a “bastion” server to access a small selection of sensitive infrastructure located in a datacentre or cloud infrastructure, a single Jump Server may suffice. Amazon and Azure may also offer pre-cooked Guacamole bastion server images, however, cloud provider supplied images often rely on pre-configured docker applications that offer only limited customisation ability without much extra complexity if targeting a service provider level of security baseline.

Depending on the scale of deployment and other network or security requirements, a mix of Nginx proxy front end VLAN / Guacamole and Tomcat middle layer VLAN / and a MySQL database backend VLAN may be appropriate. This guide focuses on the single server build configuration however the contained details in this guide can be translated into many varied deployment scenarios.

Diagram

Description automatically generated

# Ubuntu 20.4.04 Installation

If utilising a cloud provider for your Ubuntu image, skip this installation step and follow your cloud provider’s instructions for creating a Ubuntu 20.04.4 image.

Please ensure the external inbound firewall rules mentioned in the prerequisites section are temporarily turned OFF.

Boot your new server with the install media and follow the below diagrams.

1. Text

   Description automatically generated with medium confidenceSelect your language

Graphical user interface, text

Description automatically generated

1. Configure your appropriate keyboard and regional locale
2. Configure network. Text

   Description automatically generatedDHCP is the default. If you configured a static DHCP reservation as mentioned in the prerequisites, hit continue, else configure your specific network requirements here.
3. Graphical user interface, text

   Description automatically generatedDon’t configure a proxy as the Jump Server will act as its own proxy. Ensure no proxy is needed for this server to reach the internet.

1. Graphical user interface, text, website

   Description automatically generatedYou can either research and enter your closest Ubuntu repository mirror here, or simply accept the default.
2. Graphical user interface, text

   Description automatically generatedAccept the defaults for storage configuration.

1. Text

   Description automatically generatedAccept the default storage configuration summary

Text

Description automatically generated

1. This is a warning that irreversible action is about to occur to the disk. Continue
2. Graphical user interface, text

   Description automatically generatedCreate the initial user

Graphical user interface, text

Description automatically generated

1. Leave this blank, hit done.

Graphical user interface, text

Description automatically generated

1. You MUST check the “Install OpenSSH Server” box. Hit done and the installer will continue.
2. Text

   Description automatically generatedOnce the install is complete, remove your install media and reboot.
3. For Linux servers on **VMware platform only**:

To allow full integration with the VMware hypervisor, one last installation step will be to install the Ubuntu native version of VMware tools:



If using VMware, before proceeding please see the bug fix section and the end of this guide to correct two syslog flood issues.

1. **For HyperV platform users** there are a range of potential configurations and updates availale for full HyperV enhanced session compatibilty as well as for Azure-Linux integration. Please check your the most up to date vendor documentation on these points.

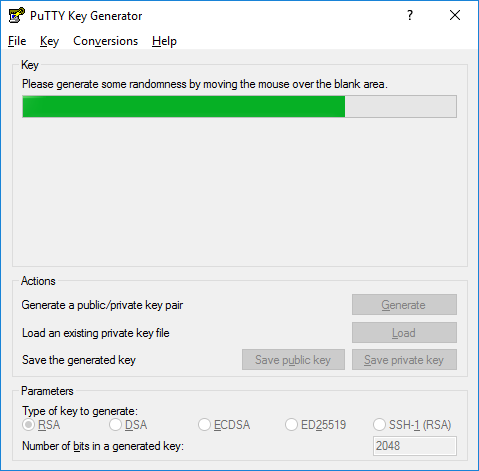
**SNAPSHOT NOW**

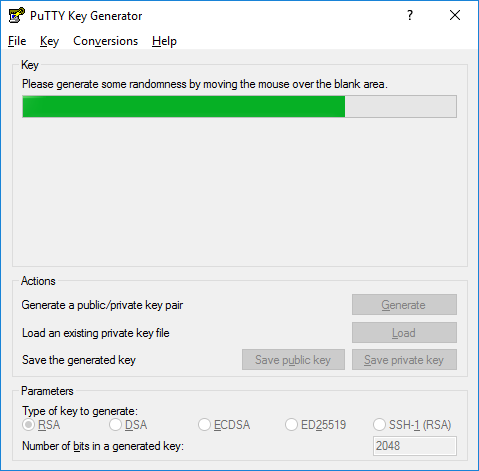
# Securing Linux Administrative Access With Public/Private Keys

If utilising a cloud provider Ubuntu image for this lab, it is likely that your provider will have a proprietary method for managing SSH access and any associated SSH keys. If so, please skip this step and follow your cloud vendor’s instructions.

## SSH Key Creation

The aim of SSH key deployment is to increase security through disabling regular username/password shell access, and to support an alternate form of SSH authentication via encryption keys. To achieve this, we must first create the SSH new keys.

On windows, open Puttygen, select RSA & enter 2048 as the key size, then click Generate.



You must keep moving your mouse pointer randomly over the blank box as Putty uses these mouse movements to create randomness. You may need to move the mouse for some time, depending on the size of your key. As you move it, the green progress bar should advance.

Once the green progress bar becomes full, the actual key generation takes place. This may take a short while. When complete, the public key should pop up in a new dialog window as pictured next.

Graphical user interface, text, application, email

Description automatically generatedNote the box as highlighted in red. Copy/paste these highlighted contents to a new text file **called jumpbox-ssh-public-key.txt**. This file will be needed later.

Do not enter a passphrase in the dialog boxes

Next, save BOTH the public and the private keys with their respective save buttons.

When saving the private key, you will be prompted about saving a private key without a passphrase. You must select YES to save the private key WITHOUT a passphrase, as a passphrase will inhibit some automations and later WinSCP access.

Give each key a clear name, e.g. **jumpbox-ssh-private.ppk** & **jumpbox-ssh-public.key**

Keep these 3 new key files you have just created in a secure place!

## Other key formats

IMPORTANT: The first **jumpbox-ssh-public-key.txt** file just created is in a format Ubuntu can accept right away.

If you take a look at the other public key file (jumpbox-ssh-public.key) you will see that this same public key is in a different format.

Security Keys come in a range of formats. The above key creation method can be used to create SSH keys for any other SSH device. Please note however that Putty formatted (\*.ppk) private keys must be converted to the OpenSSH format for use in the Guacamole GUI. To convert any ppk private key file for use with the Guacamole GUI, open Puttygen and go to:

**File** | **Load Private Key** (& browse to your \*.ppk file) | **Conversions** | **Export Openssh** | Click **Yes** (for no password) | **Save** the new file (e.g. jumpbox-ssh-private-key.openssh)

The contents of the new OpenSSH formatted private key file can now be readily pasted into a Guacamole remote Connection Profile.

## Enabling SSH Key Authentication

Open the contents of the **jumpbox-ssh-public.txt** file created above and save the contents to your clipboard. To copy the new public key contents up to the Linux server, from an SSH session:



The next commands inject the publickey.txt file just saved into the **/home/<username>/.ssh/authorized\_keys** file, which is the hidden location where Linux stores SSH public keys for each individual user account. The publickey.txt file is then removed.



## Testing SSH Key Authentication

***Graphical user interface, text, application

Description automatically generated***Pictured shows how to configure Putty to use a private key during an SSH connection.

First, delete any other saved Putty session profiles for this new server.

Next, look for the “Auth” section at bottom right then browse to the location of your saved private key .ppk file.

You can now navigate up to the “Session” section to enter in the server IP address.

Save this new session profile with the new private key information. If working correctly, every time you open an SSH session with this new profile you will be prompted only for a username before being granted access.

**SNAPSHOT NOW**

# Linux Filesystem Access Via WinSCP

SSH key authentication with WinSCP functions in a similar manner to Putty.

1. Launch WinSCP
2. Click “New Site”
3. Select SCP as the protocol
4. Enter your server’s IP address
5. Ensure the port is set to 22
6. Enter the desired username
7. Leave password blank
8. Click “Advanced” and in the popup pictured, select Authentication, then browse to the location where your private key is stored. (This key must remain accessible to WinSCP when starting up future sessions.)
9. Click Ok, and the Save. You will be prompted to give this new connection a name.
10. Graphical user interface, text, application

    Description automatically generatedTo connect, simply double click the connection name you just created. The WinSCP file then will open a file system view of your user’s home directory.

Graphical user interface

Description automatically generatedNavigating up and down file paths in WinSCP is similar to that with Explorer. To reach the root of the file system hit the “/” button. To navigate back to your home directory hit the button with a house icon.

# Installing Guacamole, MySQL & Tomcat

Locate the supplied installer script files:

**install-guacamole1.40-only.sh**

**install-ssl-letsencrypt-for-nginx.sh**

Using WinSCP, copy the supplied script files up to the Linux home directory.

Establish a new putty session to the Linux server, and type:

****The user will now be prompted for the following inputs:

*Install TOTP?* Answer **Yes** to enable regular 2FA apps such as google authenticator, or **no** to install Duo plugins instead (Duo users should follow the further prompts)

*Install MYSQL?* - Select **Yes**

*Enter MySQL root password:* Supply a strong password. Record this password (needed for backups)

*Enter MySQL guacamole\_user password:* Supply a strong password, record this password too.

The script will continue to run for several minutes, and when complete, browse to: http://ip.add.re.ss:8080/guacamole

Graphical user interface, application

Description automatically generated

Default username is **guacadmin**

Default password is **guacadmin**

**Important:** When any Guacamole user first logs on (if TOTP or Duo was selected at install) they will be prompted to set up their 2FA authentication via a QR code. Setting 2FA for the guacadmin account therefore requires some thought as to whom this QR code will be paired to.

**Important:** Before turning on external inbound firewall rules, the guacadmin password and 2FA must be setup. To change the guacadmin password, login with the default credentials and navigate to Settings | Preferences | and type the new password, then hit update password button.

**SNAPSHOT NOW**

# Adding Nginx Reverse Proxy & SSL Frontend

**External inbound firewalls rules must be on to complete this next step.**

For an added security layer, the second script automatically updates Ubuntu, then installs and configures Nginx as a reverse proxy in front of the Guacamole service. The script then requests a new SSL certificate service from Let’s Encrypt and installs all the relevant settings to enable Nginx to begin to serve trusted HTTPS sessions verified by a root CA. Let's Encrypt is a free, automated, and open certificate authority.

****The user will be prompted for:

*Enter the FQDN for you domain:* (enter the FDDN noted in the prerequisites)

*Enter your email address:*  (typically a webmaster email)

If all of the network prerequisites are setup correctly, Lets Encrypt should report a successful certificate creation and installation. Pay attention to any messages if there are any failures.

Be aware that Lets Encrypt first opens a communication with certificate servers on port 80 to negotiate a secure TLS tunnel before continuing.

Most issues at this stage will relate to firewalls and network connectivity.

Once the script is complete, test your new server <https://jumpbox.public-domain.name/>. (In some instances, depending on your network’s NAT configuration or NAT reflection rules, you may need to test from outside your network.)

## Additional Nginx Config Tweaks

At this point, Guacamole logs will show all remote connections in connection logs as emanating from 127.0.0.1. To fix this, open the Tomcat config file in the nano editor:

****Look for the XML section toward bottom named

***“<Valve className="org.apache.catalina.valves.RemoteIpValve”***

Now paste the following text into the abovenamed section:

**SNAPSHOT NOW**

# Hardening Rationale Against Brute Force, DDOS & Other Malicious Exploits

Any internet facing system, from the minute it is connected, is available to every hacker or botnet on the planet.

Although a system may be placed in a secure network behind a firewall, the less an attacker can succeed in flooding mal-crafted packets towards a system or authentication attempts (from internal or external networks), the less footprint there can be for any viable point of attack.

In the case where a Jump Server may sit behind a firewall or network of much higher throughput, tuning the upstream firewall to offer protection under a DDOS scenario is a *big hammer* approach that may also globally impact legitimate use. Also, upstream firewalls can often do little to limit attacks coming from trusted sources over legitimate traffic. Firewalls alone have no control over server memory handling, maximum open file limits, or the connection tuning limits configured on a webserver. Some DDOS attacks are well disguise and need only very low bandwidth to easily cripple a webserver right through a firewall. (e.g. SlowLoris attack)

System availability is also a factor as DDOS style attacks can degrade performance, especially if we allow connections to occur in an unmanaged or unrestricted way. We must also consider internal threats and anticipate the potential for software weaknesses or other such combinations of vulnerabilities.

As such, performing hardening not only at the perimeter, but also inside the OS and applications themselves offers an extra layer of risk management. With this combined approach, a more nuanced and tuneable level of protection can be designed to suit the exact use case as follows.

A picture containing text, sign, screenshot

Description automatically generated

## Hardening Guacamole Against Internal Bad Actors

In operation, the Guacamole server executable (Guacd) communicates to foreign systems using native RDP/VNC/SSH protocols, with the Guacamole client translating these sessions into HTML5 compliant communications

After an HTML client connection is started, the Guacamole client application running inside Apache Tomcat starts a Guacamole Protocol session with the Guacamole-server (Guacd.) Guacd then acts as an RDP/VNC/SSH protocol broker for all connections.

By default, the Guacamole client application communicates with Guacd **unencrypted** on TCP port 4822. Any unknown or future vulnerabilities against Guacd may allow the capture or hijack of session traffic by internal bad actors or malware. This aspect can largely be prevented by encrypting these internal communications on port 4822.

### To setup SSL encryption between Guacd and the Guacamole client:

Below is a script to establish all the settings required to enable an SSL wrapper around backend Guacd application traffic.



Copy the above script to your clipboard, then enter the following commands:



The script will prompt the user to enter country/state/locality/org name/ou name/cn & email fields. It is safe to leave all these options blank (simply hit enter 7 times.)

At the end of the script a new certificate is imported into the Java keystore. To complete this step the user is prompted for the password to the Java certificate keystore. The default JRE password is:



Then the script will then prompt whether to **trust** this new certificate.

****

**After reboot, confirm guacd has loaded the certificate and key, and is listening on SSL:**

****

An example running log output after guacd restart is shown below. Confirm that the following lines are present:

*using PEM keyfile /etc/guacamole/guacd.key*

*Using certificate file /etc/guacamole/guacd.crt*

*Communication will require SSL/TLS &*

*Listening on 127.0.0.1, port 4822*

A computer screen capture

Description automatically generated with medium confidence

**SNAPSHOT NOW**

## Hardening Nginx Against External Threats

Before hardening, **from a separate Kali Linux VM,** simulate a DDOS attack against the default Jump Server with the **SlowHTTP** load simulation tool:



The above command executes an aggressive “SlowLoris” method of DOS attack for just a few minutes.

If there is adequate bandwidth available for the DDOS test, the Nginx server should begin to drop and become unstable, likely showing 500 error codes for a time. Keep trying to access the site as it struggles to stay up.

Simultaneously tail the logs to observe the attack in progress:



When the test completes, look in the current directory for the dos-test-results.html report. After completing this chapter, re-run the same test and compare the reports with your own observations.

**Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generatedNginx Before DDOS Hardening Nginx Nginx After DDOS Hardening**

To apply the hardened Nginx configuration:





Finally, below are the revised Nginx start-up parameters.:



The above configuration hardens Nginx against DDOS attacks via:

* Limiting the total number of open files possible
* Setting a maximum connection processing rate for the server
* Setting a maximum volume of requests per IP address per second
* Setting a maximum connection limit per IP address
* Creation of protected memory zones.

When rate limits are breached, error logs will subsequently be created. Specific entries in the Nginx error log are needed to trigger dynamic IP blacklisting rules that will be set by the Fail2Ban application (described in the next section.)

**Sidenote:** Shown in bold within supplied Nginx configuration above, **client\_max\_body\_size 1000000M** is the maximum file size that can pass through the proxy in either direction, here shown as 1TB. Depending on the use case, this figure and can be adjusted up or down to restrict egress from the remote system.

**SNAPSHOT NOW**

## Dynamic IP Backlisting With Linux

**Fail2ban** is an intrusion prevention software framework that protects Linux servers from brute-force attacks. Fail2ban operates by monitoring log files for specific patterns and can perform multiple custom actions whenever a particular log entry appears, including dynamically adjusting firewall rules to block offending IP addresses for a specified amount of time. Alerts can also be configured containing rich network detail about an attacker.

### Installing And Configuring Fail2Ban

To install the Fail2Ban software and alerting mail server dependencies:



After running the above command, the user is prompted to select mail server configuration type. Select “Internet Site” and hit OK. The next screen allows the configuration of the mail server FQDN name that will be shown in alert emails. Enter your appropriate name.

Configure Postfix for email relay:

Test the email relay (remember to check your spam folder): 

Apply a base Fail2Ban filter with some reasonable starting limits:

It is also necessary to adjust the Fail2Ban filter to understand Guacamole 1.4.0 event logs:



### Testing Dynamic Blacklisting

Below are important settings for tuning the **/etc/fail2ban/jail.local** config

**bantime =** (Sets the banning time)

**findtime =** (Sets the period of time sampled)

**maxretry =** (Sets the grace login threshold )

**ignoreip =** (Trusted networks stops internal IP being inadvertently blocked)

Applying these settings in the [default| section will apply the settings to ALL filters. Applying these settings under and individual filter e.g. [filter-name] will provide specific settings for only that filter and override the [default] settings.

#### Testing Guacamole and SSH Blocking:

Adjust the ban time values for each filter to very low value: e.g. 1 minute:



Deliberately try to connect to Guacamole using a wrong username or password. Observer your IP address will be banned from the Jump Server. Do the same for an SSH login and observe how you are cut off for 1 minute.

#### Interrogating System Logs For Banned IP Addresses

Google can present all manner of potential commands for viewing log output. Below are some of the more direct and useful:

**For SSH Logs**

****

**For Guacamole Logs**

Show a count of individual IPs where a failed Guacamole login occurred



# Branding the Guacamole Login Page (optional)

Guacamole supports an extension that allows for custom configuration of the login page for a more professional or custom look

An example of this custom configuration extension can be downloaded from GitHub

<https://github.com/Zer0CoolX/guacamole-customize-loginscreen-extension/raw/master/branding.jar>

Unzip the downloaded branding.jar file with an archive manager (e.g. 7zip) and edit the contents. Editing this file will require familiarity with CSS, json, and some web image design skills. Save all changes and replace the images with your own within the barnading.jar file.

Copy the extensions.jar file to the SSH home directory with WinSCP, then from an SSH session move the branding.jar file to the Guacamole extensions directory:



Clear your browser cache and test

Graphical user interface, application, Word

Description automatically generated

**SNAPSHOT NOW**

# Integrating Guacamole with Active Directory (optional)

To configure LDAP authentication support in Guacamole, a Tomcat extension is required.

Navigate to <https://guacamole.apache.org/releases/1.4.0/>

Download **guacamole-auth-ldap-1.4.0.tar.gz**

To move the LDAP extension file to the Jump server, copy the downloaded file to your home directory on the Jump Server with WinSCP. Next issue the following commands via SSH:



A “Domain Users” level of privilege is sufficient for Guacamole to bind with Active Directory. Create a new Guacamole administrator account, (e.g. jumpboxboss) and then create a new domain account of **EXACTLY THE SAME NAME** in Active Directory using a **DIFERENT PASSWORD** than the matching local Guacamole account.

Disable the default guacadmin account as per **section 15.1 of this guide.**

Edit the /etc/guacamole/guacamole.properties file:



When Logging in with the new LDAP account credentials named in the *ldap-search-bind-dn* line above, that user will be able to browse all the available users based on the OU settings provided below. Confirm this under **Settings | Users**.

**Important notes on /etc/guacamole/guacamole.properties:**

**ldap-user-base-dn:** All Guacamole users must be descendants of this distinguished name.

**mysql-auto-create-accounts: true** is anOptional guacamole.properties entry. A local account is required if you want to apply Guacamole’s 2FA (or other settings) to an LDAP login. This optional line ensures that ALL LDAP user accounts will have a local account created at first logon. If just a selection of LDAP accounts requires 2FA, manually create local accounts that mirror their LDAP account names instead, *without passwords*. At first logon LDAP will authenticate the local account name and 2FA will apply.

**LDAP auth firewall requirements:**

If your Jump Server is operating in a separate network to that of your Active Directory Servers (highly recommended), the Jump Server will require TCP 389 available to all Active Directory Domain Controllers nominated in the **ldap-hostname** directive above.

**RDP Single Sign On**

If the Jump Server and remote RDP client connections will share a common AD domain:

1. Create a Global Security group (e.g. Guac\_Users) and populate it accordingly. Now add this new security group to the built-in “Remote Desktop Group”
2. Connection profiles do not need to be configured with specific usernames and passwords. For seamless RDP connection, remote connection profiles should be configured with ${GUAC\_USERNAME} & ${GUAC\_PASSWORD} as follows.

Graphical user interface, application

Description automatically generated

For testing and troubleshooting LDAP authentication, observe this log:



# Hardening Ubuntu OS

## Disabling SSH Password Authentication

Open the Ubuntu SSH config file with the nano text editor as follows:



In this Nano session, search for and confirm the following lines in the sshd\_config file read as below. Uncomment or update the following lines exactly as per below.





As a final test, try and log back into the server using Putty without directing Putty to use the new SSH private key. Password access should now fail.

## Linux OS Two Factor Local Authentication (optional)

The following config will apply a 2FA challenge to a local console login or for any sudo permissions elevation. SSH Key authentication will not require a 2FA challenge as this may impact future remote automations or tasks. **If configuring 2FA, for redundancy, there should be more than one sudo user account, or more than one auth device paired to the same account.**

To create a second sudo account:



Install Google Authenticator libraries and enable pam to use this:



Update SSH server to utilise 2FA inside the SSH session:

****

Allow 2FA authentication whilst allowing users without 2FA to log on so as to configure 2FA:



Run the 2FA setup:

****

Qr code

Description automatically generated

**Put these codes in a safe place**

To recover access to a 2FA enabled user account:



Type the above command’s output into the same user’s Authenticator app. If it is not possible to access the same authenticator app, delete the google\_authenticator file. If the “nullok” option is still enabled in the /etc/pam.d/sshd file, the user will be able to log in without 2FA and create a new QR code.

## Firewall Configuration

****

## Host Based Intrusion Detection & Prevention (HIPS/HIDS)

Future module pending: Wazuh / Tripwire

## Applying The CIS Level 2 Security Compliance Benchmark Configuration

To access Ubuntu’s proprietary Security Guide tools that can automate the many hundreds of settings required to deliver a well hardened system, a Ubuntu “Advantage for Infrastructure” subscription is required. However, although Ubuntu Advantage is a paid subscription, Ubuntu offers a free tier for 3 concurrent servers under enterprise management.

Create a Ubuntu One login at: <https://login.ubuntu.com/+login>

After creating your Ubuntu login navigate to <https://ubuntu.com/advantage>

Graphical user interface

Description automatically generated

On your new subscription dashboard, look for the free personal token at the bottom and **copy this text**

Next, log in to Ubuntu system with Putty and perform the following commands:



The Ubuntu system will now automatically configure all applicable settings to the CIS “Server Level 2” security benchmark, and prompt for reboot. Some benchmark items may report as “error” during the command output, however this is normal as not all benchmark settings will have a corresponding item to re-configure.

## Setup Automatic Updates For Ubuntu

With the CIS Level 2 benchmark now applied, confirm the Ubuntu automatic kernel security patching service “Livepatch” is now active:



**A picture containing text

Description automatically generated**

To automatically update the remainder of Ubuntu packages (non-kernel) in an unattended fashion, first confirm that automatic package update is turned on:



To allow unattended upgrades to reboot the system, open the following file with:



Now some final clean up commands:



# Security Auditing With The Ubuntu Security Guide

To see your particular CIS compliance score, the Ubuntu Security Guide tools offer CIS benchmark audit reporting:



Security reports are very sensitive and are therefore saved in a restricted path. As the root account has been disabled, to gain access to these reports they must have their ownership changed. It is then easier to move them to the home directory so they can be download with WinSCP.



Connect with WinSCP, hit refresh whilst in the home directory. The html and xml audit report files should now be visible for download to your Windows desktop.

Reports contain deep detail on how to address specific security points and can also act as an educational guide towards attaining the maximum CIS compliance score.

# Operating Guacamole

## Logging on to Guacamole

When first logging on, (if two factor auth was enabled at install) the user will be prompted with a QR code to pair their authenticator app. You must complete this step before being allowed to continue.

Once the QR code is captured to the authenticator app, login with the default admin user:

User: guacadmin

Pass: guacadmin

Enter the authenticator app cose for gaucadmin user

Text

Description automatically generated with low confidence

Graphical user interface, text, application

Description automatically generated

At this time create a new user with “Administer system” rights. E.g. **jumpboxboss**.

All available rights are pictured right. Give the new jumpboxboss account user all rights.

Graphical user interface

Description automatically generated with medium confidence

Now disable the default **guacadmin** account.

## Creating Remote Connection Profiles With Guacamole

Below are the typical items to configure in establishing a secure remote connection over to SSH. In the Guacamole web GUI navigate with: **Settings | Connections | New Connection**

**For creation of an SSH session:**

|  |  |  |
| --- | --- | --- |
| **Edit Connection:** |  |  |
|  | Enter Connection name | e.g. firewall |
|  | Select Protocol | SSH |
| **Concurrency Limits:** |  |  |
|  | Maximum number of connections | 5 (suggested) |
|  | Maximum number of connections per user | 2 (suggested) |
| **Network:** |  |  |
|  | Hostname | hostname or IP of remote host |
| Authentication |  |  |
|  | Username |  |
|  | Password |  |
|  | Private key | Private key in OpenSSH format \* |
|  | Passphrase | (If key requires passphrase) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | Port | 22 |
| **Screen Recording (Optional)** |  |  |
|  | Recording path | /home/<linux-user>/recordings |
|  | Recording name | firewall-${GUAC\_DATE}-${GUAC\_TIME}-${GUAC\_USERNAME}-${GUAC\_CLIENT\_HOSTNAME} |

\*See Other Key Formats section of this guide in 6.2

**For creation of an RDP session:**

|  |  |  |
| --- | --- | --- |
| **Edit Connection:** |  |  |
|  | Enter Connection name | e.g. win2k8 |
|  | Select Protocol | RDP |
| **Concurrency Limits:** |  |  |
|  | Maximum number of connections (in total) | 5 (suggested) |
|  | Maximum number of connections per user | 1 (suggested) |
| **Network:** |  |  |
|  | Hostname | hostname or IP of remote host |
|  | Port | 3389 |
| **Authentication:** | Username |  |
|  | Password |  |
|  | Security Mode | NLA |
|  | Ignore server certificate | yes |
| **Display** | Colour Depth | 16/24/32bit, 256 colour |
|  | Read only | Tick read only for view only (non-interactive) session |
| **Device Redirection:** |  |  |
|  | Support audio in console | No (Default No) |
|  | Disable Audio | Yes (Default Enabled) |
|  | Enable audio input | No (Default Disabled) |
|  | Enable printing | No (Default Disabled) |
|  | Redirected printer name | blank |
|  | Enable drive | Yes (Default No) |
|  | Drive Name | TempShare |
|  | Disable file download | No (Default Enabled) |
|  | Disable file upload | No (Default Enabled) |
|  | Drive path | /tmp/${GUAC\_USERNAME} |
|  | Automatically create drive | Yes (Default No) |
| **Screen Recording (Optional):** |  |  |
|  | Recording path | /home/<linux-user>/recordings |
|  | Recording name | win2k8-${GUAC\_DATE}-${GUAC\_TIME}-${GUAC\_USERNAME}-${GUAC\_CLIENT\_HOSTNAME} |

## Viewing Recorded Guacamole Sessions

The above “Recording name” setting <connection name>-${GUAC\_DATE}-${GUAC\_TIME}-${GUAC\_USERNAME}-${GUAC\_CLIENT\_HOSTNAME} creates a compressed record of each connection as a file named by date-time-username-client\_host\_name\_or\_IP

**To expand a recorded session into a viewable m4v format:**



Alternatively, to convert ALL files in the recordings directory (use with caution as CPU and disk space may be constrained)



Calendar

Description automatically generatedExample output:

Simply copy the new m4v files via WinSCP to your local Windows system for viewing.

## Managing Open Session Restrictions

As every connection to a server creates additional loads, efficient practices need to be applied. If a user leaves Guacamole without exiting their session, each session will remain open indefinitely (unless specific forced timeouts are available and applied at the remote end). This situation could lead to a range of degradations in service. In some cases, there might also be hard limits set on the number of concurrent session allowable at the remote system due to licencing considerations. Because a specific number of concurrent connections is set for each Guacamole Connection Profile, overall and per user, the above issues can be better managed.

However, if a Guacamole user has exhausted their pool of allowed connections, he or she will be unable to reconnect until that pool has unused connections available. To process, some abandoned connections will need to be closed. To close abandoned connections, the user must navigate to: **Settings | Active Sessions**

Here a list of open connections for all systems in their use will be presented. The user can simply select one (or select several via tick boxes) and kill the offending sessions with the “Kill Sessions” button. The user will now be able to start a fresh new connection where previously blocked.

## Administering Guacamole Users & Groups

## RBAC Group Access Strategy

**Settings | Groups | edit or new (group name)**

Hierarchical remote connection groups can be created following the Role Based Access Control principle (RBAC).

With thoughtful design, granular controls around what connections individual users can access can be controlled with inheritance, and centrally managed with simplicity.

**Tip:** For large numbers of remote system connections, establishing a solid naming convention that identifies connections by location and type will help display clearly sorted dropdown lists. Username naming conventions will also be more efficient to work with.

Graphical user interface, text, application

Description automatically generated**To achieve a hierarchical (or RBAC) access model:**

**Graphical user interface, text, application

Description automatically generated**Separately and descriptively named groups should each be created for users, and for remote connections. -

I.e. *Client A Connection-Profiles*, Clie*nt A Users*, or similar.

**For Connection Profile named groups, allocate only**:

* 1. Connection Profile objects under “Connections”
  2. User Groups under “Member Groups”

**For User Groups, allocate only:**

1. Individual usernames under “Member Users”
2. Connection Profile groups under “Parent Groups”

To provide consistent access and avoid configuration drift over time, it is advisable to stick to one method of allocating permissions. Also, although it is possible, avoid allocating permissions at the individual user account level. User groups matched to Connection Profiles is a more scalable way to address consistent access.

## Organisational Strategy For Managing Larger Numbers of Connection Profiles

One extra category of group, allows a further level of hierarchy to be established:

“Organisational Connection Groups” for Users and Connection Profiles act like folders to store connections of a like type or site all together. This becomes particularly useful working with larger numbers of Connection Profiles.

**Organisational Connection Groups**

Graphical user interface, text, application, email

Description automatically generated**Settings | Connections| Create Group**

In the dropdown box, the default group type is set to “organisational group” Give the new organisational group a name.

All new Connection groups will appear in the master list of available connections.

See below. Once one or more connection groups are created, the location field will become a drop-down field where a hierarchy of groups and subgroups can be displayed. From here a logical group structure can begin to be built.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

New Connection Profiles can be created inside of a Connection Group helping keep profile objects and users organised. Existing Connection Profiles can be moved between Connection Groups through “location” field in each Connection Profile’s properties.

## Sharing Of Connection Profiles

Graphical user interface, text, application

Description automatically generated

In some circumstances, being able to supply a 3rd party a view (or full access) to a remote session may be very useful.

Graphical user interface, text, application, email

Description automatically generated Shared Connection Profiles can be created as a duplicate, and optionally for read-only (view only) access.

Once created, shared profiles can be created by any Guacamole user who has been granted the “Create new sharing profiles” right.

Graphical user interface, text, application, email

Description automatically generatedTo access a shared profile, whilst already connected to the connection to be shared, Ctrl+ATL+Shift brings up the Guacamole share menu.

If a shared profile exists, a public web link is provided for access to the shared profile in this menu.

IMPORTANT NOTE: No password or 2FA is required when accessing a shared link. The shared link remains active only until the original 1st sharer session is closed.

## Remote Clipboard Access

Graphical user interface, text, application

Description automatically generated

Ctrl+ATL+Shift brings up the clipboard tool, the remote clipboard is ready to paste the tool’s contents.

When copying text in the remote system, the contents of the remote system clipboard will automatically update in the clipboard tool.

Text

Description automatically generated

## Remote File Transfer (Upload)

Graphical user interface, text

Description automatically generatedGraphical user interface, text, application

Description automatically generatedIf upload is enabled for the specific Connection Profile, and if the remote RDP host supports drag and drop, simply drag files into the TempShare directory on the remote client. A screen refresh may be required to see the uploaded file. Another Option is to open the Guacamole menu with Ctrl+ATL+Shift, and click “Shared Drive”

## Remote File Transfer (Download)

If downloading is enabled for the specific Connection Profile, to collect file(s) from the remote RDP session, simply copy or drag files to the “Download” folder. A save as dialog box will pop up just as though a file was being downloaded from the web.

## MYSQL Database Backup & Restore

In production, the Guacamole database should always be backed up.

Backups can also be used to migrate the Guacamole database between test and production systems.

In the case where there is a need to move to a load balanced scenario, or where maximum security requirements dictate, backups can also be used to move the database backend to a standalone system behind multiple front end Guacd /Nginx systems.

### Manual MySQL database backup:



The above command will start a transaction before running. Rather than lock the entire database, this will let MySQL dump read the database in the current state at the time of the transaction, making for a consistent data dump. The dump file will be named according to file date in YYYY-MM-DD format.

### Scripting MySQL database backup:

Below is a script that will create a zipped backup of the Guacamole database. The script uses a number of variables to make automated scheduling of commands and actions simpler.

You will need to adjust some script variables for your own specific requirements. A description of the variables is as follows:

DB\_BACKUP\_PATH=………*(local filesystem backup location)*

MYSQL\_HOST= ……………..*(host to backup i.e. local or remote MySQL)*

MYSQL\_PORT=………………*(MySQL port, default is 3306)*

MYSQL\_USER=……………….*(MySQL root username)*

MYSQL\_PASSWORD=……..*(Your root password created at install)*

DATABASE\_NAME=…………*(The DB name given at install is g*uacamole\_db*)*

BACKUP\_RETAIN\_DAYS=…*(days to keep a local backup copy)*

RECIPIENT\_EMAIL=…………*(a backup messages alert mailbox)*



The backup script will extract the database backup from Guacamole, compress it, an lastly check for any previous backups over a certain age delete. After you have set your desired variables, copy the backup script up to Ubuntu via an SSH session and manually run it:



### Automating MySQL Database Backup With Cron:

Cron is the Linux equivalent to “Task Scheduler” in Windows.

Cron uses a **MINUTE HOUR DAY MONTH DAY\_OF\_WEEK** format which

equates to 5 placeholders. Each placeholder specifics a specific value for time. Each placeholder can either be a number, or an asterisk.

┌───────────── minute (0 - 59)

│ ┌───────────── hour (0 - 23)

│ │ ┌───────────── day of the month (1 - 31)

│ │ │ ┌───────────── month (1 - 12)

│ │ │ │ ┌───────────── day of the week (0 - 6) (Sunday to Saturday)

│ │ │ │ │

│ │ │ │ │

│ │ │ │ │

\* \* \* \* \* /command/to/execute

Where there is a \* included in a placeholder, it equates to every minute/hour/month/day\_of\_week depending on its relative position in the \* \* \* \* \* placeholder syntax.

Here is an example of the Cron format in use. The below sets a schedule to run a script at 2:15am every Sunday:

**15 2 \* \* 7 /path/to/script/file**

Cron offers much configurability when step values (/), range values (-) and list values separated by a comma as above are introduced:

Here is another to run at midday and midnight only on Saturdays and Sundays In this case the second placeholder for “hour” includes multiple values:

**0 12/24 \* \* 6,7 /path/to/script/file**

Cron granularity is limitless. Below outlines the schedule to run a script at minutes 1, 3, 5, 7, and 9 past every hour but only between 9am and 12pm, and then again at 3pm on Mondays, Thursdays, and Fridays

**1,3,5,7,9 9-12,15 \* \* 1,4,5 /path/to/script/file**

For this lab, here is a suggested starting point for a backup to run at 3am every weekday, please adjust as necessary and copy to your clipboard:



**To save the above schedule into Cron:**



(The first time this command is run the user is prompted to select which text editor should be associated with editing Cron tasks. Select Nano.)

**Text

Description automatically generated**

Right click to paste your schedule in the Cron editor. Multiple lines of schedules can be added in this way. All Cron tasks set up this way will run as root.



A great resource to help calculate your desired backup frequency can be found at <https://crontab.guru/>

### Copying Linux MySQL Backups To Another location

Because SSH is configured to allow authentication without a 2FA challenge, it is simpler to pull files from the Linux system from a Windows system rather than push.

To schedule a copy of MySQL backups (or any other backup) to another location.

1. Ensure putty is installed on the Windows system
2. Schedule your adapted command as follows (enter as one line):



### Manually restoring or migrating the Guacamole MySQL database

Locate your desired backup file.

From an SSH prompt:



# Root Access After OS Hardening

Now that the system has been hardened, all interactive root logins from console or SSH have been disabled. It will no longer be possible to perform some tasks, particularly relating to user accounts, privileges, ownership and certain file system access in this way. Another type of “run as” root is still available, (**su -**) At rare times however, full root access can be more practical or even necessary.

**Root interactive logins can be re-enabled root the following way:**

(This is a temporary measure, revert these changes once root access requirement is complete):

To elevate yourself as root access, from an SSH shell,



Edit the SSH config file:



Edit the Pluggable Authentication Module (PAM) File



Then unlock the root account with



(You will be prompted to give root a new password)

**Reverting to the disabled root configuration:**

To lock the root user once again and restore the previous secure configuration:



Now reverse all above changes made to these files:





# Ubuntu 20.04.4 Bug Fixes

## VMware Log Flood Bug Fix

Under some conditions with Ubuntu 20.04.4, VMware ESXi can cause the Ubuntu syslog to flood with many *sda: failed*device messages, and this many impede the ability to troubleshoot with the syslog.

If using VMware ESXi and the /var/log/syslog shows a constant stream of sda/disk type messages:

Place the text below on your clipboard, then open an new SSH session. Right click in the Putty session to paste the below text as a whole– then press enter.



To check the above changes were copied into the /etc/multipath.conf file:



Now restart the multipath service:



## Noisy Syslog in VMware Fix



Some log features in Ubuntu have recently been depreciated and this can create extra log noise.

Find the following lines and comment these out 

