

FREE
DVD

Tracking MITM

systemd Security

ADMIN
Network & Security



ADMIN

Network & Security

ISSUE 67

systemd Security

Tricks for locking down your system

VMware Workspace ONE

A secure and user-friendly digital workplace

What's New in Windows Terminal

Clickable links and other
command-line innovations

Processor and Memory Affinity Tools

Man-in-the-Middle Attacks

Track down the culprit with Wireshark

Zero Trust Security Strategy

Nitrokey

Hardware-enhanced
security

MITRE

ATT&CK and D3FEND
knowledge databases

DMARC

Secure email
communication

Gatling

Generate load on
servers and services

Endpoint Security
for Windows 10

LINUX NEW MEDIA
The Pulse of Open Source

MEET 150+ LEADING CLOUD & TECHNOLOGY PROVIDERS. JOIN HUNDREDS OF BUSINESS AND TECHNOLOGY LEADERS FOR THOUGHT-PROVOKING PANEL DISCUSSIONS, INSPIRATIONAL KEYNOTES AND PRACTICAL CASE STUDIES. UNMISSABLE NETWORKING OPPORTUNITIES. REGISTER FOR YOUR FREE TICKET TODAY: WWW.CLOUDEXPOEUROPE.COM/ADMINMAGAZINE-LINUXMAGAZINE



Transform the future of your business.

After two-years out, the global event of choice for everyone committed to the design, build and management of digital initiatives and technology architecture is back. If you're a technologist or a business leader in the public, voluntary and private sector, this is the place to give your future a jet-propelled boost of inspiration, ideas and innovation. All the very best suppliers and providers combine with expert-delivered content in one unmissable event.

Learn about the As-A-Service Model, Digital Acceleration, Emerging Tech, Hybrid and Multi-cloud, Sustainable Cloud and more. Come and see your digital transformation soar.

Register for your FREE ticket today:
www.cloudexpoeurope.com/AdminMagazine-LinuxMagazine



CLOUD EXPO EUROPE

2 - 3 March 2022 ExCel, London
www.cloudexpoeurope.com

PART OF

TECH SHOW
LONDON

INCORPORATING



CLOUD EXPO
EUROPE



DEVOPS
LIVE



CLOUD & CYBER
SECURITY EXPO




BIG DATA
& AI WORLD



DATA CENTRE
WORLD

ORGANISED BY

 CloserStill

THE MOST IMPORTANT TECHNOLOGY EVENT FOR BUSINESS IN THE UK

PLATINUM
SPONSORS



GOLD
SPONSORS



PARTNERS



Technology Conferences: We Need Each Other

Technology conferences allow us to exchange ideas, discuss solutions, learn new things, and geek out. Most of all, they give us that human connection we can't get in video chats or from email and text messages.

I think the primary thing that the pandemic has taught us, if nothing else, is that we miss each other. We miss gathering. We miss discussing. We miss learning. And we miss clinking glasses. Technology conferences gave us technonerd everything we could want in the latest in technology: the learning opportunities, the gathering of like minds, the toasting with beer glasses, and the geeking out over someone in a Storm Trooper costume or a modified DeLorean. Tech conferences gave us what we need most: each other.

As a non-monetized sideline, I perform weddings. I've only done a few, but I'd love to do more. I customize the wedding for each couple. I make them personal, and the focus is 100 percent on the two people getting married. That's why, in part, I've never been able to deliver the message I feel describes our need for each other. The story is short, simple, and beautiful.

Whether you believe that an all-powerful God drew us out of the earth or that lighting struck a pool of chemicals some half billion years ago, the first gift given to us was life itself, and the second gift was that of each other. Since those first two one-celled organisms huddled together in the darkness, we have needed each other.

Technology conferences satisfy our need to gather. Zoom calls and virtual meetings just don't have the same effect on our psyches. Sure, we can see each others' faces, we can discuss business, we can share lunches, and we can cover a multitude of topics, but when we look around the room, we are still alone. We all know that isolation isn't healthy, but during a pandemic the opposite, at least physically, is true.

I need to speak directly to a person in a booth. I need to pick and choose my swag. I need to tell my favorite joke when I have the opportunity to introduce a speaker. I need to covet the T-shirt I didn't get. I need to ask questions during a breakout session. I need to sit down uncomfortably at a table full of strangers at lunch time and wait for an opening in the ongoing conversation to satisfy my need to interact with other humans. I'm one of those people who wants to know where everyone is from, what you do, and what your hobbies are. I don't know why. I either find myself being terribly shy and withdrawn or so outgoing that I become the glue of a random group. It's weird, but I must interact.

I have worked from home for so long that my internal filter and whatever boundaries are supposed to exist have eroded away along with some of my manners. I seem to speak almost in chat style now. No, I don't say, "LOL," but my conversational style and tone are instant messaging-esque. I'm sure you understand that statement, having worked remotely yourself.

I like working from home. I really do. I always hated going into an office. I prefer solitary time when I work so that I can accomplish my tasks without interruption or distraction. As you all know, people in an office can be very distracting. There's always an off-topic discussion to focus on, a "Where's lunch today?" question, or some object being tossed between two or more coworkers. I like the interaction, but I also don't want to work an extra two or three hours a day to make up for my lack of productivity because of it.

Technology conferences are an escape from the grind. They give us a chance to connect with people who aren't our coworkers. They provide an outlet for the discussions we often don't get to have in our own jobs. We can ponder the "what ifs" and hypothetical situations together. We can also learn from each other. "Hey, how do you deal with X?" is always a good conversation starter. Maybe it's just me, but I've learned almost as much through lunch table discussions as I have from the breakout sessions. That one-to-one interaction is far more valuable than the cost of the conference. I always return to my cubicle or my home office energized and ready to try new things, improve old things, or interact with my newfound friends.

We need each other. It's in our genetic makeup. We need technology conferences. The predictions of conferences becoming extinct are exaggerated. As long as there are technology jobs, there will be a need for tech conferences. We need each other.

Ken Hess • ADMIN Senior Editor

ADMIN

Network & Security

Features

This issue, we look at how to secure systemd services and its associated components.

10 **systemd-homed**

The Homed service sets up a portable home directory, and FIDO2 or PKCS#11 secure the stored files.

16 **Interview: Lennart Poettering**

We talk to the primary systemd maintainer about the sense and purpose of some systemd features.

20 **Container Security**

Two systemd container management functions - Nspawn and Portabled - allow many programs to run more securely through isolation.

26 **systemd-analyze**

This systemd utility determines the security of your system, letting you track how any service can be secured step-by-step in a sandbox.

News

Find out about the latest plays and toys in the world of information technology.

8 **News**

- WhiteSource releases free Log4j detection tool
- Critical RCE Zero Day vulnerability found in Apache library
- The Linux Foundation to host the Cloud Hypervisor project
- CronRAT malware targets Linux servers

Tools

Save time and simplify your workday with these useful tools for real-world systems administration.

30 **BitLocker and PowerShell**

BitLocker provides current Windows versions a good, closely integrated encryption solution and lets you manage and maintain this feature with PowerShell.

34 **Gatling**

Generate load on servers and services with this load-testing tool.



40 **ThinLinc**

The revival of terminal servers during the COVID-19 crisis depended on the IT administrator to enable home office workplaces, with the help of tools like ThinLinc, a Linux remote desktop server.

43 **Windows Terminal Preview**

The command prompt at the terminal is under active development and comes with a wide range of configuration options.

46 **Processor Affinity Tools**

Get better performance from your nodes by binding processes and associating memory to specific cores.

Containers and Virtualization

Virtual environments are becoming faster, more secure, and easier to set up and use. Check out these tools.

52 **Nutanix Community Edition**

The free Community Edition of the hyperconverged infrastructure is offered alongside its commercial product for those looking to take their first steps in the environment.

58 **VMware Workspace ONE**

We look at the features, components, and architecture of this secure and user-friendly digital workplace.



Management

Use these practical apps to extend, simplify, and automate routine admin tasks.

78 **eSIMs**

eSIM technology opportunities, deployment, and management for the mobile workplace.

80 **Zero Trust**

We look into the principles of zero trust concepts and why zero trust models must replace endpoint security in local networks.



- 43 New features in Windows Terminal Preview**
Interesting features in the latest preview include state-preserving quick access to windows and configuration of settings in a GUI.



- 74 Endpoint Security for Windows 10**
Build 21H1 has numerous protection mechanisms that expand the list of new security features and achieves an improved level of protection.



fedora 35 
Server 64-bit

- Modular
- Easy administration
- Advanced identity management

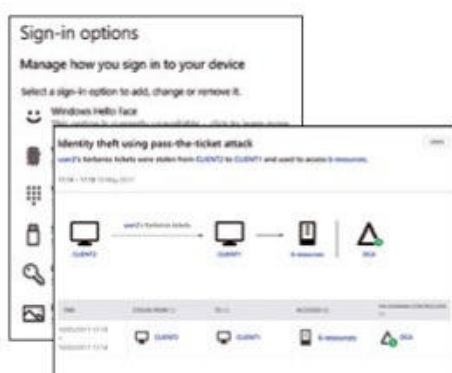
Security

Use these powerful security tools to protect your network and keep intruders in the cold.

- 62 DMARC**
Targeted configuration and the combined abilities of SPF and DKIM safeguard and protect against spam and phishing.
- 64 Nitrokey**
Hardware authentication devices raise data encryption, key management, and user authentication security to the next level.



- 70 MITM Analysis**
Wireshark and a combination of tools comprehensively analyze your security architecture.
- 74 Win 10 Endpoint Security**
We look at on-board protection mechanisms to delay updates and harden the operating system.



Nuts and Bolts

Timely tutorials on fundamental techniques for systems administrators.

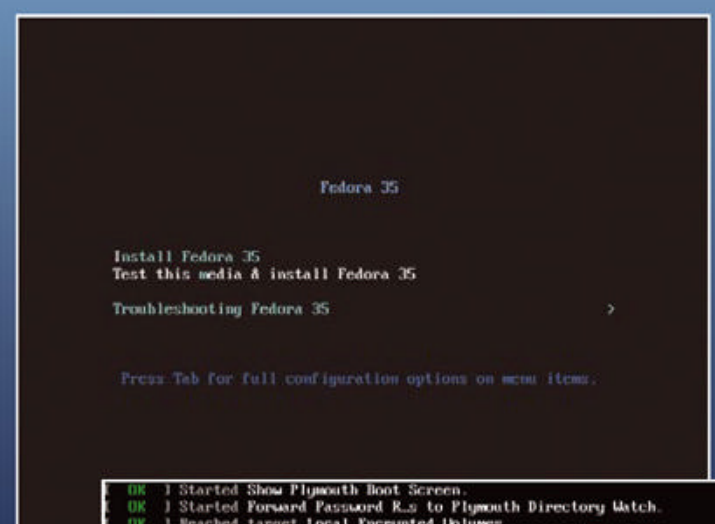
- 86 MITRE ATT&CK & D3FEND**
These knowledge databases provide useful techniques for securing your IT infrastructure.
- 88 Optimizing X Window Displays**
Two command-line tools, xandr and xinput, let you optimize your X Window display from the terminal.
- 90 Rescuing macOS Data**
macOS on-board tools and third-party applications can help prevent the loss of files and make security and backup your first priority.



- 94 Performance Dojo**
Compressed memory solutions for small memory problems.

Service

- 3 Welcome**
- 4 Table of Contents**
- 6 On the DVD**
- 97 Back Issues**
- 98 Call for Papers**



```
OK | Started Show Plymouth Boot Screen.
OK | Started Forward Password R.s. to Plymouth Directory Watch.
OK | Reached target Local Encrypted Volumes.
OK | Reached target Path Units.
OK | Started cancel waiting for multipath siblings of sda.
OK | Finished Wait for udev To Complete Device Initialization.
Starting Device-Mapper Multipath Device Controller...
OK | Started Device-Mapper Multipath Device Controller.
OK | Reached target Preparation for Local File Systems.
OK | Reached target Local File Systems.
OK | Reached target System Initialization.
OK | Reached target Basic System.
OK | Started cancel waiting for multipath siblings of sda.
udev/xr0: 764c3f8d01052dc8dd0e8df6858ec76d
fragment sum: 66443c40fa05651727439dc0ff3143dc983343d1d41fc34bb7427b213e29
fragment count: 28
supported ISO: yes
Press [Esc] to abort check.
Checking: 882.3s_
```

See p 6 for details



Fedora 35 Server (Install)

On the DVD

The **Fedora community delivers** a short-lifecycle server operating system for seasoned system administrators with the latest technologies available in the open source community. Fedora Server boasts:

- Modularity – keep your stacks and software, even when your OS upgrades to a newer version.
- Easy administration – view and monitor system performance and status and deploy and manage container-based services with Cockpit.
- Advanced features – create your enterprise domain with advanced identity management, DNS, certificate services, and Windows domain integration.

```
[ OK ] Started Show Plymouth Boot Screen.
[ OK ] Started Forward Password R...s to Plymouth Directory Watch.
[ OK ] Reached target Local Encrypted Volumes.
[ OK ] Reached target Path Units.
[ OK ] Started cancel waiting for multipath siblings of sda.
[ OK ] Finished Wait for udev To Complete Device Initialization.
       Starting Device-Mapper Multipath Device Controller...
[ OK ] Started Device-Mapper Multipath Device Controller.
[ OK ] Reached target Preparation for Local File Systems.
[ OK ] Reached target Local File Systems.
[ OK ] Reached target System Initialization.
[ OK ] Reached target Basic System.
[ OK ] Started cancel waiting for multipath siblings of
/dev/sr0: 764c3f8d01852dc0dd8e8df6858ec76d
Fragment sums: 66443c48fa85651727d39dc8ff3143dc903349d1d4
Fragment count: 20
Supported ISO: yes
Press [Esc] to abort check.
Checking: 002.3%_
```

Fedora 35

Install Fedora 35
Test this media & install Fedora 35
Troubleshooting Fedora 35 >

Press Tab for full configuration options on menu items.

Automatic boot in 39 seconds...

DEFECTIVE DVD?

Defective discs will be replaced, email: cs@admin-magazine.com

While this *ADMIN* magazine disc has been tested and is to the best of our knowledge free of malicious software and defects, *ADMIN* magazine cannot be held responsible and is not liable for any disruption, loss, or damage to data and computer systems related to the use of this disc.

Resources

- [1] ChangeSet: [\[https://fedoraproject.org/wiki/Releases/35/ChangeSet\]](https://fedoraproject.org/wiki/Releases/35/ChangeSet)
- [2] Distribution-wide changes: [\[https://docs.fedoraproject.org/en-US/fedora/f35/release-notes/sysadmin/Distribution/\]](https://docs.fedoraproject.org/en-US/fedora/f35/release-notes/sysadmin/Distribution/)
- [3] Release notes: [\[https://docs.fedoraproject.org/en-US/fedora/f35/release-notes/\]](https://docs.fedoraproject.org/en-US/fedora/f35/release-notes/)

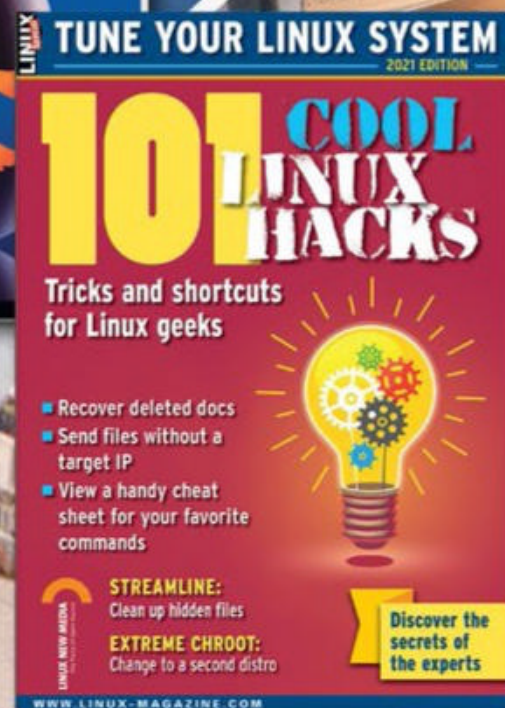
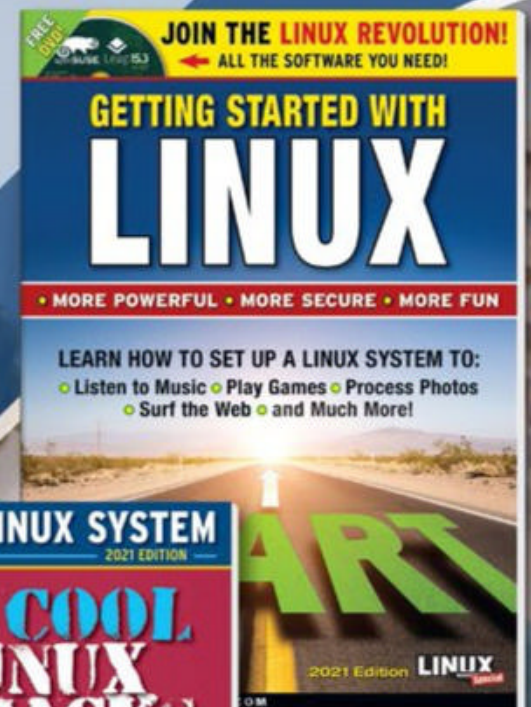
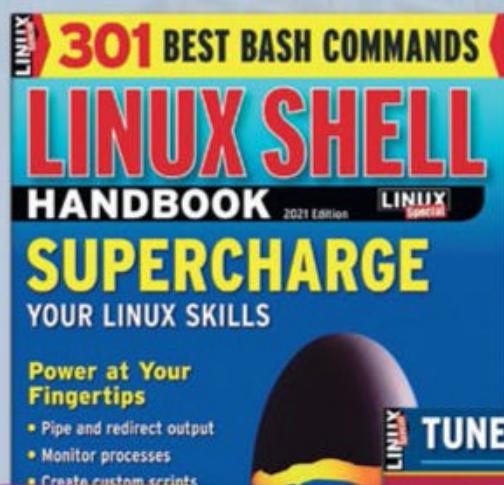
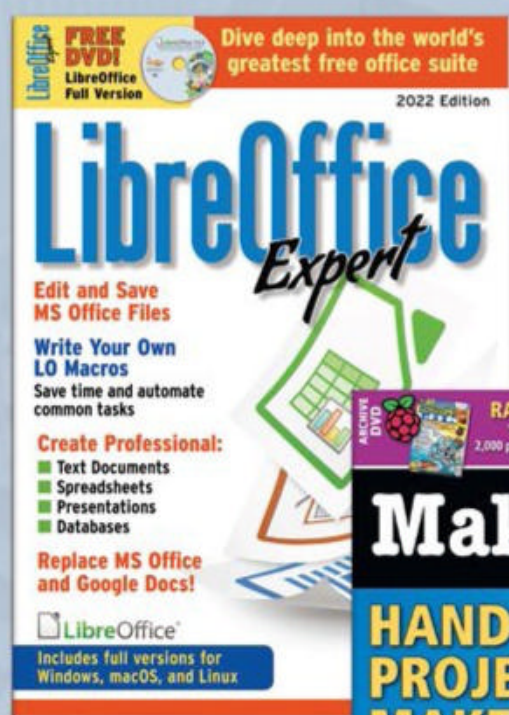
Hone your skills with special editions!

Get to know Shell, LibreOffice, Linux, and more from our Special Edition library.

The *Linux Magazine* team has created a series of single volumes that give you a deep-dive into the topics you want.

Available in print or digital format

Check out the full library!
shop.linuxnewmedia.com



News for Admins

Tech News

WhiteSource Releases Free Log4j Detection Tool

As the Log4j vulnerability continues to wreak havoc on the IT landscape, everyone is trying to prevent disaster from striking. A number of companies and development teams have released tools to help with the detection and remediation of the vulnerability. One such company is WhiteSource. Their new tool, Log4j Detect (<https://github.com/whitesource/log4j-detect-distribution>), is an open source, command-line utility that scans your projects to detect the following known CVEs:

- CVE-2021-45046
- CVE-2021-44228
- CVE-2021-4104
- CVE-2021-45105

Once the scan is complete, it will report back the exact path of the vulnerable files as well as the fixed version you'll need to remediate the issue. Log4j Detect should be run within the root directory of your projects and will also search for vulnerable files with both the .jar and .gem extensions. Log4j Detect supports the Gradle, Maven, and Bundler package managers.

In order for Log4j Detect to run properly, you'll need to install either Gradle (if the project is a Gradle project) or mvn (if the project is a Maven project). The developers have also indicated both maven and bundler projects must be built before scanning. Once you have Log4j Detect installed, the scan can be issued with the command `log4j-detect scan -d PROJECT` (where PROJECT is the directory housing your project).

For more information about this tool, make sure to read through the project README (<https://github.com/whitesource/log4j-detect-distribution/blob/main/README.md>).

Critical RCE Zero Day Vulnerability Found in Apache Library

Chen Zhaojun, from the Alibaba Cloud Security team, recently reported to the Apache Foundation that an Apache library (Log4j) contained a vulnerability that allowed attackers to control log messages and log message parameters and execute arbitrary code loaded from LDAP servers when message substitution is enabled.

This vulnerability (CVE-2021-44228) (<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228>) was found in Log4j2 versions 2.14.1 and earlier and received the maximum possible CVSS score of 10.0.

The Log4j library is in wide use with enterprise Java software, so it's imperative that anyone using this upgrade to Log4j v2.15.0.

John Hammond, a senior security researcher with Huntress, warned, "If your organization uses Apache log4j, you should upgrade to log4j-2.15.0.rc2 immediately. Be sure that your Java instance



**Get the latest
IT and HPC news
in your inbox**

**Subscribe free to
ADMIN Update
and HPC Update
bit.ly/HPC-ADMIN-Update**



is up-to-date; however, it's worth noting that this isn't an across-the-board solution. You may need to wait until your vendors push security updates out for their affected products."

Even printers and CCTV systems are at risk. A new GitHub project (<https://github.com/YfryTchsGD/Log-4jAttackSurface>) has been created to map out potentially affected manufacturers and components.

This vulnerability should not be taken lightly. If you use the Log4j library, make sure you start taking steps immediately to mitigate any risk to your company, your clients, and your data.

The Linux Foundation to Host the Cloud Hypervisor Project

Backed by several powerhouses in the tech industry, The Linux Foundation is set to release a virtual machine monitor specific for modern cloud workloads. The Cloud Hypervisor virtual machine monitor will be written in Rust, focus on security, and will be capable of monitoring CPU, memory, device hot plug, will be able to monitor both Linux and Windows guests with a minimal footprint, and will be able to perform device offload with vhost-user.

The backers of this new platform include Alibaba, ARM, Intel, and Microsoft. According to Arjan van de Ven, fellow at Intel, "Cloud Hypervisor has grown to the point of moving to the neutral governance of The Linux Foundation." He continues, "We created the project to provide a more secure and updated VMM to optimize for modern cloud workloads. With fewer device models and a modern, more secure language, Cloud Hypervisor offers security and performance-optimized for today's cloud needs."

Of the new project, Gerry Liu, senior staff engineer at Alibaba, said, "Cloud Hypervisor is a great innovation project and evolves rapidly. Moving it to Linux Foundation will help to build a stronger community and speed up the adoption."

Find out more from the official Linux Foundation announcement (<https://www.linuxfoundation.org/press-release/linux-foundation-to-host-the-cloud-hypervisor-project-creating-a-performant-lightweight-virtual-machine-monitor-for-modern-cloud-workloads/>).



CronRAT Malware Targets Linux Servers

Security researchers at Sansec (<https://sansec.io/research/cronrat>) have found a new stealth attack that targets Linux servers and uses a nonexistent calendar day to stay off the radar. This Remote Access Trojan (RAT) masks the actions of the attack by using the date February 31 and targets Linux-based web stores to trigger online payment skimmer threats.

The new CronRAT attack can execute fileless malware, launch malware in separate subsystems, control servers disguised as Dropbear SSH services, hide payloads in legitimate cron tasks, and run anti-tampering commands. CronRAT bypasses browser-based security scans and has already been discovered in live online stores. The threat was injected into servers via a Magecart (payment skimming) attack.

This attack is made possible because cron only checks for a date format and not that the date of the task is legitimate. The crontab date specification for CronRAT is 52 23 31 2 3, which would generate a runtime error upon execution. However, that runtime will never happen, because the date doesn't exist.

Once CronRAT is executed, it contacts a Command and Control (C2) server at IP address 47.115.46.167:443 using a fake banner for the Dropbear SSH service. The payloads of the commands are obfuscated with multiple layers of compression and Base64 encoding.

CronRAT is considered a serious threat to Linux e-commerce servers and has managed to bypass most detection algorithms. Sansec had to rewrite its algorithm to catch this dangerous threat.



Portable home directory
with state-of-the-art security

Home, Sweet Home

The systemd Homed service makes it easy to move your home directory, and FIDO2 or PKCS#11 can secure the stored files. By Martin Loschwitz

Your home directory (~) stores personal data and configuration files for the programs you use (e.g., the Google Chrome or Firefox profile and the GTK configuration for the look and feel you prefer for the installed desktop). Moreover, your home directory also contains Thunderbird mail, your music collections, and your photos. Like your own home, the personal folder on a Linux system is typically a place you want to keep safe. If you use more than one Linux computer, you will not find your personal files on all of these devices, and fixing this issue is by no means trivial. The question of what users can do to share their home directory efficiently with a variety of systems is not new. Systemd boss Lennart Poettering finally came up with a solution – Homed – that works on recent systems, relying on systemd in the background and making it possible for users to take an external disk with

their home directories from machine A to machine B and on to machine C without getting into any trouble (**Figure 1**).

The whole systemd-homed setup is highly secure and efficient. Encryption with multifactor authentication is part of the overall package, as is dynamic user account creation. Although Homed does not support encryption and login with the TPM module, PKCS#11 and FIDO2 can be used. In this article, I introduce Homed and look into its technical details. Before that, however, it will not hurt to look at the specific problems that Homed solves – all the more so because it will contribute in a fundamental way to understanding how Homed works.

Attempted Thus Far

For decades, the market has tried solutions from a wide variety of approaches that share your home

directory with a variety of systems. Sometimes shared storage such as NFS or synchronization solutions such as Rsync came into play. However, Rsync requires a functioning network connection between systems. A company laptop that has to be connected to the VPN to get an Internet connection does not meet this condition, and you are inevitably left out in the cold. On the other hand, you could use Samba to resolve the issue, viewing Linux as something similar to a domain client in the Windows universe. None of these approaches really cornered the market. The idea of the portable home directory assumes a few things that are not necessarily commonplace on Linux or POSIX-like operating systems in general. First is the problem of user management. If you are using an account named *Martin* on your system, you will want to use this name elsewhere, too, when you

use your mobile home directory. Of course, you can't expect every Linux system in the world to have a preconfigured user account with a suitable name that is just waiting for someone to plug in an SSD with the appropriate home directory. Instead, you need the user account to be created explicitly.

To begin, a user account must be created on a system. When the system then detects that a mobile home directory exists for the respective user, it integrates it. Even here, systemd faces a challenge because before Homed it did not play a role in the system's user management. Now, however, it has to be able to create users and groups. The whole thing must also work the other way around. Imagine, say, publicly accessible systems that are intended for use by several people with portable home directories. It quickly becomes clear that the system also must be able to delete the user accounts it creates as soon as the user logs off and removes the disk with the storage. User accounts that cannot be disabled are not just useless, they are dangerous. IT history shows that

old, forgotten accounts have been used in attack scenarios.

User IDs

Another factor plays a major role in the dynamic use of home directories: user IDs, although they are closely related to usernames. On Linux systems, the username is effectively only the human-readable variant of the user ID, which is assigned to the respective account at the system level. If you want the system to create a user dynamically after a disk has been plugged into a USB port, the process implicitly creates a user ID. For a user ID (and, in parallel, a group ID), however, the Linux system defines various parameters for each file (e.g., who owns it and who has access to it). This information is stored on the filesystem, which means that it is also stored on the USB stick or SSD that contains the portable home directory. When the user plugs in an SSD, the UID on the system needs to match the contents on the USB stick. If this is not the case, a corrective mechanism is needed; otherwise, access to the files

on the disk would fail permanently because of missing permissions.

Mandatory Encryption

Another must-have that, surprisingly, many users still do not have on their radar to this day is disk encryption. With many PCs and certainly with the vast majority of devices used professionally, the value of the data stored on the machine clearly exceeds that of the hardware. Even a top-of-the-range Thinkpad "only" costs a good \$4,000 (about EUR3,600). However, if the blueprint for an innovative machine or the company's current tax return is stored on the device and it falls into the wrong hands, somebody is in trouble. The damage caused by industrial espionage and damage to the company's reputation can exceed the value of the hardware many times over. Manufacturers have long since recognized this vulnerability. Microsoft, for example, offers BitLocker to encrypt all PC storage devices automatically in the background. Apple does the same with FileVault, and the popular Linux distributions now also rely on comprehensive encryption of storage devices, especially on desktops. An NVMe or SSD drive, of course, must also be encrypted if it contains most of the most information in your life. How can secure encryption of mobile devices be achieved in a meaningful way when the computer is missing? A 64-digit password would provide some security, but it would fail to provide effective protection if no one could remember it. Encryption by means of a certificate or multifactor access control (e.g., FIDO2 in addition to the password) is more useful, preventing data leaks even if the stick and the password fall into the hands of a third party, as long as they don't have the second authentication factor. However, if this kind of technical overhead is required to encrypt the device, then the token you rely on to do so can also be used to log the user onto the system. Clearly, mobile home directories are a great idea, and they seem simple to implement in theory, but if you take an in-depth look at the technical

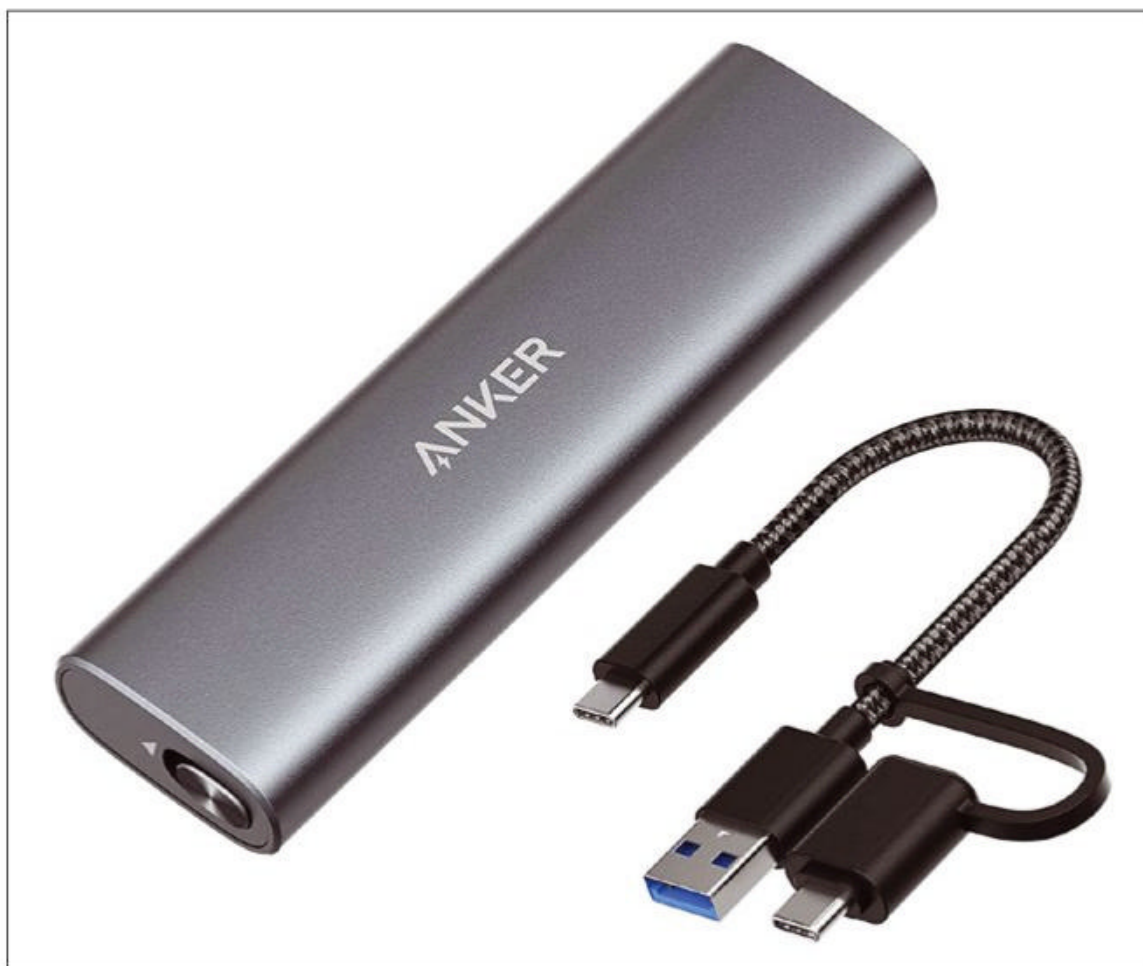


Figure 1: M.2 SSDs (for high-performance storage in power-constrained devices) are so small and light that even large volumes of data can now be moved back and forth with ease, which allows home directories that are not tied to a single system.

challenges, disillusionment quickly sets in. Homed at least claims to address the challenges described above. How does it do this in detail, how does the user benefit from it, and what are the limits of the system?

Getting Started with Homed

The various desktop distributions are sufficiently up-to-date as to include Homed. The situation might not be as easy for more exotic systems such as Raspberry Pi OS. Often older systems such as Debian GNU/Linux “Buster” serve as a foundation, and the Homed version included is outdated and less than satisfactory. Of course, Debian GNU/Linux 11, alias “Bullseye,” has been released in the meantime, so there is hope of updates for these systems in the near future.

A sufficiently recent `systemd` automatically includes Homed; the `homectl` utility [1], which creates users, and the `userdbctl` query tool should be in place on the system. Clearly, `systemd` does not find the username automatically on the basis of plugged-in devices. Theoretically, it would be quite conceivable for Homed to create a suitable user account as soon as a device is plugged in, which would mean total flexibility for the home directory because it would then even be available on public terminals that support Homed. However, the developers have deliberately not taken this approach. Instead, Homed waits for the user to create the account,

```
homectl create <user>
```

for which you need the rights of the root system administrator. Homed, therefore, is primarily designed for use cases in which the user is only sharing their personal directory between several systems over which they have full control.

Handling User IDs

Homed implements the process of creating and deleting user accounts, bypassing the existing system

mechanisms (e.g., `/etc/passwd` and `/etc/group`). Instead, it taps into the dynamic authentication mechanisms with a separate pluggable authentication module (PAM), complementing the existing login system in the process. For the administrator, this means if they want to use Homed on several systems to manage a central home directory, they can influence the user IDs with `homectl` parameters. In this way, you can ensure that the IDs of the users in question match on all participating systems, which is expressly recommended; otherwise, Homed will use a crude hack to keep things tidy. After logging in, Homed simply runs `chown` over the entire home directory and changes its contents to the user ID and group ID that the user has on the system.

Assuming the user ID needs to be 2000 on all systems, the following command creates the user and adds some background information:

```
$ homectl create martin \
  --real-name="Martin Loschwitz" \
  --uid=2000
```

In this case, a user named *martin* is a member of a group with the same name. Unlike before, however, the UID is not randomly selected by Homed but defined manually. Still unsolved is the problem with the user’s home directory. Homed has created this directory and encrypted it with Linux Unified Key Setup (LUKS), but this by no means makes it mobile. Unless the user specifies otherwise, Homed uses LUKS to create an encrypted home directory and a loop device in `/home/<user>.homedir/` and mounts it in `/home/<user>/` after successfully logging in to the system – and only then. This process is basically programmatic with Homed: A user’s personal directory is only accessible while the user is logged in. As soon as the user’s last login session expires, `systemd` automatically unmounts the directory containing the user’s personal data. A new login is mandatory to put it back into operation.

Truly Mobile Home Directories

Anyone who has ever dealt with encrypted volumes on Linux knows that working with LUKS and the like is not necessarily a pleasant experience. Homed takes a large part of the work off your hands by configuring LUKS in the background in line with your specifications – but without forcing them into direct contact with the LUKS tools themselves. Again, it is just a question of the right parameters for `homectl` to avoid loading the user directory created by Homed locally and putting it on a USB stick instead.

From the `homectl` command used before, the following command uses a USB stick to store the home directory:

```
$ homectl create martin \
  --real-name="Martin Loschwitz" \
  --uid=2000 \
  --image-path=/dev/disk/by-id/\
    usb-SanDisk_Ultra-\
    4C530000060908106243-0:0
```

You need to modify the part after `--image-path` if the USB stick is referenced by its unique device ID, as in the example. Homed again takes care of all the administrative work by first deleting all the existing files on the USB stick. Then it creates a partition table and proceeds to create a LUKS-encrypted device. The USB stick is now genuinely portable. If the owner of the directory logs out of the system so that there is no longer a current session, Homed automatically logs off the LUKS device. The user then simply unplugs the USB stick on which it resides and takes it to another device to log in with an account managed by Homed, which recognizes the home directory from the USB stick and automatically enables it on the new system.

PKCS and Tokens

Users created as shown by the examples in this article still do not have passwords, and no alternative login

methods are specified, so they can't log in at all. Fortunately, systemd offers far more options than mere passwords. The team led by Poettering has much to offer, and the biggest hurdle might be to combine the respective authentication device with the appropriate option when creating the user in Homed. For once, however, this is not down to systemd itself but to the

multitude of standards and their options that already exist on the market for this task.

The two best known representatives of crypto keys are probably PKCS#11 and FIDO2. PKCS#11 is a slightly different standard; older YubiKeys (Figure 2) use it, but it will primarily be familiar from classic smartcards (Figure 3). If a smartcard or an older YubiKey is to be used to unlock the account, the biggest challenge is to identify the path

(i.e., the URI) to the device on the system. Homectl at least offers a way out. The command

```
homectl --pkcs11-token-uri=list
```

displays a list of all available devices. In order for the command to find the device, it must be mounted at the time of the call.

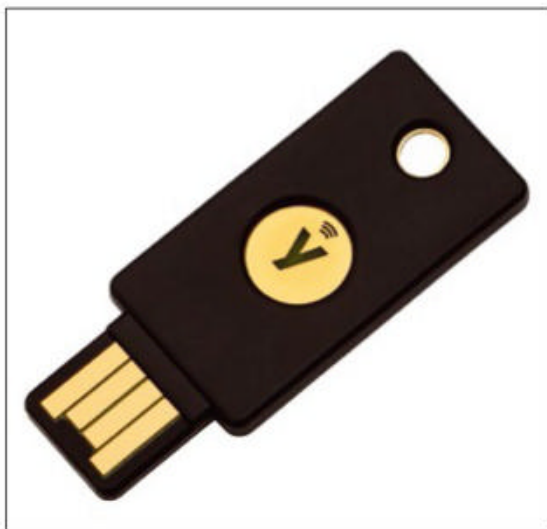


Figure 2: Older YubiKeys use older versions of authentication tokens. © YubiKey

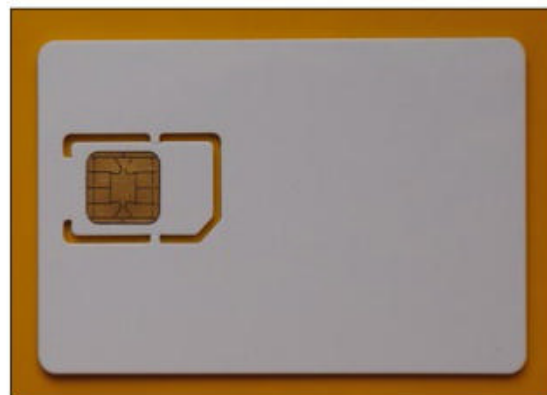
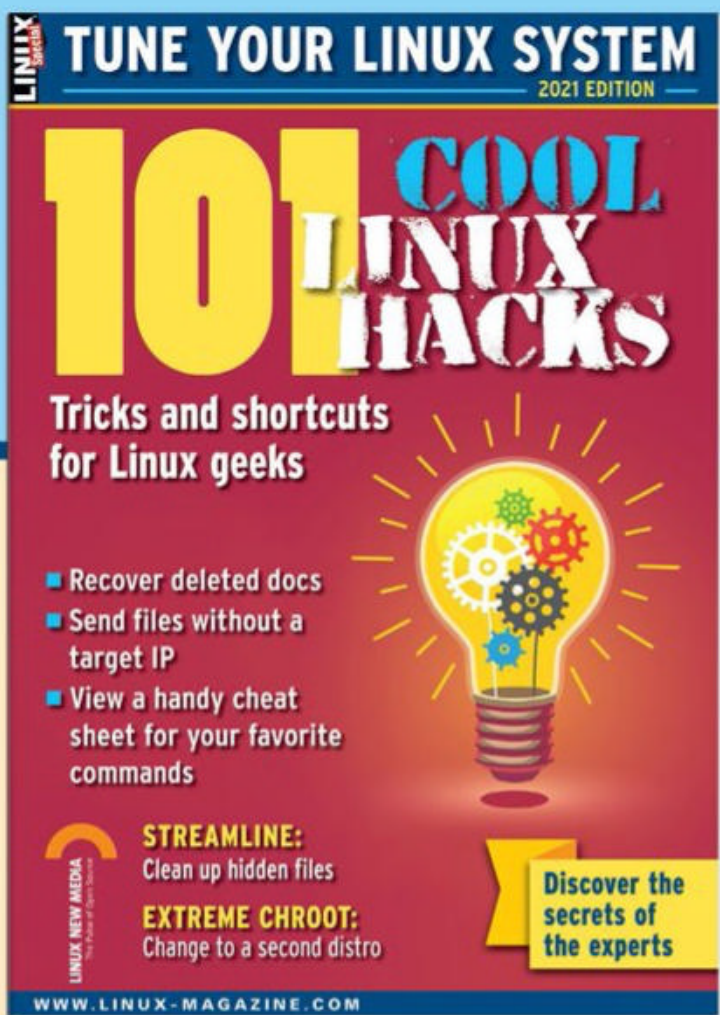


Figure 3: PKCS#11-based authentication usually relies on physical smartcards. © Cardomatic



Figure 4: More modern FIDO2 tokens can also be connected to Homed. If you specify the right parameter when creating the user, the FIDO2 key then unlocks the account, including the home directory. © Feitian



SHOP THE SHOP
shop.linuxnewmedia.com

GET PRODUCTIVE WITH
101 LINUX HACKS

Improve your Linux skills with this cool collection of inspirational tricks and shortcuts for Linux geeks.

- Undelete lost files
- Cure the caps lock disease
- Run C one-liners in the shell
- Disable your webcam and mic
- And more!



ORDER ONLINE: shop.linuxnewmedia.com/specials

If you use an authenticator according to the FIDO2 standard instead (**Figure 4**), you need the `--fido2-device=` parameter, which also supports the `list` keyword, which brings to light a list of available devices along with their URIs on the system. Additionally, the `auto` keyword works if only one device fits the bill. The entire call is then:

```
$ homectl create martin 2
--real-name="Martin Loschwitz" 2
--uid=2000 2
--image-path=/dev/disk/by-id/2
usb-SanDisk_Ultra_4C53000002
60908106243-0:0 2
--fido2-device=auto
```

The user created in this way has a personal directory on a USB stick and logs on to the system with an authentication device.

PKCS#11 Special Case

If you opt for PKCS as your login method, you definitely need to remember that the YubiKey has to be set up before the Homed configuration. The `ykman` commands

```
ykman piv reset
ykman piv generate-key 2
-m RSA4096:*9d pubkey.pem
ykman piv generate-certificate 2
--subject "Homed" 9d pubkey.pem
```

delete old keys from the device, create a new key, generate the certificate required for the key, and load it onto the YubiKey. Then,

```
rm pubkey.pem
```

removes the key file from the filesystem. After that, the login can be configured as described earlier.

Playing It Safe

The `homectl` command also supports the `--recovery-key` parameter. If you have ever dealt extensively with cryptography, you will be aware

that if you lose the device for generating tokens or the original key, you can't access the data, no matter what you try. Good practice dictates generating an emergency key with the device and keeping it in a safe place. It is essential to protect the key against access by unauthorized persons because anyone who has the key can decrypt the encrypted volume; therefore, you should keep the key as a hard copy on paper in a safe place (e.g., a safe).

By appending the parameter `--recovery-key=yes` to the command for creating the user, you can make sure that Homed automatically creates a suitable key. The key is then displayed onscreen and can be copied from there.

Retroactive User Changes

Regularly, not all the parameters you need are configured when a user is first created in Homed. For example, if you don't have a YubiKey or smart-card when you create a user, you won't be able to use them. The good news is that Homed lets you add details such as decryption devices and modify the details of an account after the event with the `homectl update` command.

If you want to enable PKCS#11-based or FIDO2-based authentication for an account, you can use either of:

```
homectl update martin 2
--pkcs11-token-uri=auto
homectl update martin 2
--fido2-device=auto
```

Because the commands and parameters are the same as when setting up the user, you avoid the need to learn new parameters.

Limits

Homed takes the promise of the mobile home directory quite seriously and implements it sensibly. Despite all the euphoria about the technology, users and admins should not forget

that the principle is subject to technical limitations that even Homed cannot define away.

The most relevant limitation here is by no means on the Homed level but relates to the applications you use with your portable directory. The home directory will fill up with garbage relatively quickly if you use it on different systems with different distributions because the configuration files will then contain competing entries. For example, if you use Ubuntu 18.04 on one system and Ubuntu 21.04 on the other, you will find different KDE versions on the two distributions. If you plug the home directory of the KDE version from Ubuntu 18.04 into the computer with Ubuntu 21.04, KDE will find the old configuration files and convert them accordingly. However, the return route is blocked: KDE on Ubuntu 18.04 cannot understand the new configuration and, in the worst case, will create a completely new one. The problems become even more obvious when you imagine different systems or distributions. A home directory from openSUSE Leap is unlikely to harmonize with Raspbian as used on a Raspberry Pi.

If you want to avoid compatibility problems, you need to take manual steps to prevent some files ending up in your home directory, which has the unpleasant side effect that you then have to configure your own desktop again on every system you use. Alternatively, you can take care to use the shared home directory only on systems that are mutually compatible in the broadest sense. ■

Info

[1] `homectl`: [\[https://www.freedesktop.org/software/systemd/man/homectl.html\]](https://www.freedesktop.org/software/systemd/man/homectl.html)

The Author

Freelance journalist Martin Gerhard Loschwitz focuses primarily on topics such as OpenStack, Kubernetes, and Ceph.



CLOUDFEST

**WE ARE BACK,
EXCITED AND
PUMPED.**

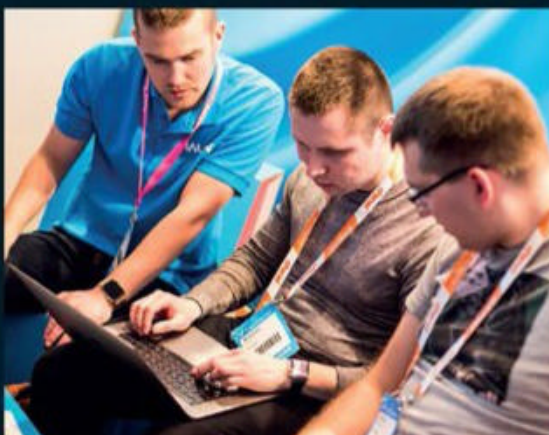
📅 March 22 - 24, 2022 | 📍 Europa-Park, Germany

**REGISTER NOW
AND SAVE € 399!**
WITH FREE CODE: **We-Love-Admin**



REG.CLOUDFEST.COM

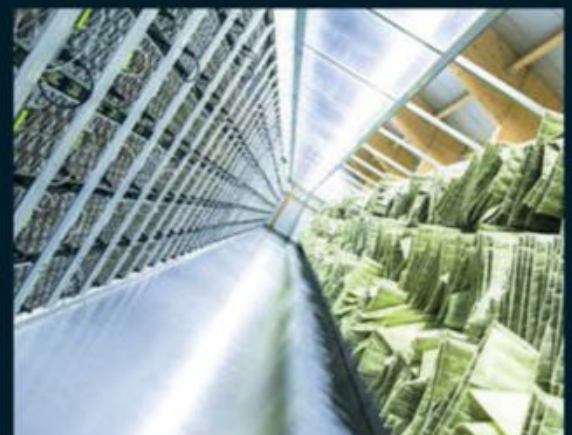
THEMES FOR 2022



**THE INTELLIGENT
EDGE**



**OUR NEW
DIGITAL WORLD**



**THE SUSTAINABLE
CLOUD**

cloudfest.com

 /cloudfest

 @cloudfest



CC-BY-SA-3.0 [1]

The achievements of and plans for systemd

Extending Integration

We talked to systemd maintainer Lennart Poettering about the sense and purpose of some systemd features. By Jens-Christoph Brendel

Linux Magazine: *If you take stock of the last three or four years, what have been the most important innovations in systemd during this time?*

Lennart Poettering: That would be, firstly, all the security features we have added and made visible with the `systemd-analyze security` tool. Regular system services can now be locked into effective sandboxes with relative ease, but can still be integral parts of the host operating system. I believe this has advanced Linux system security quite a bit. Another important innovation might be `systemd-tmpfiles` and `systemd--sysusers`. Strictly speaking, they are more than four or five years old, but it is only in the last three or four years that they have finally seen more widespread use in the popular distributions. We are looking to move to a declarative description of the system and its components, leaving behind imperative scriptlets in packages and the like. This improves robustness, security, and reproducibility. The dynamic user strategy makes it possible to allocate system users dynamically when starting system services that are automatically released again when the service terminates. This takes into account that system

users are *the* original mechanism used to implement privilege separation on Unix and Linux. No matter which subsystem you look at, access control based on users is always implemented on Linux. Other concepts – such as SELinux labels, Access Control Lists (ACLs), other Mandatory Access Controls (MACs), and so on – are not universally available and are nowhere near as popular or as universally well understood.

Classically, however, such system users are expensive, with only 1,000 of them (or sometimes only 100 or 500, depending on the distribution), and they are allocated individually during package installation. So traditionally they can only be used roughly to secure large services but not to protect individual instances or transactions. There are simply too few of them for that. The dynamic user concept solves the dilemma: It makes cheap what was previously expensive. Dynamic users can be allocated for a short time and returned after use. This practically breathes new life into an old Unix strategy and is a mechanism that can definitely contribute a great deal to further improving the security of Linux systems.

Last but not least is `systemd-homed`, a really secure home directory

management system where the user password is the encryption key. I could continue this list for a long, long time – after all, there are so many useful new features in systemd. If you want to know more, take a look at the `NEWS` file in the systemd sources, which is where we write everything down in more detail, while hopefully keeping things reasonably understandable.

Maybe a word about one last set of innovations: We recently added support for FIDO2, PKCS#11, and TPM2 security chips to systemd for disk encryption or user authentication. For the first time, this makes it possible to set up truly secure systems on Linux with practically on-board tools, without getting lost in massive manual scripting sessions or reducing security to passwords.

LM: *What else is on the wish list for the near future?*

LP: Many people working on systemd have different interests. Personally, I have a great interest in simply making Linux even more secure, and, by that, I mean the classic, generic Linux distribution.

It hurts a bit that other operating systems like macOS or Windows currently protect user credentials better

than we commonly do on Linux with our home directories. Even the non-traditional Linux systems like Chrome OS or Android are generally far better secured than classic Linux distributions because they detect and prevent offline modifications of the system, for example. Applications also run on them in relatively secure sandboxes by default. None of this really exists on classic Linux so far. There are projects in this vein, but only a few of them have reached the “mainstream” of Linux distributions so far.

This is exactly where I hope to improve the situation. The basic infrastructure is certainly provided by most distributions, but there is a lack of integration, of connecting the various subsystems to make them useful, which is exactly what the support for TPM2/PKCS#11/FIDO2, mentioned earlier, is aimed at. The subsystems for the respective technologies have existed for a long time, but few specialists actually use them together because the required integration with the rest of the operating system just never happened or was incomplete. I see systemd as the project that can do precisely that in a good way – determining where the journey should go and then integrating the subsystems needed to get there. For example, tying disk encryption to TPM2/PKCS#11/FIDO2 fits right into this scheme, but there is far more to be done in this area.

Thus, while many – possibly even most – users use disk encryption on Linux, typically typing the disk password does not protect the program code very well, which leaves you helplessly exposed to an evil maid attack, an offline attack in which someone simply exchanges the boot code of the system being attacked. You have no way to tell whether the cryptsetup binary to which you give your password is really the one you trust or perhaps a hijacked one that immediately sends the password to an attacker.

Other operating systems are doing much better, including Linux-based

ones. I’d like to see us catch up there with generic Linux distributions so that the data on our laptops remains at least as secure in every way as, say, on a Chrome OS system. It’s downright embarrassing that this is not yet the case. We need to do better, especially in this age of Pegasus and similar systematic security threats. I think systemd can and should play a certain role in making generic distributions more secure: more TPM2, meaningful secure boot, more sandboxing, more encryption, more integrity – and all without really demanding more knowledge from the admin.

Another related topic in this context is Rust: Sooner or later we should move away from C. It’s just too hard to use the language correctly, and even the best developers make mistakes all the time. Rust is probably the first language that has a chance to replace C on a broad front. For systemd, that means we have to figure out how to make the transition as developers. We don’t want to be pioneers but instead wait for other projects to solve the most pressing problems for us before we make the leap ourselves. After all, for us, a programming language is just a tool, not a purpose in itself.

LM: *One of the systemd goals was to accelerate and standardize boot sequences. This goal can be considered achieved today with most distributions relying on systemd, but did this not happen at the cost of a far larger number of systems being affected in the case of security-relevant errors than the case would be with more diversity?*

LP: Acceleration was never the primary goal of systemd development but simply a side effect of the work to implement the boot process in a reasonably state-of-the-art way. We have emphasized this time and time again. We always try to find the balance between having a manageable, modular system while booting quickly and in a reasonably straightforward way. If in doubt,

however, we have always opted for correctness and manageability.

By way of an example, we work a lot with small files in drop-in directories, such as unit files located in `/usr/lib/systemd/system/*`. These support modularity, so package managers can easily and elegantly add and remove components from the operating system. In terms of boot speed, this is more of a disadvantage: If we packed the service descriptions into a single large file instead of many small ones, they could almost certainly be read many times faster, but then nothing would be modular. However, modularity is more important to us than plain speed at boot time, so we went for drop-in files anyway.

I don’t think it’s a good idea to balance supposed speed advantages against security gains from more init system alternatives – they have nothing to do with each other. Sure, it would be good if there were convincing Linux init system alternatives to systemd – competition stimulates business, monoculture is not ideal – but I still believe that the very best thing for more computer security is better technical security strategies: more sandboxing, lockdown, integrity checks, and so on. You certainly don’t do general computer security any favors by continuing to maintain multiple init systems that offer no security strategy at all. However, if we give systemd security features that are then widely used, that’s worth far more at the end of the day.

To put it another way, I find it far more interesting to make one class of attacks completely impossible than to hope that “only” one half of the Linux world is vulnerable to it because the other half uses a slightly different system. On top of this, sure, there is definitely some interesting competition driving the security of computer systems, such as Chrome OS, Android, Windows, macOS, and so on. For us, this is highly relevant inspiration.

I would like to set one thing straight: Thus far, systemd actually looks very

good when it comes to code quality and vulnerabilities. We have significantly fewer CVEs or the like (admittedly not a good metric) than other projects with similar numbers of lines of code. It should also be remembered that projects such as `wpa_supplicant` weigh in with more lines of code than `systemd` (even the kernel has many times that), so with all the components that come with `systemd`, the `init` system is not exactly the primary component to worry about. The attack surface of the WiFi stack or the kernel turns out to be far larger, so a monoculture there certainly causes bigger problems.

LM: *Originally, systemd was intended as a replacement for the SysVinit system. In the meantime, however, it manages all kinds of resources, including its own out-of-memory (OOM) killer. In 2018, Facebook already came out with OOMd developed in-house as a competitor to the implementation in the kernel. What makes the systemd version better than the two predecessors?*

LP: The `systemd-oomd` service integrated into `systemd` was programmed by Facebook developers. It is a simplified evolution of the old separate `OOMd`. `Systemd` manages system services – that is its very specific task. Two facets of this management are lifecycle management and resource control (i.e., correct and clean startup and shutdown of services at the right times and the allocation of resources and their limits). An OOM service directly intervenes in exactly these two parts. Depending on individually configurable parameters, it shuts down services as needed. This works best when the OOM service and `systemd` agree on what to do. That's why we integrate strategies: `systemd-oomd` can analyze the system and become active; the `systemd` service manager knows about it and informs the administrator correctly. Additionally, the following applies here: We always add components to `systemd` when we assume that the

service will ultimately benefit a significant majority of users. This should also be the case with `systemd-oomd`. To use available resources in the best possible way under load, you need a service like `OOMd`. Unlike, say, the OOM killer in the kernel, it keeps an eye on the whole system. It tries everything to handle resource bottlenecks and the resulting latencies as locally as possible and not to affect the whole system. This is needed to utilize thick servers as fully as possible but also to achieve maximum performance in embedded systems with few resources – and helps on the desktop, as well. For the first time, you can no longer freeze your laptop with `make -j` on the wrong build tree.

LM: *How does systemd fit into a world where applications are increasingly no longer launched directly from the operating system but in the form of containers?*

LP: Here, too, you need an underlying operating system. The container strategy is (among other aspects) primarily about isolation from the host OS. However, extensive isolation from the host operating system is neither helpful nor possible for many applications. A service that makes extensive use of hardware can only be run in a container if you rely on hacks and workarounds. Containers are without question very useful but are more for payloads than for system components. For the latter, you need infrastructure like that provided by `systemd`. The sandboxing offered by `systemd` for system services is ultimately inspired by container strategies, but it takes into account that complete isolation (e.g., a complete directory tree of its own) is more of a hindrance for system services. Therefore, it allows for far more modular sandboxing that tries to support integration, while still minimizing the attack surface for hacks as much as possible.

LM: *What distinguishes containers launched by Nspawn from portable services?*

LP: The `systemd-nspawn` tool is versatile and so are portable services. Where one makes more sense than the other is not always clearly defined. Basically, though, I would say `systemd-nspawn` is about working in a similar way to lightweight VMs (virtual machines). For example, with `Nspawn`, a more or less complete Linux can be booted without any overhead, almost like in a VM. Portable services is more about making individual system services a bit more portable (i.e., making it easier to move relatively integrated system services between machines). You could also say that the first program that runs as a payload in a VM is the operating system kernel. In an `Nspawn` container this is an OS `init` system instead, whereas in a portable service it is the main program of a service. The latter may resemble a Docker container, but Docker containers tend to run isolated from the host OS, which is not so much the case for portable services.

LM: *Kubernetes and its offshoots such as OpenShift have become widely accepted for container management. What niche can systemd-nspawn best serve?*

LP: `systemd-nspawn` can run containers, whereas `Kubernetes` orchestrates containers in clusters – two very different tasks. `Kubernetes` normally uses a tool like `runc` to run the containers. If you want, you could use `systemd-nspawn` instead of `runc` to do this; the infrastructure would lack very little. For example, `systemd-nspawn` already has direct support for running OCI containers onboard. I personally have certain doubts about the `Kubernetes` approach. It seems to me that a lot of things have not been thought through to the end but glommed together with hot glue. That's why I haven't done anything yet to make `systemd-nspawn` usable as a back end for `Kubernetes`. I think such an approach would have advantages in terms of security and especially resource control.

Basically, however, Docker-style containers usually only run individual services in them, not the entire operating system. As mentioned before, the focus of systemd-nspawn is more on the latter. We want to make it easy to run full Linux userspaces in them, much like in a VM or on a physical system. So, the focus of systemd-nspawn is a bit different from runc and Kubernetes.

LM: You propose migratable home directories that bring the user account information right along with them. Does that only work if the user mounts their home directory on their own host? Who else would create such a directory on a portable medium? In the conventional system, the write protection for /etc/passwd ensures that a user cannot add their account to arbitrary groups, for example. If this information is located

directly in the home directory, the user should not be allowed to edit it there. Who can do this if the directory is to be mounted on arbitrary hosts?

LP: Typically, home directories are still located on your own laptop's hard drive, but if you let systemd-homed manage them, you can also put them on, say, a USB stick and move them safely back and forth between different systems. I'm sure some users will find this helpful, but it's more of a side effect of the design and not the goal. I myself use systemd-homed to manage my home directory, but I just store it on my laptop's SSD.

The user records that systemd-homed manages are cryptographically signed, and the daemon only accepts records that match the local machine. This signing and

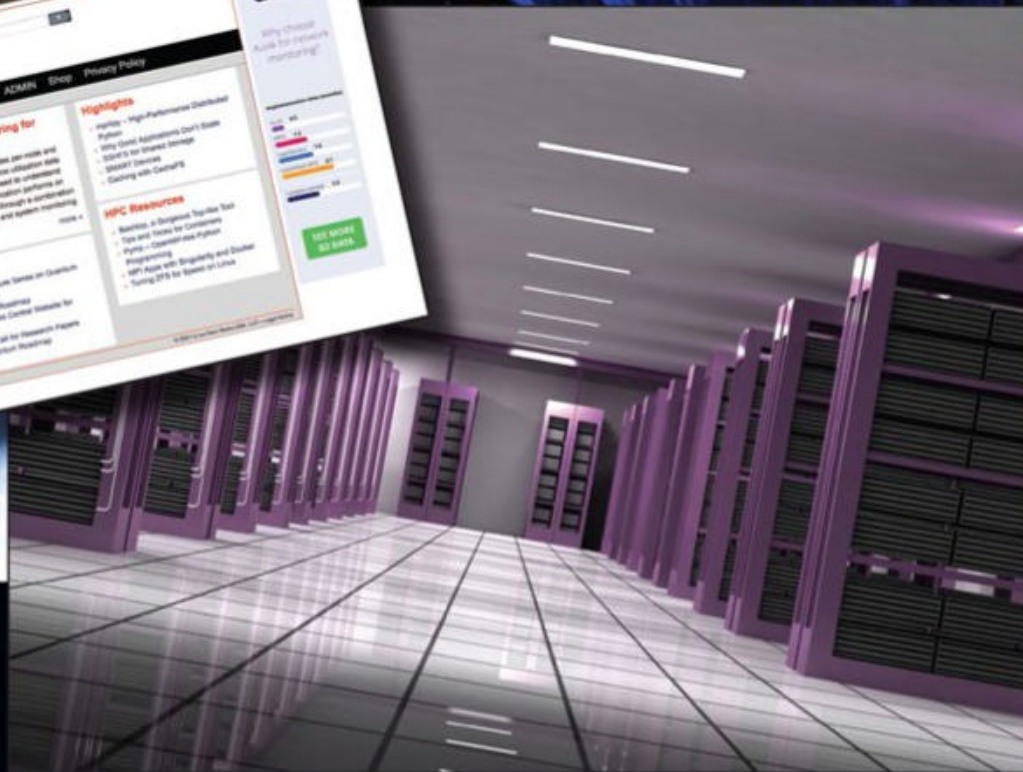
verification takes place completely automatically, without the user having to do it manually. This means two things: First, users cannot easily modify their own user records unless they know the system's secret key, which is protected under /var and should therefore only be known to the system and root. Second, when moving a home directory from one system to another, you have to make sure that the signature key of the first machine is also accepted on the second machine, which can be done by a simple scp.

LM: Mr. Poettering, thank you very much for the insightful interview. ■

Info

[1] Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0): [<https://creativecommons.org/licenses/by-sa/3.0/>]

A Webzine for High-Performance Computing Specialists



ADMIN
Network & Security

If you work with high-performance clusters, or if you're ready to expand your skill set with how-to articles, news, and technical reports on HPC technology.

admin-magazine.com/HPC



Create secure simple containers with the systemd tools Nspawnd and Portabled

Isolation Ward

Systemd comes with two functions for container management that allow many programs to run more securely through isolation. By Martin Loschwitz

The debate surrounding systemd, originally launched with the simple goal of replacing the ancient SysVinit scripts in most Linux distributions with a contemporary solution, has caused even venerable projects like Debian GNU/Linux to split into a pro-systemd faction (Debian) and an anti-systemd faction (Devuan). However you look at it, though, success has proved systemd originator Lennart Poettering right. No major distribution today would seriously consider replacing systemd with another solution. The init system's

relevance is dwindling in any case in the age of containerized applications. If MariaDB is just a container you need to launch, then the init system hardly needs to perform any magic. If you follow Red Hat, SUSE, and its offspring, clearly containers is where the journey is headed (see the "Container Advantages" box). A container-first principle now applies to all enterprise distributions, with the exception of Debian. Systemd has a few aces up its sleeve that most admins don't even know about – not least because of the sometimes almost hysterical

controversies surrounding the product. In the container context, these functions include Nspawnd and Portabled. When deployed correctly, they draw on features from the container world to make conventional applications more secure. If you use Nspawnd wisely, you could even save yourself the trouble of needing Docker or Podman. In this article, I provide an introduction to these two functions and explain how you can use the solutions to supplement your own setups.

Unknown Container Runtime

When asked about runtime environments for containers, most admins intuitively think of one of two candidates: Docker or Podman. Docker returned containers on Linux to the land of the living and provided a decent business model. That containers are considered commercially attractive at all today is largely thanks to Docker's persistent work. Podman, on the other hand, is known by most admins as the anti-Docker solution created by Red Hat that exists because the Docker developers once tangled with the crimson chapeau and, as expected, got the wrong end of the stick. Because Podman is meant to work as a one-to-one replacement for Docker. However, it adopts much of its

Container Advantages

From the point of view of both vendors and software producers, containers are convenient, with the distribution only having to provide a few components: a kernel and a runtime environment. The software provider, in turn, also only needs one container in their portfolio because it runs on basically every system with a functional container runtime. Where Red Hat and its associated distros used to have to maintain different versions of MariaDB, PostgreSQL, and practically all the relevant tools for their own distributions, today they only provide a shell and a kernel. The provider of the software itself steps into

the breach and offers precisely one container that runs everywhere. Brave new world – and so elegant.

As great as this hip stuff may be, the inventory of current IT environments will remain around for a while yet, as well as the question of how this inventory can be used and managed more sensibly and in a better way. What is particularly annoying is that conventional environments do not benefit from the many advantages that containers undoubtedly offer, such as the separation of permissions, isolated access to your filesystems, and monitored network traffic.

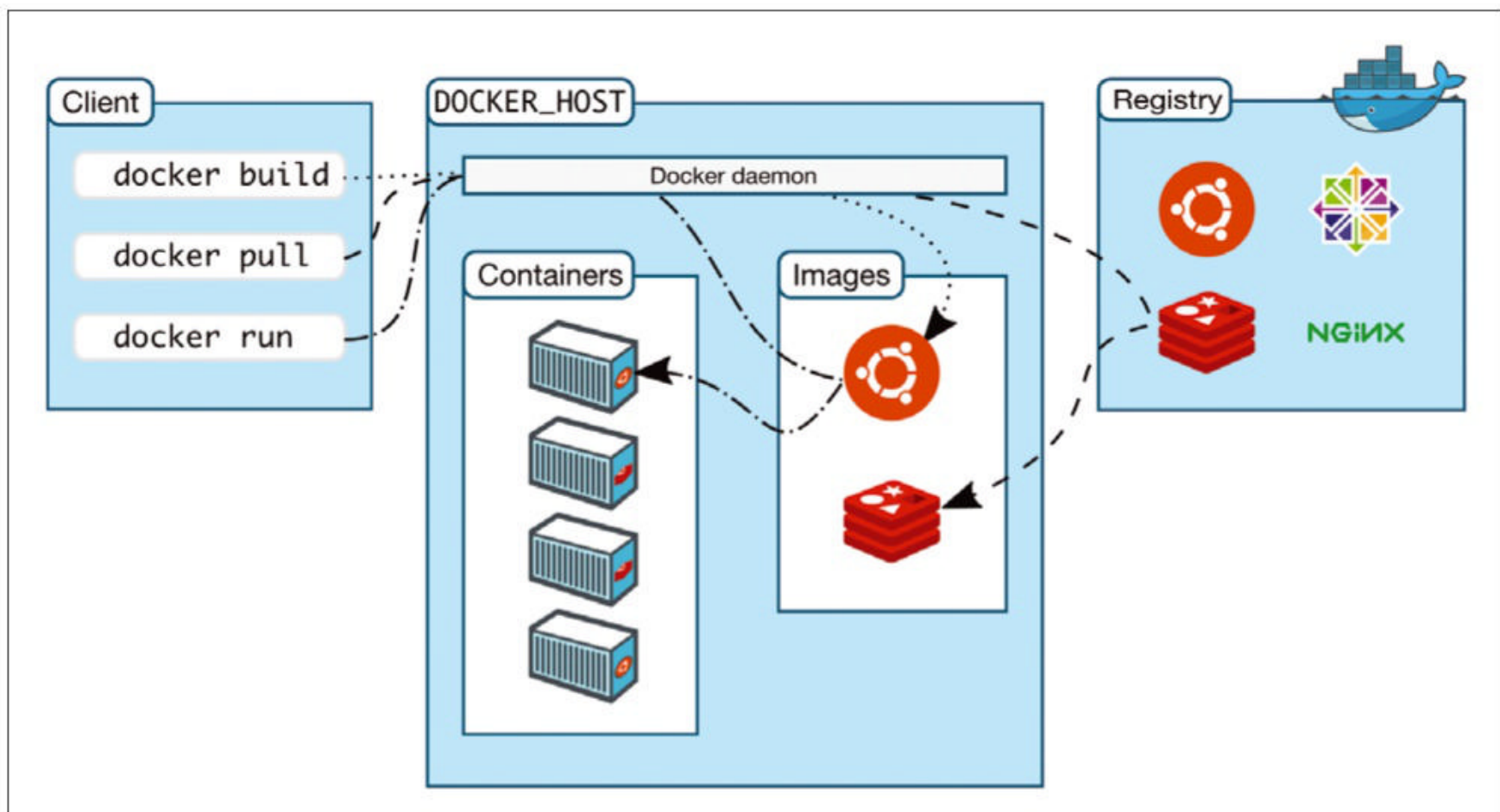


Figure 1: Docker comprises a multitude of services and components. If you only need simple protection, you could quickly feel overwhelmed by the features (credit: Docker docs [1]).

architectural assumptions, and they're tough, because the Docker notion of containers is complex (Figure 1) and can overwhelm you with feature bloat. Containers should be simple. All container implementations on the market ultimately rely on a relatively small set of security features that the Linux kernel itself has offered for a few years.

No container implementation can do without namespaces, which logically separate individual parts of the system (Figure 2). A network namespace, for example, lets you create virtual network cards without giving them direct access to the physical NICs of the host. Instead, this access must be established

by a bridge or some other means. Namespaces do not only exist for network stacks; they also apply to individual points in the filesystem, to process IDs, and to the assignment of user IDs on a Linux system. They always work along the same principle: As soon as a certain process starts in a namespace, the namespace acts like a jail from which it is impossible to break out.

Control groups (cgroups) are added on top in many container environments. Again, they are deeply embedded in the Linux kernel. In very simplified terms, cgroups control access by individual processes to the system's resources. They complement namespaces nicely because they help you enforce an even tighter set of rules for applications and processes than would be possible with namespaces alone.

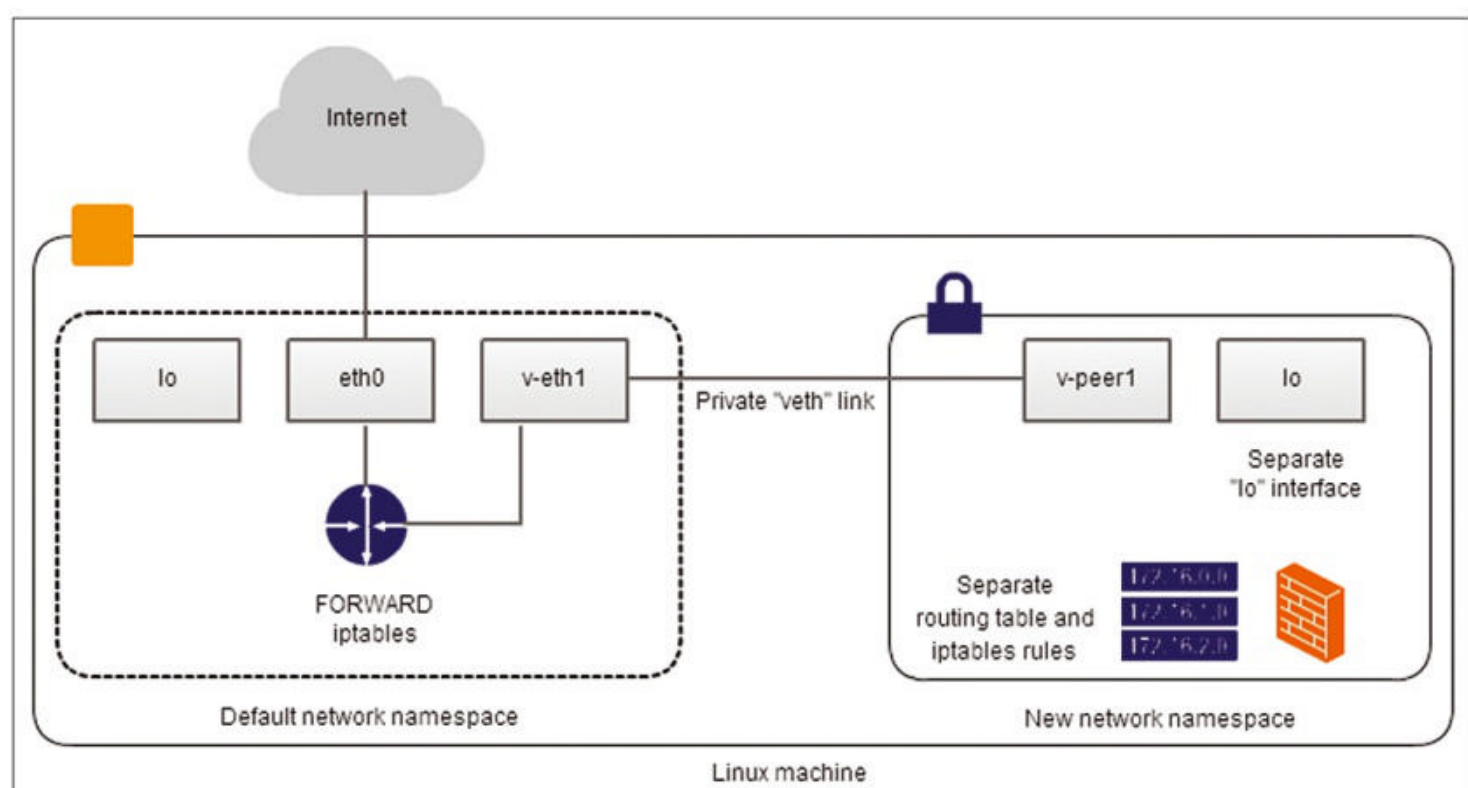


Figure 2: The kernel namespaces feature has many uses in the context of containers, allowing areas in a virtual system to be isolated from the main system (credit: Ivan Zahariev [2]).

More than Runtimes

If Nspawnd is a runtime environment for containers, yet at least two well-functioning environments already exist in the form of Docker and Podman, why, some might ask, does Poettering have his fingers in the pie again? The answer to this question is stunningly simple: `systemd-nspawn` targets admins who really only want to use basic kernel features to isolate individual processes.

The problem with Podman and Docker, after all, is that you never just get the program in question. Instead, they come with a huge pile of assumptions and prerequisites about how to run a container well and sensibly. You might not even want to deal with things like volumes, software-defined networking, and other stuff if all you want to do is put an Apache process in a virtual jail. Also, you might not want to install dozens of megabytes of additional software for Docker or Podman, thereby raising the maintenance overhead, although this step is not strictly necessary from a functional point of view. Anyone who can see themselves in this scenario – simple containers that use built-in

tools without too much tinsel – is a candidate for the target group that Nspawnd has in mind, even if the scope of Nspawnd has naturally expanded in recent years.

The daemon has been part of `systemd` since 2015, so it's an old acquaintance. The “N” in the name – you probably guessed it after following the article up to this point – stands for “namespaces.” Reduced to the essential facts, Nspawnd is a tool that sets up the namespaces required for isolated operation of applications and then starts the applications. Some developers jokingly refer to it as “Chroot on steroids,” which works well as a metaphor. In the context of concrete technology, however, the comparison is misleading.

Containers, Pronto!

Nspawnd is now included in most distributions, so a container can be created on a normal Linux system in next to no time. Creating a usable template takes longest; in Docker or Podman parlance, this would be referred to as an image. Nspawnd only requires a working filesystem on a Linux distribution. You can put this in place in different ways.

The following example assumes Debian GNU/Linux 11 alias “Bullseye” as the distribution used in the container. In the first step, you build an empty folder on a Debian system after installing the *debootstrap* package (Figure 3), which contains a basic Bullseye system:

```
# debootstrap --arch amd64 bullseye /mnt/containers/bullseye-1 https://debian.inf.tu-dresden.de/debian/
```

To log in to the container as root, `pts/0` must be in `/etc/securetty`:

```
echo "pts/0" >> /mnt/containers/bullseye-1/etc/securetty
```

If you now want to start a running container from the directory you just created, type:

```
systemd-nspawn -D /mnt/containers/bullseye-1
```

You can now run `passwd` to change the password for root in the container or add new users. All other commands that you will be familiar with from a normal Debian system are available to you. The recommendation is to store central files such as the package

sources in the template and to update the package sources in the template immediately by running `apt update`. You need to delete the `/etc/hostname` file in the template so that the container uses the name assigned by Nspawnd later. Finally, D-Bus needs to be installed in the container because the `machinectl` userland tool (Figure 4), which you use to control the containers from the

```
Terminal - mloschwitz@debian: ~
File Edit View Terminal Tabs Help
mloschwitz@debian:~$ sudo debootstrap --arch amd64 bullseye /mnt/containers/bullseye-1 https://debian.inf.tu-dresden.de/debian/
I: Target architecture can be executed
I: Retrieving InRelease
I: Checking Release signature
I: Valid Release signature (key id A4285295FC7B1A81600062A9605C66F00D6C9793)
I: Retrieving Packages
I: Validating Packages
I: Resolving dependencies of required packages...
I: Resolving dependencies of base packages...
I: Checking component main on https://debian.inf.tu-dresden.de/debian...
I: Retrieving libacl1 2.2.53-10
I: Validating libacl1 2.2.53-10
I: Retrieving adduser 3.118
I: Validating adduser 3.118
I: Retrieving libapparmor1 2.13.6-10
I: Validating libapparmor1 2.13.6-10
I: Retrieving apt 2.2.4
I: Validating apt 2.2.4
I: Retrieving apt-utils 2.2.4
I: Validating apt-utils 2.2.4
I: Retrieving libapt-pkg6.0 2.2.4
I: Validating libapt-pkg6.0 2.2.4
I: Retrieving libargon2-1 0-20171227-0.2
I: Validating libargon2-1 0-20171227-0.2
I: Retrieving libattr1 1:2.4.48-6
I: Validating libattr1 1:2.4.48-6
I: Retrieving libaudit-common 1:3.0-2
I: Validating libaudit-common 1:3.0-2
I: Retrieving libaudit1 1:3.0-2
```

Figure 3: A container suitable for running in Nspawnd or Portabled is quickly built with typical Debian tools like *debootstrap*.

host, cannot communicate with the particular container otherwise. Once the template is created, copy it to a location with a suitable name (e.g., `/var/lib/machines/webserver-1`). The next step is to start the container with:

```
systemd-nspawn -M webserver-1
-b -D webserver-1
```

After that, you can install the user-land software that you want to run in the container (e.g., an Apache web server).

Unit Automation

If you want to wake up the container automatically at system startup, you can do so with a systemd unit file (**Listing 1**), where you can also configure the network for the container. Systemd basically offers shared networking over a bridge or with a variety of other options. The shared variant, however, is most convenient if it is only a matter of passing through individual ports. The file from **Listing 1** is a ready-made unit file for a Bullseye web server container that Nspawn starts at boot time.

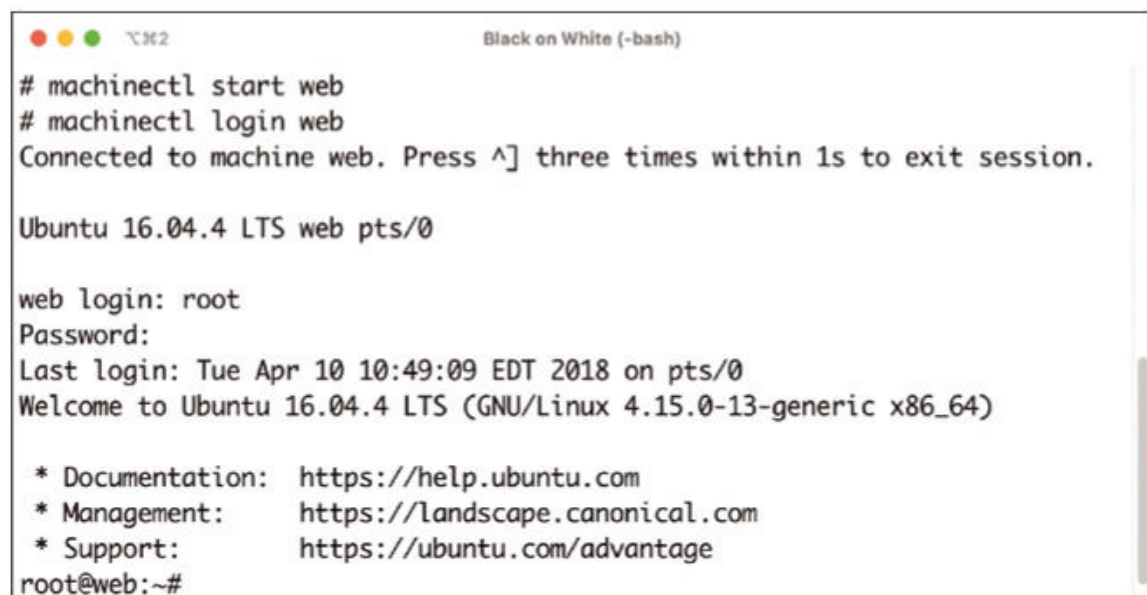
After a systemd reload, the command

```
machinectl start webserver-1
```

starts the freshly created container. If you now configure the directory you just copied to run a web server, it will run autonomously and in isolation from the rest of the system. Even if someone breaks into an unmaintained Joomla or TYPO3 on the web server, they do not automatically gain access to the resources of other users or the host – and completely without Docker, Podman, or other hipster stuff.

Mini-Containers

To understand what the second service I talk about in this article (systemd-portabled) does, you need to revisit the functionality of systemd-nspawn. In practical terms,



```

# machinectl start web
# machinectl login web
Connected to machine web. Press ^] three times within 1s to exit session.

Ubuntu 16.04.4 LTS web pts/0

web login: root
Password:
Last login: Tue Apr 10 10:49:09 EDT 2018 on pts/0
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.15.0-13-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
root@web:~#
  
```

Figure 4: The `machinectl` command manages containers in Nspawn in the shell. If so desired, the process can be automated with a unit file.

Portabled plays a very similar tune: Under the hood, it uses much of the functionality that Nspawn also uses.

Portabled has been part of systemd since version 239, so it should certainly be in place on recent distributions. Although Podman and Docker fans won't like to hear it, Portabled essentially offers precisely the features that Red Hat, SUSE, and their offspring have in mind when they talk about "rump systems" and look to deliver their software in containers. However, it does so without most of their complexity.

Admittedly, the container and its environment consequently lack a few features that Docker and others give you. When it comes to just isolating services and making them portable, though, Portabled is very handy, especially for existing systems that you want to harden without having to switch completely to Docker or Podman.

The basic idea behind Portabled is that you build small container images containing one or more services along with a matching configuration. If a Linux system has a current kernel with support for namespaces and a current systemd environment, the container image can be rolled out on this host and operated there – so the theory goes. The highlight is that this process is completely independent of the package manager in place.

Finding an Image

For an image to be usable with Portabled, it only needs to meet a few requirements. As in the previous example, the recommendation is to use tools like `debootstrap` to create a basic filesystem. As in the case of Nspawn, portable images do not need their own kernel or bootloader, but if you want to use a RAW image, it must be equipped with a suitable partition table that the Linux kernel on the host system understands. The systemd in the image also needs a working unit file for the service or services that the container will be running.

The `/etc/machine-id` file must be in place, as must `/usr/lib/os-release`. Also, a `resolv.conf` is required for the services in the container. Everything else is taken care of automatically by tools like `debootstrap`. The example here assumes that you have created a `lamp.raw` file that contains a basic Debian GNU/Linux 11 and has Apache 2, MariaDB, and PHP. Most importantly, the systemd

Listing 1: Systemd Unit for Container

```

# /etc/systemd/nspawn/webserver-1.nspawn
[Exec]
PrivateUsers=pick

[Network]
Zone=web Port=tcp:443

[Files]
PrivateUsersChown=yes
  
```


Listing 2: mkosi Config File

```
[Distribution] Distribution=debian
Release=bullseye

[Output]
Format=gpt_ext4 Bootable=yes Output=image.raw

[Packages]
Packages=openssh-client vim

[Validation]
Password=secret
```

unit files must be located in the image in `/usr/lib/systemd/system/lamp-apache.service` and `/usr/lib/systemd/system/lamp-mariadb.service` for Portabled to find them later. When Portabled then starts the container on the target system, it copies these files on the host and adds various custom settings that can relate to, say, logging or handling output on stdout. Clearly, the systemd developers wanted you to have to do as little work as possible with portable images.

Basic and Overlay Images

Talking of keeping the workload manageable, the workflow described here suggests that you have to copy the default image created at the beginning for each container that will contain a service or multiple services; however, this is not true because Portabled also allows you to combine multiple partial images and create a complete image with OverlayFS. For this to work, the extension images in the `/usr/lib/extension-release.d/` directory must contain a file with an arbitrary name that contains at least the `ID=` lines with the extension image ID, and `SYSEXT_LEVEL=` and `VERSION_ID` entries with the

extension image data that extends the original image. Like Nspawn, Portabled also has its own command-line interface for manipulating containers: `portablectl`. For example, the command

```
portablectl attach 2
--extension lamp_1.raw 2
debian-bullseye_1.raw lamp
```

attaches the `lamp` extension to the image for Debian GNU/Linux Bullseye. In the next step, the container that Portabled stitches together with OverlayFS can then be treated as if it were a complete container in its own right. The workflow described here makes it quite easy to maintain a basic image and to vary the use of many small extensions.

mkosi Can Help

If you are not comfortable with the distributors' tools for creating an image, you will find that the `mkosi` (make operating system image) tool is a good alternative. In simple terms, `mkosi` is a small tool that creates a folder with the complete filesystem of a Linux distribution, which can then be used in systemd with Nspawn or Portabled like a normal container. The program can be found in the systemd GitHub directory [3]; its use is

almost trivial. To create a Debian image that corresponds to that described in detail above, you just need the command:

```
$ mkosi -d debian 2
-r bullseye 2
-t gpt_ext4 -b --checksum 2
--password secret 2
--package openssh-client,vim 2
-o image.raw
```

An `image.raw` file will then contain a Bullseye image that in turn contains `openssh-clients` and `Vim`, along with the standard selection of packages. If you are not a fan of command-line parameters, you can alternatively pass in a configuration file to `mkosi`. The example from Listing 2 has the same effect as the command above.

One disadvantage of `mkosi` is that it does not take care of installing the packages it needs when creating images. On Debian systems, the task of manually installing the `debootstrap` and `debian-archive-keyring` packages falls to you before invoking `mkosi` (Figure 5).

Accessing System Resources

One last question remains to be clarified in the context of this article; otherwise, the container fun might come

```
Terminal - mloschwitz@debian: ~
File Edit View Terminal Tabs Help
mloschwitz@debian:~$ sudo apt-get install debootstrap debian-archive-keyring
[sudo] password for mloschwitz:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
debian-archive-keyring is already the newest version (2021.1.1).
Suggested packages:
  ubuntu-archive-keyring squid-deb-proxy-client
The following NEW packages will be installed:
  arch-test debootstrap
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 88.2 kB of archives.
After this operation, 554 kB of additional disk space will be used.
Get:1 http://ftp.tu-chemnitz.de/debian bullseye/main amd64 arch-test all 0.17-1 [12.9 kB]
Get:2 http://ftp.tu-chemnitz.de/debian bullseye/main amd64 debootstrap all 1.0.123 [75.3 kB]
Fetched 88.2 kB in 0s (401 kB/s)
Selecting previously unselected package arch-test.
(Reading database ... 96696 files and directories currently installed.)
Preparing to unpack .../arch-test_0.17-1_all.deb ...
Unpacking arch-test (0.17-1) ...
Selecting previously unselected package debootstrap.
Preparing to unpack .../debootstrap_1.0.123_all.deb ...
Unpacking debootstrap (1.0.123) ...
Setting up debootstrap (1.0.123) ...
Setting up arch-test (0.17-1) ...
Processing triggers for man-db (2.9.4-2) ...
mloschwitz@debian:~$
```

Figure 5: With `mkosi`, building images is far easier, but you have to install the distribution-specific tools up front.

to an abrupt end. How exactly do you provision host system resources in containers? I do not refer so much to specific hardware, because containers do not need their own kernel; they access the hardware directly through the host system's kernel anyway, although they detour through namespaces to do so, adding a degree of control. What I mean is much more likely to be programs that need access to parts of the /sys tree or that retrieve information from /proc. Occasionally, it also happens that an application in one container needs to access the Unix socket of an application in another container. The answer to how this works is quite simple: You make sure the directory in question exists on the host and tell systemd to provide the folders as bind mounts in the container by means of the `BindPaths=` and `BindReadOnlyPaths=` directives in the container's systemd unit files. However, keep in mind that this is a deliberate, intentional blurring of security boundaries. Therefore, you should only go for this option if you have absolutely no alternative.

Conclusions

Very few admins are aware of the systemd components Nspawn and

Portabled discussed in this article, and this cluelessness is a mistake, despite your opinion of systemd. If you use one of today's major distributions, chances are you have a setup with systemd. If it is already in place, why not just use it?

Both tools presented here offer genuine added value. Chroot is now considered insecure, and for good reason: Several scenarios have been documented for breaking out of a chroot environment. Namespaces in the Linux kernel are not only more modern, but also far more focused on security, where they offer considerable benefits. If you want to isolate applications, either from each other or from the rest of the system, without having to deal with the complexity of Docker or Podman, it is a very good idea to take a closer look at the systemd add-on Nspawn.

The same goes for Portabled. Strictly speaking, the idea behind it is nothing other than what the major vendors are currently pursuing with their container strategies. Instead of the dependency hell of the usual package managers, cleanly defined container images contain just the bare necessities and otherwise have no external dependencies.

Portabled can be forgiven for not following the container mantra "a microarchitecture application in a container" – especially against the background that Portabled is more likely to be used in classic environments in most cases anyway. In return, you can look forward to more convenience, enhanced security, and better administrability.

Anyone who is concerned about isolating services and securing their systems should definitely have these two standard systemd functions on their radar. ■

Info

- [1] Docker architecture: [\[https://docs.docker.com/get-started/overview/\]](https://docs.docker.com/get-started/overview/)
- [2] "Private networking per-process in Linux" by Ivan Zahariev: [\[https://blog.famzah.net/2014/06/05/private-networking-per-process-in-linux/\]](https://blog.famzah.net/2014/06/05/private-networking-per-process-in-linux/)
- [3] mkosi: [\[https://github.com/systemd/mkosi\]](https://github.com/systemd/mkosi)

The Author

Freelance journalist Martin Gerhard Loschwitz focuses primarily on topics such as OpenStack, Kubernetes, and Ceph.





Harden services with systemd

A Hard Nut to Crack

Systemd comes with a metric for determining the security of your system, letting you track how any service can be secured step-by-step in a sandbox. By Jens-Christoph Brendel

One of the most important goals in the development of systemd is securing Linux. Of course, you can only improve what can be measured, which is why Galileo Galilei advised: “Measure what is measurable, and make measurable what is not.” Following this maxim, systemd now

makes system security under Linux measurable and improvable. More specifically, it is the `systemd-analyze` security command that allows this measurement. When executed, it returns a table like that shown in [Figure 1](#), listing each service managed by systemd (*UNIT*);

a numerical value for the degree of protection (*EXPOSURE*, where 10 is both the highest and worst value); a verbal translation of this value (*PREDICATE*); and another version of the rating (*HAPPY*) in the form of an emoji.

Additionally, `systemd-analyze` can reveal how it arrives at its assessment: To see this, start it with the name of a service unit. As shown in [Figure 2](#), it lists all the factors that have been checked, along with a checkmark for passed or an X for failed.

Not a Tough Cookie

After that, the user knows systemd’s opinion on the security status of the services it checked, but what can be done to improve the bad scores? To find out, you can build a minimal service, whose security you then elevate step-by-step. As an example, first create a minimal HTML page in an empty directory (e.g., `/home/$USER/Python/sectest/`, which will serve later

UNIT	EXPOSURE	PREDICATE	HAPPY
ModemManager.service	5.8	MEDIUM	😞
NetworkManager.service	7.8	EXPOSED	😞
accounts-daemon.service	9.6	UNSAFE	😞
acpid.service	9.6	UNSAFE	😞
alsa-state.service	9.6	UNSAFE	😞
anacron.service	9.6	UNSAFE	😞
apport.service	9.6	UNSAFE	😞
arangodb3.service	9.6	UNSAFE	😞
avahi-daemon.service	9.6	UNSAFE	😞
blueman-mechanism.service	9.6	UNSAFE	😞
bluetooth.service	6.8	MEDIUM	😞
cassandra.service	9.6	UNSAFE	😞
cockpit-wsinstance-http-redirect.service	9.2	UNSAFE	😞
cockpit-wsinstance-http.service	9.2	UNSAFE	😞
cockpit.service	9.2	UNSAFE	😞
colord.service	8.8	EXPOSED	😞
containerd.service	9.6	UNSAFE	😞
cron.service	9.6	UNSAFE	😞
cups-browsed.service	9.6	UNSAFE	😞
cups.service	9.6	UNSAFE	😞
dbus.service	9.6	UNSAFE	😞
dm-event.service	9.5	UNSAFE	😞

Figure 1: Tabular rating of services by security aspects.

Lead image © Diddeman, Pixabay.com

as the document root of a small web server) (Listing 1). The easiest approach is to borrow the web server itself from Python, which already has a simple model that can be used with virtually no configuration. Next, wrap the server start in a systemd unit file – again, keeping it as simple as possible (Listing 2). Now save the unit file as /lib/systemd/system/helloworld.service and the HTML page as index.html in the document root directory. After typing

```
systemctl start helloworld.service
```

enter localhost:8080 in the address bar of a web browser to bring up the plain Hello World page. In this state, without any precautions, the service is completely unprotected. In the output of systemd-analyze security, it appears with a high score of 9.6 as UNSAFE and a shocked emoji (Figure 3).

Fundamentals

In the first step, add a line reading

```
NoNewPrivileges=true
```

to the Service section of the unit file to prevent the process from escalating its privileges later (e.g., with setuid or setgid bits). After this (as for all subsequent additions to the unit file),

you need to reload all unit files and restart the service:

```
systemctl daemon-reload
systemctl restart helloworld.service
```

If you now look at the output of systemd-analyze security, the exposure value of helloworld.service has already dropped slightly, from 9.6 to 9.4. Admittedly, this still counts as unsafe. On with the task: A whole class of attacks can be rendered impossible by adding

```
PrivateTmp=yes
```

to the unit file, which causes systemd to create a new, exclusive filesystem namespace for the process and to mount /tmp and /var/tmp/ there. Therefore, the temporary files are no longer shared publicly and are

grub-common.service	9.6	UNSAFE	😱
addtemp.service	9.6	UNSAFE	😱
helloworld.service	9.6	UNSAFE	😱
influxdb.service	9.2	UNSAFE	😱
irqbalance.service	6.1	MEDIUM	😬
scsiid.service	9.5	UNSAFE	😱
kerneloops.service	9.2	UNSAFE	😱

Figure 3: Starting point: The new service is completely insecure.

immediately deleted after the process ends. Attacks based on swapping or manipulating temporary files now come to nothing. The exposure value drops to 9.0, but the rating remains unsafe. The next step is to add

```
RestrictNamespaces=uts ipc pid user cgroup
```

to the unit file, which prevents the process from accessing the listed namespaces. The list deliberately excludes the net namespace and a few others that the web server has to use. After this action, the exposure value drops below 9 (to 8.8) for the

Listing 1: Minimal HTML Page

```
<!doctype html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Hello World</title>
  </head>
  <body>
    <p><h1>HELLO WORLD!</h1></p>
  </body>
</html>
```

Listing 2: Unit File

```
[Unit]
Description=Simple HTTP Server
Documentation=https://docs.python.org/3/library/http.server.html

[Service]
Type=simple
WorkingDirectory=/home/jcb/Python/sectest
ExecStart=/usr/bin/python3 -m http.server 8080
ExecStop=/bin/kill -9 $MAINPID

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


geoclue.service	7.4	MEDIUM	😬
getty@tty1.service	9.6	UNSAFE	😱
grub-common.service	9.6	UNSAFE	😱
hddtemp.service	9.6	UNSAFE	😱
helloworld.service	5.7	MEDIUM	😬
influxdb.service	9.2	UNSAFE	😱
irqbalance.service	6.1	MEDIUM	😬
iscsid.service	9.5	UNSAFE	😱

Figure 4: For the first time, the emoji is not dissatisfied: You have achieved a medium level of security.

first time, and the rating is no longer unsafe, only *EXPOSED*. The emoji’s expression changes from horrified to merely unhappy.

Kernel and Control Groups

The next step is to enable additional protections in the unit file:

```
ProtectKernelTunables=yes
ProtectKernelModules=yes
ProtectControlGroups=yes
```

The kernel variables, which users can access via `/proc/sys/`, `/sys`, `/proc/sysrq-trigger/`, `/proc/latency_stats/`, `/proc/acpi/`, `/proc/timer_stats/`, `/proc/fs/`, and `/proc/irq/`, are now read-only and therefore no longer editable for the process. In any case, the system should only have write access to these variables during booting, so you are not losing any functionality. Because the web server does not need any special kernel modules, you have

also stopped it loading and unloading such modules for the web server process. From now on, it cannot access the control groups. Although container administration software might need this access, a web server does not. This step pushes the exposure value down to 8.1. Finally, you can set:

```
ProtectSystem=strict
PrivateUsers=strict
```

The first line mounts `/usr` and the bootloader directories `/boot` and `/efi` in read-only mode for all processes that this unit starts. The second line configures a user group mapping for the process that maps root and the user that starts the unit’s main process to itself – but maps all other users or groups to *nobody*. The system’s user and group database is thus decoupled from the process running in its own sandbox. The exposure value now drops below 8 (more precisely, to 7.8).

Capabilities

Finally, you can limit the capabilities that will be available to the process. These are rights that can be granted to unprivileged processes in small chunks, which makes it unnecessary to give a process completely

unrestricted superuser rights just because it needs a single special right. You can take a fairly restrictive approach here and define

```
CapabilityBoundingSet=2
CAP_NET_BIND_SERVICE 2
CAP_DAC_READ_SEARCH
```

which excludes, for example, the assignment of `CAP_SYS_ADMIN`, `CAP_DAC_OVERRIDE`, or `CAP_SYS_PTRACE` to the process and deducts many points. The exposure value now drops to 5.7 (**Figure 4**). The rating now confirms a *MEDIUM* level of security, and for the first time the emoji now looks neutral and no longer unhappy about the situation.

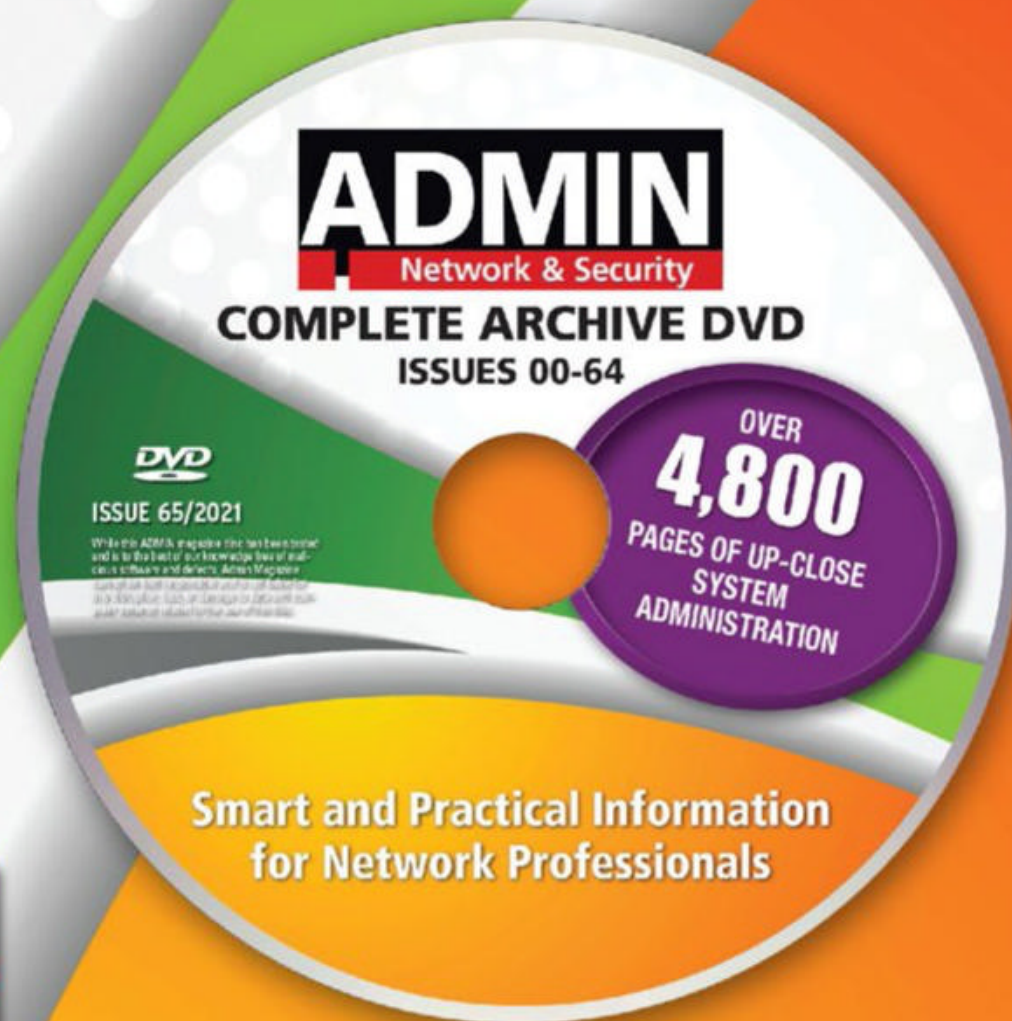
Conclusions

Quite a few options are yet left to provide additional security. A good compilation of all systemd options suitable for hardening services and that open up a wide field for further optimizations is provided in a description on GitHub **[1]**. With `systemd-analyze` as a measuring tool, you can track your progress in each case. ■

Info

[1] Hardening options for systemd services: [\[https://gist.github.com/ageis/f5595e59b1cddb1513d1b425a323db04\]](https://gist.github.com/ageis/f5595e59b1cddb1513d1b425a323db04)

12 Years of ADMIN on One DVD



This searchable DVD gives you 64 issues of ADMIN, the #1 source for:

- DevOps
- ransomware
- edge computing
- automation
- orchestration
- and more from the front lines of IT

Clear off your bookshelf and complete your ADMIN library with this powerful DVD!

ORDER NOW!
shop.linuxnewmedia.com



Managing BitLocker with PowerShell

Babylonian Letters

BitLocker provides current Windows versions a good, closely integrated encryption solution and lets you manage and maintain this feature with PowerShell. By Thomas Bär and Frank-Michael Schlede

Encrypting drives on Windows in user circles is still not as widespread as is prudent from a security perspective. Especially for mobile systems such as notebooks, hard drives should be secured with a transparent encryption solution such as BitLocker so that data is protected, even if a thief removes the hard drive and attempts to access it. System administrators should seriously consider taking advantage of the option to automate the setup and configuration of hard drive encryption.

This is where PowerShell comes into play. Microsoft provides command-line tools and matching cmdlets in the form of `manage-bde` and `repair-bde`. Both the PowerShell cmdlets and command-line commands let you handle all the tasks and settings that are supported through the control panel. Although the BitLocker setting is now linked in the new system settings in the pre-release version of Windows 11 and version 21H2 of Windows 10, it still takes you to the settings in the legacy Windows interface from the pre-Windows 10 era.

You are better advised to use the command line and PowerShell. We tested the examples in this article both with version 5.1, which is currently installed on Windows systems by default, and with the open source variant PowerShell 7 on Windows 10 and Windows Server 2019 computers. BitLocker cmdlets and options were not different between systems.

Adding Encryption

BitLocker is not available on the current Windows 10 Home version. The device encryption software is available by default on Windows 10 Professional or Enterprise but usually has to be enabled. Exceptions include, say, Surface devices by Microsoft – the same type of encryption is automatically enabled there. The server versions of Windows also support the use of drive encryption. However, for the server operating systems, you need to install BitLocker as a feature in Server Manager or the new Windows Admin Center. Additionally, the option to unlock operating system volumes of

client systems on domain networks automatically by means of network unlocking on reboot can then be added to the system as a feature. Of course, you can add BitLocker to a Windows server with PowerShell, but first check to see whether BitLocker is already installed on the system with:

```
Get-WindowsFeature -Name Bitlocker
```

If the *Install State* column of the cmdlet reports that the feature is installed, the software is ready to use. If, on the other hand, it says *Available*, the administrator has to install BitLocker (**Figure 1**), which can be done with the associated sub-features and tools by entering:

```
Install-WindowsFeature Bitlocker ?
-IncludeAllSubFeature ?
-IncludeManagementTools
```

After that, the cmdlet reports the *Success* status as *True* and explicitly points out that a restart of the operating system is now needed. You can initiate this with `Restart-Computer` directly in PowerShell.

cmdlets

After the reboot, drive encryption is then available, as well as the corresponding PowerShell module with the cmdlets [1] for BitLocker management. Typing

```
Get-Command -module BitLocker
```

shows you which special cmdlets are now available to you (Figure 2). If you want to find out about all the drives on your Windows system and their encryption status, call (with administrator privileges) Get-BitLockerVolume. The command lists all drives

that are connected to the computer. To display only a specific drive, use the -MountPoint <drive> parameter:

```
Get-BitlockerVolume -MountPoint "E:"
```

The Protection Status, VolumeStatus, and AutoUnLock enabled options show important information about the current status of hard disks or partitions.

If the drive is encrypted, Protection Status is set to On. VolumeStatus is even more precise and shows whether the disk or partition is already fully encrypted. The AutoUnlock enabled option shows whether you have configured the disk for automatic drive

unlocking. On a system that has a lot of drives, listing only those drives with full BitLocker encryption by querying the VolumeStatus property of the corresponding object can be handy:

```
Get-BitlockerVolume | Where-Object { $_.VolumeStatus -eq 'FullyEncrypted' }
```

With the help of two cmdlets, Enable-Bitlocker and Disable-Bitlocker, you can then encrypt or decrypt a drive or a partition by specifying the desired drive again. The command for encryption requires a few more parameters.

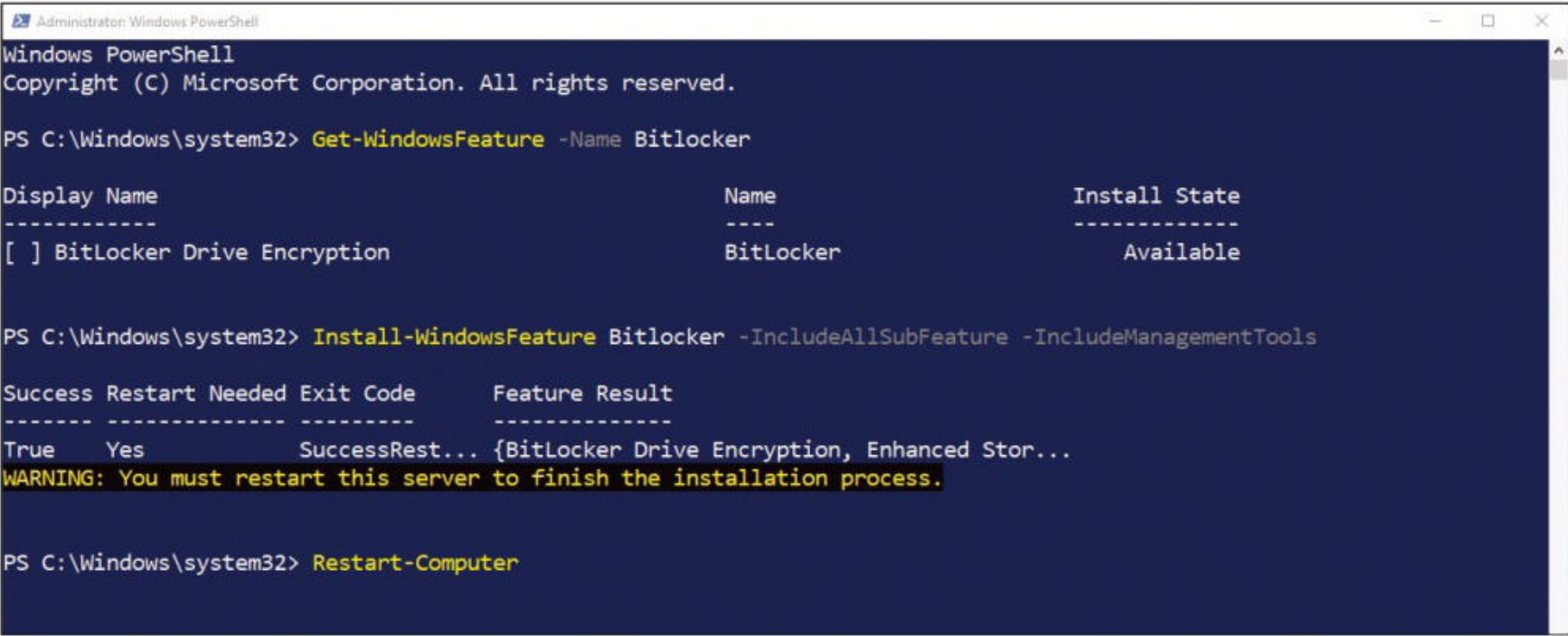


Figure 1: On Windows Server (version 2019 is shown here), BitLocker needs to be installed.

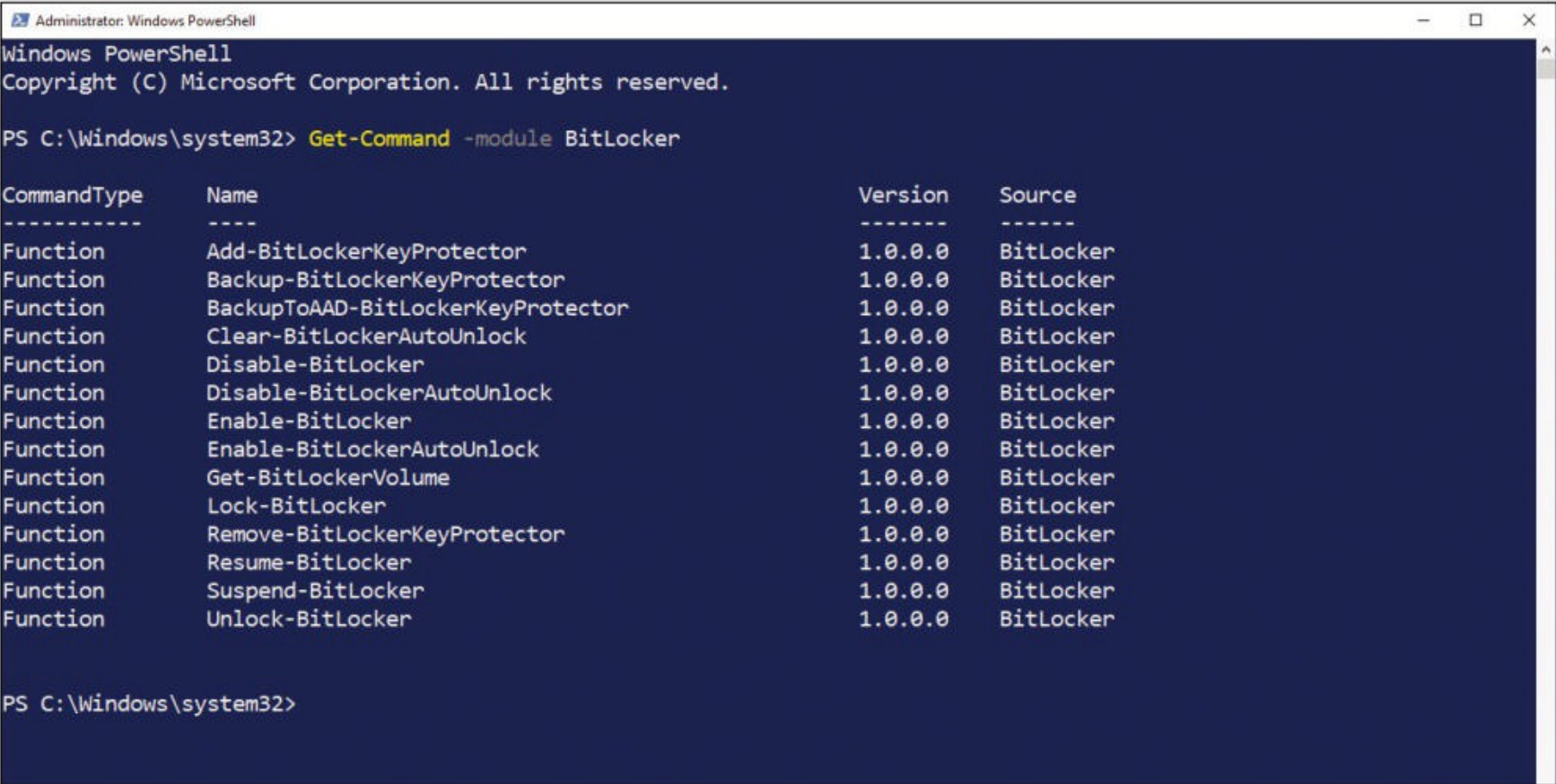


Figure 2: Finding the special cmdlets for working with BitLocker.

In addition to the encryption method, which you specify with the `-EncryptionMethod` parameter, you must define the password you use to lock the drive (**Figure 3**). Instead of the `-RecoveryKeyPath` parameter, you can, among other things, specify the path to a recovery key that is located elsewhere (e.g., on a USB stick). Although we used the `-EncryptionMethod AES256` parameter for the encryption method, you can choose between AES256 and AES128 in your script. The `-UseSpaceOnly` parameter additionally specifies that you only want to encrypt the space on the volume that is occupied by data. We simply passed the password to the `$PassW` variable with the `Read-Host` cmdlet to demonstrate the use of `Enable-Bitlocker`:

```
$PassWd = Read-Host -Prompt "Password, please?" -AsSecureString
Enable-Bitlocker -MountPoint "E:" -EncryptionMethod AES256 -UsedSpaceOnly -Password $PassWd -PasswordProtector
```

```
-Password $PassWd
-PasswordProtector
```

After this call, the drive is encrypted but not yet “locked,” which is accomplished with the help of the final command in **Figure 3**:

```
Lock-Bitlocker -MountPoint "E:"
```

Now, users who want to change to this drive need to enter the password in Explorer or use the `Unlock-Bitlocker` cmdlet with the password. Again, the assumption is that you stored it in the `$PassW` variable:

```
Unlock-Bitlocker -MountPoint "E:" -Password $PassW
```

The `Enable-Bitlocker` cmdlet in particular offers a large number of other parameters and options. Microsoft provides detailed documentation online [\[2\]](#), including a description of how to use a trusted platform module (TPM), among other things. With the help of the command

```
get-help Enable-Bitlocker -full
```

you can display a whole series of examples onscreen that show you how the various parameters come into play.

At the Command Line

Microsoft also offers the `manage-bde` command-line tool on its operating systems for scripting BitLocker calls. The tool offers a useful choice of options for rolling out and managing drive encryption. To use it, you will need administrator rights (i.e., a command prompt or a PowerShell window with elevated rights). The following call then shows the BitLocker status on the local system:

```
manage-bde status
```

The output is quite detailed and not only shows you the conversion status (Is the entire hard drive encrypted or only the occupied space?) but also the encryption method and the type of key protection device used. You can then immediately see whether the

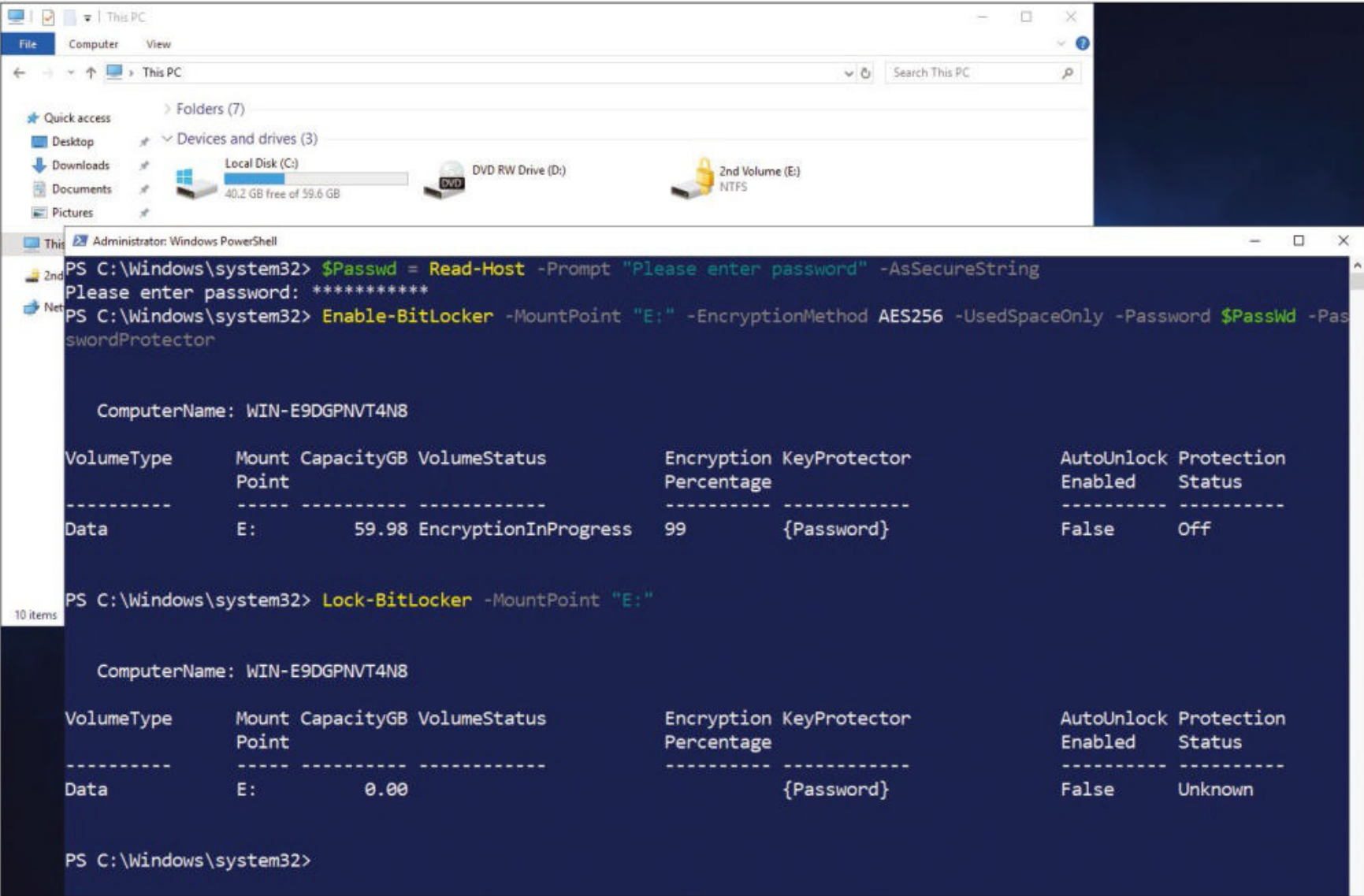


Figure 3: Encrypting the device.

system is equipped with a TPM and whether it is used on the volume with the operating system. Calling

```
manage-bde on <drive>
manage-bde off <drive>
```

then lets you switch on or off encryption for the respective drive. For example, if you want to unlock a drive that is protected by BitLocker, run the following command with the appropriate key (always 48 digits) or with the recovery password in the following form:

```
manage-bde -unlock C: 2
-RecoveryPassword 670499-444444-307582-2
555555-209561-145200-316107-999999
```

With the help of the `-RecoveryKey <drive>` parameter, you can then also load a key that is stored on an external drive:

```
manage-bde -unlock F: -RecoveryKey T:\
```

According to the documentation, `manage-bde` also offers the option to read and configure the TPM module. However, calling

```
manage-bde -tpm
```

caused an error message in our lab on both the current Windows 10 (Professional and Enterprise in version 21H1) and Windows Server 2019

(version 1809) versions. The message said `manage-bde` could not manage the trusted platform module “in this version of Windows” and was followed by the recommendation to edit the Microsoft Management Console *Manage TPM* snap-in or the corresponding PowerShell cmdlets.

We were then able to display the TPM settings on the various devices without any problems by calling the `Get-TPM` cmdlet.

As an administrator, `TrustedPlatformModule` provides a whole series of cmdlets with which you can not only discover the trusted platform module but reset it or examine which features the module supports on the specific device, among other things.

Finally, we would like to mention the `repair-bde` command-line tool for the sake of completeness. This software is an additional offering from Microsoft that, according to the description, tries to reconstruct critical parts of a severely damaged drive and recover any recoverable data. However, this explicitly only applies if the drive was encrypted by BitLocker and the user has a valid password or recovery key for decryption. The tool has the syntax:

```
repair-bde <input_volume> 2
<output_volume_or_images>
```

You should be able to use the key package and recovery password and

key to decrypt parts of a BitLocker-protected drive, even if the disk is damaged. However, the command cannot repair a drive if anything failed during the encryption or decryption process.

Conclusions

Encrypting drives significantly reduces the vulnerability of a Windows system, especially on mobile devices. Because BitLocker is an integral part of the professional versions of Windows 10 – and likewise with Windows 11, as well – it is a good choice for encrypting both drives built into computers and external storage media such as USB sticks. Microsoft offers many ways to automate the configuration and monitoring of drive encryption by scripts with a series of PowerShell cmdlets and additional command-line programs. ■

Info

- [1] PowerShell BitLocker management: [\[https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-use-bitlocker-drive-encryption-tools-to-manage-bitlocker\]](https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-use-bitlocker-drive-encryption-tools-to-manage-bitlocker)
- [2] Enable-BitLocker cmdlet: [\[https://docs.microsoft.com/en-us/powershell/module/BitLocker/enable-BitLocker?view=windowsserver2019-ps\]](https://docs.microsoft.com/en-us/powershell/module/BitLocker/enable-BitLocker?view=windowsserver2019-ps)



Gatling load-testing tool

Stressed

Generate load on servers and services with the Gatling load-testing tool. By Christopher Dock

When questioned at work about how the test tool Gatling [1] compared with the somewhat old JMeter, I was at a loss for words. However, it did give me the opportunity to learn more about Gatling and eventually present another possible tool in the arsenal against weak servers and services.

In retrospect, I'm not sure how I missed Gatling, which is no recent upstart. Gatling's first stable release

was in 2012, and only a few years later, its founder created Gatling Corp. to develop and maintain the software. Over the years, Gatling morphed into a product with both open source and Enterprise variants that can hold their own against other test tools.

Yet, Gatling has taken a significantly different path from some of the other load-testing products such as JMeter and LoadRunner. With Gatling, you

don't have a custom IDE for development. Instead, you use the Gatling framework in your favorite editor to write your own test script in Scala.

Getting Started

All Gatling tests start pretty much the same, by subclassing from the Scala `Simulation` class. The simulation is essentially a collection of individual

Listing 1: Simple Gatling Example

```
01 package com.mypackage
02
03 import scala.concurrent.duration._
04 import io.gatling.core.Predef._
05 import io.gatling.http.Predef._
06
07 class BlazeDemoV1 extends Simulation {
08
09   var qualifiedhost = "http://blazedemo.com"
10   var proxyport = 0
11   var proxyhost = "not set"
12
13   // headers for HTTP call
14   val headers = Map(
15     "Accept" -> "text/html",
16     "User-Agent" -> "LinuxMagazine/1.0.1",
17     "Accept-Encoding" -> "gzip, deflate",
18     "Connection" -> "keep-alive",
19     "DNT" -> "1" )
20
21   // http connector
22   var httpProtocol = http
23     .baseUrl(qualifiedhost)
24
25   if (proxyport > 0)
26     httpProtocol = httpProtocol.proxy(Proxy(proxyhost, proxyport))
27
28   val scn = scenario("BookFlight")
29     .exec(http("step_1")
30       .get("/")
31       .headers(headers)
32       .check(status.is(200)))
33     .exec(http("step_2")
34       .get("https://google.com/")
35       .headers(headers)
36       .check(status.is(200)))
37     .pause(5)
38
39   setUp(
40     scn.inject(atOnceUsers(1))
41   ).protocols(httpProtocol)
42 }
```


requests that a user or program would normally make. Because your test is a program, you are given the flexibility of creating variables, constants, and methods, and because you are using a proper programming language, you can create your own library of support objects or methods.

Your performance tests can include one or more classes, but you can create and include a lot of regular Scala objects, as well. Scala also supports an object type that appears to be similar to a class, but it is more akin to a singleton than a normal Java or C++ class.

You don't have to be a Scala programmer to enjoy the freedom that Gatling provides. Most developers, despite having a favorite language or toolset, can easily learn enough Scala to create their own test scripts in a few hours. Because the test script is pure source code, you receive extra advantages that are commonly enjoyed with normal software development: the use of common developer tools such as `git`, `diff`, and `grep`.

The first five lines of [Listing 1](#) assign the class to a package and include all of the necessary reference information for the program. Of course, this

process is reminiscent of Java. One of the neat technology choices that was made is that the Scala code is compiled by the Java compiler and then uses the Java Virtual Machine (JVM) installed on your computer.

This one feature alone allows you to use any of the existing standard Java libraries or even your own custom code. Yet Java code is subtly different in syntax, so even using standard `Java.io` calls to read files will look fairly foreign once completed.

Line 7 defines the class file, which inherits from the Scala base class `Simulation`, and lines 9-11 define a few variables. In a proper production-quality script, these variables would probably be replaced with values that are passed in or perhaps read from a configuration file. The proxy configuration is not being used in this example, so simply setting these variables for your proxy will allow you to run the script over the proxy without any further modifications.

Lines 13-19 define the key pairs to be used as header values for HTTP statements. This collection has been defined as a constant with the keyword `val`, whereas the variables in the previous lines are re-assignable because they use the keyword `var`.

Listing 2: Parallel Load Tests

```
01 setUp(
02   scn1.inject(
03     constantConcurrentUsers(2).during(60.seconds),
04     rampConcurrentUsers(2).to(4).during(10.seconds)
05   ).protocols(httpProtocol),
06   scn2.inject(
07     constantUsersPerSec(2).during(15.seconds),
08     rampUsersPerSec(2).to(4).during(10.seconds)
09   ).protocols(httpProtocol)
10 )
```

The framework does have the `http` object, which encapsulates all of the logic for connections between machines over the Internet. Instantiating a variable with quite a number of different parameters is possible with this class. The most important parameter is the URL of the machine to which to connect. Just like with other languages, additional method calls can perform additional variable setup. Lines 21-26 demonstrate how to create the HTTP protocol variable and how to override it in favor of a proxy server, if one is needed.

Lines 28-37 show the creation of a `scenario` that is just a list of all the different statements that will be called. The `scenario` test will be run again and again. In this example, one

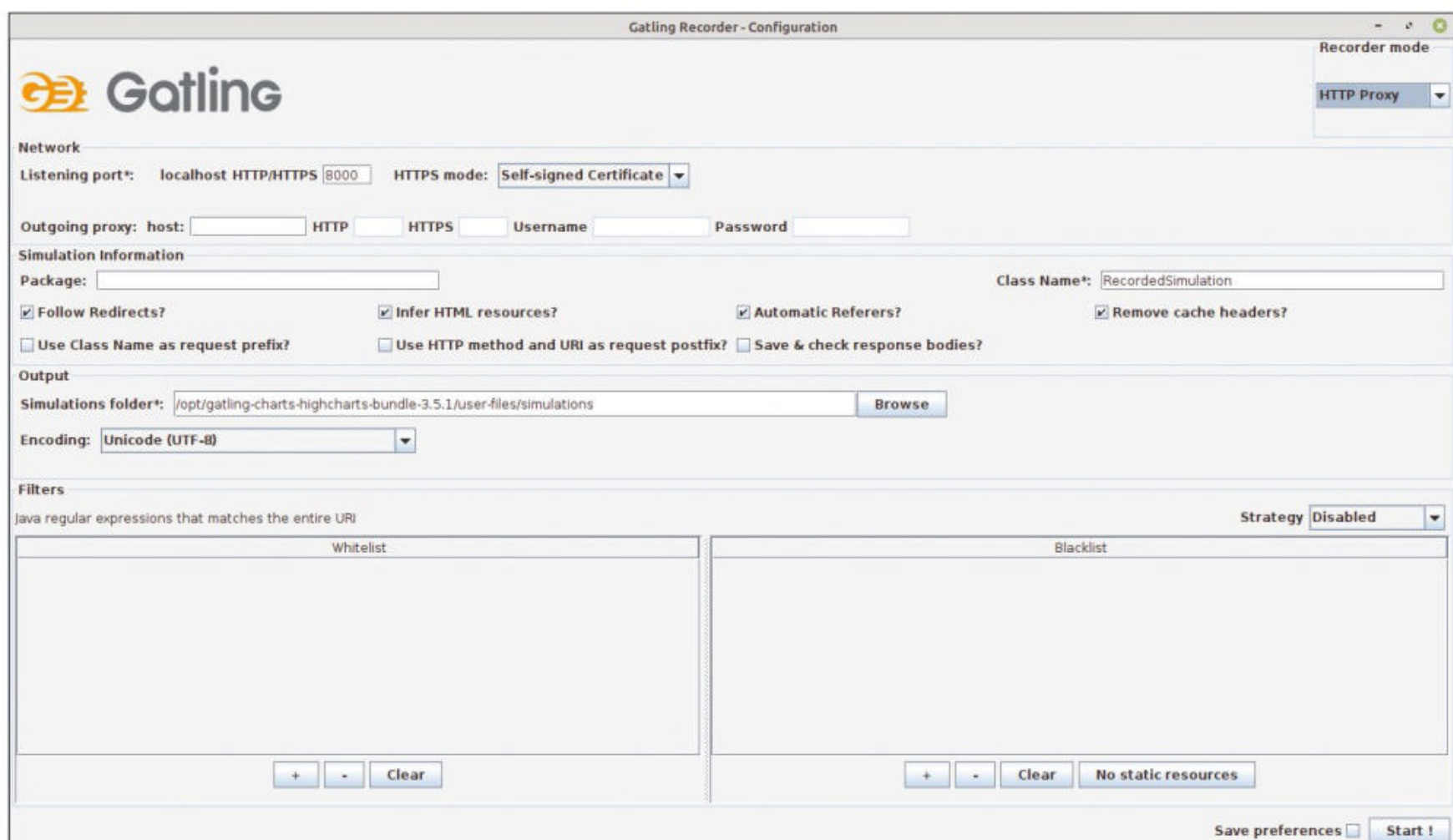


Figure 1: Gatling proxy server recorder setup.

Listing 3: Proxy Recorder Excerpt

```

01 val scn = scenario("acmewebsite")
02   .exec(http("request_0")
03     .get("/")
04     .headers(headers_0)
05     .resources(http("request_1")
06       .get(uri2 + "/resources/sites/phoenix/style/font/teleneo-variable.woff2")
07       .headers(headers_1),
08       http("request_2")
09       .get(uri2 + "/binaries/assets/fonts/TeleGroteskScreen-Regular.woff")
10       .headers(headers_2),
11       http("request_3")
12       .get(uri2 + "/binaries/assets/fonts/phx-core-icons.woff")
13       .headers(headers_2),

```

of the calls uses the default URL that has been set up in the HTTP protocol. The second call uses a different URL that has no connection to the URL that was used during object creation. Both of these statements use pre-defined header values but could just as easily be different sets for different calls. Each of these statements check that the call receives return code 200, indicating a successful call.

The final and most important part of this script is lines 39-41. All lines up to this point define and prepare a test scenario, but in these three final lines, the user scenario is executed by the framework. In this particular code, the test is only run once with a single user. Although not a very realistic load test, it is a perfect smoke

test to verify that everything is running correctly.

Gatling provides quite a number of ways to create different load scenarios. Listing 2 shows a more complex setup that includes two different scenarios running in parallel. Each of these scenarios use different methods for generating user load.

Point-and-Click Testing

Not everyone is a natural-born software developer. However, quite a few people who might not be able to create a program from scratch can make small modifications to or extend existing programs.

Gatling provides a crutch for people who either cannot write their own

scripts or who want to lighten their load, by providing a proxy server that will record the steps that pass through it (Figure 1).

To capture a URL and what you do while at that URL, simply start up and point your web browser to the proxy server. The information is then processed into a Scala source file and saved with a class name you provide. This output is saved into the default location for Scala simulations.

To use this recorder, just press the *Start* button, which opens another window (Figure 2) that lets you follow along as you select pages in your web browser.

A side effect for scripts that have been created by the recorder can be seen in Listing 3. That is, the code is pretty difficult to read because most modern web pages have a lot of resources, Java scripts, or other frameworks. Everything that is downloaded by the website will also be downloaded in tests created with the proxy recorder. These resources are necessary for a web page but might not be required for performance testing.

The proxy recorder is a convenient way to create a sample script that can be used as a source when writing your own test scripts. This recorder can be run for all of the steps in your test. Then, you have all the headers as well as the URLs you will need in code form. However, all but the most simple tests should probably be refactored into a few Scala objects and classes to make the code understandable and the test script maintainable.

Extending Scala and Java

Perhaps the major advantage of Gatling tests is that they are written as small programs that can be organized into smaller, easy-to-understand units, which also makes the code calling it much easier to understand. The most reasonable decomposition would be to extract the testing steps from Listing 1 and put them into their own object, similar to Listing 4. The new DemoSteps object contains the headers and two steps, each

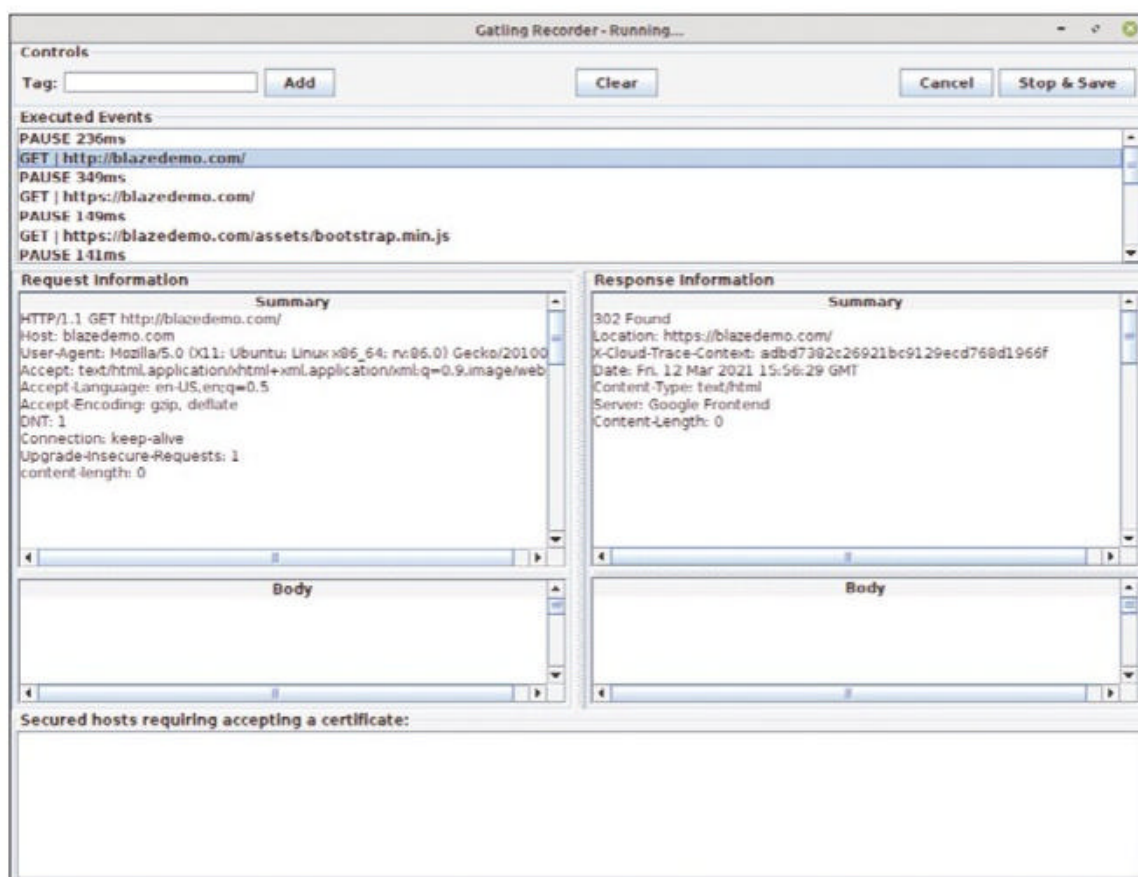


Figure 2: Proxy recorder window.

Table 1: Gatling Command-Line Args

Command	Function
-sf <simulation dir>	Path for Gatling class and object sources
-rf <results dir>	Location of test reports
-bf <temp dir>	Temp directory of compiled objects
-rsf <resource dir>	Location of input files
-s <classname>	Name of Scala class to run
-rd <description>	Descriptive text added to a report

having a meaningful name. This code is easy to understand and maintain and makes the scenario definition four very readable lines:

```
val scn = scenario("BookFlight")
  .exec(
    DemoSteps.getSitePage,
    DemoSteps.getGooglePage);
```

If this test had many steps and a more meaningful object name, it would be possible for anyone to have a complete understanding of what the test does and how it is organized. You can create simple methods within a Scala object that use plain old Java code and library calls:

```
def getnow() : String = {
  var calendar = Calendar.getInstance()
  var now = calendar.getTime() + "";
  now
}
```

In this way, you can leverage all of your expertise and custom libraries in your tests.

Running Tests

Gatling is a command-line program, which makes it easy to script tests. The `gatlingdemo.sh` script [2] can be run directly without parameters, in which case it will use all of the default directories and provide a list of all available Gatling tests in the `simulations` directory. A convenient way of organizing tests is to have the source code and test results in separate directories, which can be accomplished by having all of the Gatling directories in a separate directory by test (Table 1). The example in Listing 5 is a simple script

that runs a test and uses those local directories. All of the output goes to stdout. A small status block (Listing 6) lets you keep an eye on the status of the script that is running Gatling, with the successes and failures for each step output every five seconds. Normally you would not watch this output, but it is a convenient way to verify that everything is working as the script gets started. Once Gatling is finished running, it will output a final block with some statistics, as well as the directory where the output report has been generated.

Gatling Output

Reports are generated as HTML (Figures 3 and 4), and all reports can be made available over your intranet, where users can view them with their web browser. Because the reports are powered with a bit of JavaScript, you can zoom in and inspect the steps and define the starting and ending times.

Listing 4: Test Code in Separate Object

```
01 package com.mypackagename
02
03 import io.gatling.core.Predef._
04 import io.gatling.http.Predef._
05
06 object DemoSteps {
07
08     // headers for HTTP call
09     var headers = Map(
10         "Accept" -> "text/html",
11         "User-Agent" -> "LinuxMagazine/1.0.1",
12         "Accept-Encoding" -> "gzip, deflate",
13         "Connection" -> "keep-alive",
14         "DNT" -> "1")
15
16     var getSitePage =
17         exec(http("step_1_getpage")
18             .get("/")
19             .headers(headers)
20             .check(status.is(200)))
21
22     var getGooglePage =
23         exec(http("step_1_getgoogle")
24             .get("https://google.com")
25             .headers(headers)
26             .check(status.is(200)))
27 }
```

Listing 5: Running a Local Test

```
#!/bin/bash
CLASS=$1
SRC=`pwd`/simulations
RESOURCE=`pwd`/resources
RESULT=`pwd`/results
TMP=`pwd`/temp
DESC="desc goes here"

gatling.sh -sf $SRC -rsf $RESOURCE -rf $RESULT -bf $TMP
-s $CLASS -rd DESC
```

Listing 6: Test Status

```
=====
2021-06-20 22:49:24                                17s elapsed
---- Requests -----
> Global (OK=24 KO=0 )
> step_1_getpage (OK=3 KO=0 )
> step_1_getpage Redirect 1 (OK=3 KO=0 )
> step_2_reserve (OK=3 KO=0 )
> step_2_reserve Redirect 1 (OK=3 KO=0 )
> step_3_purchase (OK=3 KO=0 )
> step_3_purchase Redirect 1 (OK=3 KO=0 )
> step_4_confirm (OK=3 KO=0 )
> step_4_confirm Redirect 1 (OK=3 KO=0 )
---- BookFlight -----
active: 0 / done: 3
=====
```




Figure 3: Results in tabular form.

Full Test

A BlazeMeter [3] sample website simulates a travel site that can be used for testing JMeter. HTTP calls made to this site step through the simulated process of booking a ticket. Because Gatling can also make HTTP calls,

you can also use this same website to test your skills. Booking a ticket on the site takes four simple steps:

- Choose departure and destination cities
- Pick an airline
- Enter payment details
- View the confirmation



Figure 4: Graphs of response times.

The calls for this test have been stored in their own Scala FlightSteps object for better clarity. The Scala code for this object is quite similar to those seen in Listings 1 and 3. All of the code for this test is available online [2], but perhaps the most interesting part of the test is the main script (Listing 7).

Most of the first 26 lines of the test script are similar to the first Gatling example. It includes all the required Gatling include files (i.e. import io.gatling.core.Predef._), as well as two custom objects. Lines 13-14 retrieve values from Bash environment variables. When these variables are defined, the values will be used to set up the proxy. Lines 28-35 define the test scenario that performs the four steps of purchasing a ticket. The most important lines are 37-42. In this example, no real load is generated against the server, but it would be just as easy to change the code to run the scenario 50 or 100 times in parallel.

Although Gatling is a nice tool, you should compare it against alternative performance testing tools, such as the proprietary LoadRunner or the open source JMeter [4]. This exact same test with the BlazeMeter test site is also available in a previous article [5] for a direct comparison with JMeter.

Info

- [1] Gatling: [https://gatling.io/]
- [2] Code for this article: [ftp://ftp.linux-magazine.com/pub/listings/admin-magazine.com/67/]
- [3] BlazeMeter test site: [https://blazedemo.com]
- [4] Apache JMeter: [https://jmeter.apache.org/]
- [5] "JMeter Performance Testing" by Christopher Dock, ADMIN, issue 66, 2021, pg. 72

Author

Christopher Dock is a senior consultant at T-Systems onsite services. When he is not working on integration projects, he likes to experiment with small embedded solutions such as the Raspberry Pi or Arduino. To this end, he has authored a book to help people get started in the area of DIY electronics, *Getting Started with Arduino and Raspberry Pi*, ISBN 978-1952930027.

Listing 7: Gatling Script for BlazeMeter

```

01 package com.mypackagename
02
03 import scala.concurrent.duration._
04 import io.gatling.core.Predef._
05 import io.gatling.http.Predef._
06
07 import com.mypackagename.mySupport._
08 import com.mypackagename.FlightSteps
09
10 class BlazeBookFlight extends Simulation {
11
12   var qualifiedhost = "http://blazedemo.com"
13   var proxyhost = fetchEnvString("proxyhost","not set")
14   var proxyport = fetchEnvInt("proxyport",0)
15   var debug = 0
16
17   // http connector
18   var httpProtocol = http
19     .baseUrl(qualifiedhost)
20
21   if (proxyport > 0)
22   {
23     httpProtocol = httpProtocol.proxy(Proxy(proxyhost,proxyport))
24     println(s"proxyhost $proxyhost");
25     println(s"proxyport $proxyport");
26   }
27
28   val scn = scenario("BookFlight")
29     .exec(
30       FlightSteps.getSitePage,
31       FlightSteps.reserveFlight,
32       FlightSteps.purchaseFlight,
33       FlightSteps.confirmFlight
34     )
35     .pause(5)
36
37   setUp(
38     scn.inject(
39       constantConcurrentUsers(1).during(6.seconds),
40       rampConcurrentUsers(1).to(2).during(4.seconds)
41     ).protocols(httpProtocol)
42   )
43 }

```

Too Swamped to Surf?

Our ADMIN Online website offers additional news and technical articles you won't see in our print magazines. Subscribe today to our free ADMIN Update newsletter and receive:

- Helpful updates on our best online features
- Timely discounts and special bonuses available only to newsletter readers
- Deep knowledge of the new IT



Protect **QEMU**-based virtualized load from vulnerabilities without complicated orchestration and rebooting. [EXPLORE TUXCARE LIVE PATCHING SERVICES](#)

ADMIN
Network & Security

ADMIN Update – Hottest Links

- Network Connections in Container Environments
- Sophos Acquires Capsule in a Bid to Beef Up Linux Production Servers
- IBM Set to Acquire DevOps Startup BoxBoat
- Untangle Firewall
- Compromise an AWS Account

Highlights

Network Connections in Container Environments
Traefik promises not only to manage mesh implementations for container environments reliably, but to do so in a way that makes them enjoyable to administer. (more)

Sophos Acquires Capsule in a Bid to Beef Up Linux Production Servers
Sophos is looking to add powerful server and container security to its Adaptive Cybersecurity Ecosystem. (more)

IBM Set to Acquire DevOps Startup BoxBoat
IBM has acquired more than 11 companies over the past year and is about to pick up another with DevOps startup BoxBoat. (more)

Untangle Firewall
To what extent does the Untangle NG Firewall, where apps come together like pieces of a jigsaw, meet customer criteria for protection, usability, price, and support? (more)

Compromise an AWS Account
Malicious attackers are trying to conquer your AWS castle in the cloud. To mount a strong defense, you'll need a deeper understanding of privilege escalation and shadow admin permissions. (more)

Most Read Articles

Further Reading

- Linux Kernel 5.13 Has Been Released
- Protecting Onboard Resources with Hyper-V
- Linux Now Set to Always Reserve the First 1MB of RAM
- Logging with Sudo 1.9
- Docker and iptables

AlmaLinux
A 1:1 BINARY COMPATIBLE RHEL FORK, A FREE LINUX OS FOR THE COMMUNITY, BY THE COMMUNITY
From CentOS to AlmaLinuxOS with ZERO migration downtime
[LEARN MORE](#)

101 Cool Linux Hacks

TUNE YOUR LINUX SYSTEM
101 COOL LINUX HACKS
Tips and shortcuts for Linux geeks
By Bruce Strieder 2021

ADMIN
Network & Security

bit.ly/HPC-ADMIN-Update



Remote access with ThinLinc 4.12

Timeless Classic

The revival of terminal servers during the Covid-19 crisis depended on the IT administrator to enable home office workplaces, with the help of tools like ThinLinc, a Linux remote desktop server. By Holger Reibold

The COVID-19 pandemic is credited with being a catalyst in the area of digitalization. Processes had to be digitalized within a very short time at a speed that previously seemed impossible. Virtually overnight, the importance of the role of IT administrator grew enormously, with administrators frequently benefiting from technologies that long since seemed outdated. Terminal services, an almost forgotten technology that allows home office workplaces to be set up with a minimum of time and effort, come from an age when clients were little more than simple character displays. For administrators, they are a welcome tool, because they allow simpler administration of centrally operated applications and settings than is the case with high-maintenance desktop computers. The use of classic terminal server technologies with one (or more) central servers promises optimum utilization of a central server system instead of the often inefficient distribution to desktops. In principle, a server-based solution increases availability and, thus, ultimately security. Now 18 years old, ThinLinc is available in

version 4.12 and includes both the server-based elements (session broker, load balancer, admin interface) and the end-user components (client software, customization tools). The interaction of these different components produces a complete solution that fulfills all requirements for a modern terminal environment. In this article, I'll show you how to set up the software, adapt it to a Windows environment, and ensure access is protected.

Technical Foundation

In technical terms, ThinLinc is a Linux remote desktop server that primarily uses open source software such as TigerVNC, noVNC, OpenSSH, common Unix printing system (CUPS), and PulseAudio. From these tools, the developer, Cendio of Linköping, Sweden, developed a robust, stable environment for server-based computing. ThinLinc provides the resources of the Linux server in use by converting a regular Linux distribution into a remote desktop server, which simplifies typical tasks for administrators

because they only need to keep the server in mind, making it much easier to monitor the environment.

In principle, ThinLinc supports all Linux distributions that use the RPM or DPKG package managers. The tool's system architecture is based on a typical client-server architecture designed in such a way that the terminal server can be integrated effortlessly into an existing IT infrastructure (**Figure 1**). Apart from regular system authentication, integration of NetIQ eDirectory identity infrastructure management, Active Directory (AD), Network Information Service (NIS), and other elements is also possible. ThinLinc uses pluggable authentication modules (PAM) for authentication.

The software supports clustering and offers high availability and load balancing. To ensure high availability, two systems act as virtual systems management (VSM) servers. If a system is down, the VSM server handles requests so as to ensure that no or only minor disruptions to services occur. Load balancing distributes the user sessions uniformly to the servers in a ThinLinc cluster,

Photo by Clem Onojeghuo on Unsplash

with the servers burdened equally insofar as is possible.

The architecture is characterized by the master, agent, and client. The ThinLinc master server (`vsmserver`) is responsible for initial authentication and selection of the terminal server. To do this, it tracks all sessions and distributes the load to several agents in a cluster. Its task is also to ensure that the clients receive the relevant session information.

The tasks of the ThinLinc agent (`vs-agent`) include starting and hosting the processes from which a session is generated. The agent also establishes tunnels for graphical and local devices. The tunnels are multiplexed for each user by means of a Secure Shell (SSH) connection. Finally, the client establishes two connections: one with the master first, and then another with the agent that the master qualifies as being optimal.

Installation Preparations

Before performing the installation, you should check the Linux server system you want to use with ThinLinc to make sure it fulfills the necessary system requirements. The hardware side has no special requirements. The server configuration should be guided primarily by the requirements that users place on the environment.

To provide every user with a KDE or Gnome desktop, you need to ensure around 200MB of storage space in each case. Much more important is the availability of RPM support, SSH, and a correct Network Time Protocol (NTP) configuration.

ThinLinc is offered under a proprietary license, but the developers provide a demo version for download that is limited to five users working simultaneously. The cost for five to 10 users is to \$78 per year per user and for 11 to 49 users is \$66 per year per user. To install, download the ZIP archive from the ThinLinc website [1], unpack it into a directory of your choice, and start the installation script with:

```
sh ./install-server
```

If you have a valid license, save the relevant text file with the file extension `.license` in the `/opt/thinlinc/etc/licenses` directory. In principle, the basic system is now ready for use. You can install the printer as a next step or set up web access.

Configuring Printers with CUPS

Despite the increased acceptance of the paperless office, printing documents remains an important

function that a central server solution also needs to support. According to the project website, ThinLinc is used in particular in university and official contexts, and the option of printing is especially required in the latter case despite growing digitalization. The tool therefore relies on the renowned CUPS print server for print functionality.

ThinLinc essentially offers two variants: access to a local printer or to the nearest printer. In the first case, you have two operating modes: device-independent and device-dependent. The two can be combined.

You need to revert to the ThinLinc setup to install the PDF conversion filter, the back end, and the queue in CUPS on all machines. The setup adds a new queue called *thinlocal* to the CUPS server and makes it available to the users. The *thinlocal* printer is cluster-enabled: If a user sends a print request to a node in a ThinLinc cluster that is not hosting the user session, the print request is forwarded automatically to a suitable node.

The local printer is ready to use once the installation is complete; you only need to ensure in the ThinLinc client configuration that the diversion has been activated.

Implementing Single App Access

ThinLinc offers various possibilities that are not all recognizable at first glance. Instead of a complete desktop, you can limit the user environment to a single application – a very interesting option from a security perspective. In practice, you can limit access, for example, to a web browser. Only web-based access with Firefox is then available to employees.

For this purpose, changes are needed to the ThinLinc configuration file `opt/thinlinc/etc/conf.d`. To do so, generate a new profile by editing the `profiles.hconf` file, which could look as follows in an instance where Firefox is the only permitted application:

Name	Hardware address	Location
terminal0	01:23:45:67:89:AB	New location
terminal1	AB:AB:BC:BC:CD:CD	New location

Add new terminal

Terminal name (required):

Hardware (MAC) address (required):

Location:

Printers:

☐ Yes, really delete New terminal

Figure 1: New terminals can be created and printers shared in the admin web interface.


```
[/profiles/firefox]
name=Firefox only
description=Example configuration of 2
            single app access to ThinLinc
cmdline=${TLPREFIX}/bin/2
            tl-single-app firefox
testcmd=type firefox
```

You must also add the new profile to the order parameter:

```
order=firefox
```

After saving and restarting the ThinLinc server, the new profile is available from the profile selection.

Adapting to Windows

The option of accessing Active Directory and a Windows file server is important for integration in a heterogeneous network environment. First, you must install various components on the ThinLinc clients:

```
yum install realmd sssd 2
    oddjob oddjob-mkhomedir 2
    adcli samba-common-tools 2
    krb5-workstation
```

The next step is to activate collaboration with the Windows DNS service so

Listing 1: Windows Integration

```
[localhost]# realm discover <Domain>
DOMAIN
  type: kerberos
  realm-name: <Domain>
  domain-name: <Domain>
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: oddjob
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common-tools
```

Listing 2: /etc/pam.d/thinlinc

```
##PAM-1.0 e
Auth          required    pam_sepermit.so e
Auth          substack    password-auth e
Auth          required    pam_google_authenticator.so nullok e
Auth          include     postlogin e
account       required    pam_sepermit.so e
account       required    pam_nologin.so e
```

that the AD domain can be recorded correctly. **Listing 1** shows what this must look like. To join the domain and provide the admin password for access and verify the user resolution in Active Directory, use the commands:

```
realm join <domain>
id testuser@<domain>
```

Additionally, the system security services daemon (SSSD) configuration `sssd.conf` must be adapted to make ThinLinc recognizable as a service. To do this, edit the file `/etc/sss/sssd.conf` and add the line:

```
ad_gpo_map_remote_interactive = +thinlinc
```

The changes take effect after entering

```
systemctl restart sssd
```

to restart SSSD.

Enhancing Access Security

When accessing the ThinLinc server, you have the option of using two-factor authentication, a one-time password (OTP), or a standard password for authentication. The procedure for enabling OTP authentication for accessing ThinLinc starts with installing the Google Authenticator:

```
sudo dnf install google-authenticator
```

Next, configure SSHD so that the daemon permits authentication by editing the file `/etc/ssh/sshd_config` and activating the ChallengeResponseAuthentication option:

```
# Change to no to disable s/key passwords

ChallengeResponseAuthentication yes
```

This system modification also takes effect after a restart. The next step involves configuring the interaction between PAM and Google Authenticator. To do this, edit the file `/etc/pam.d/thinlinc`

and make the changes shown in **Listing 2**. After installing Google Authenticator on the terminal, the user still needs to activate OTP functionality on the ThinLinc server. To do this, execute the `google-authenticator` program, which opens an interactive dialog. During configuration, a QR code is output that the user scans with the terminal device to activate the functionality.

Access in a Web Client

ThinLinc also provides a web client. For this purpose, it uses noVNC, a VNC client JavaScript library. Both administrators and users benefit from the fact that noVNC **[2]** can be executed in mobile environments like Android and iOS. The advantage for the user is clear: The installation of the ThinLinc client is not absolutely necessary, and nothing hinders access by mobile terminals. Although “traditional” ThinLinc clients all use SSH for encrypting the client-server connection, the web client uses TLS as a protocol. noVNC provides HTML5 functionality. Implementation of bring-your-own-device policies is therefore also possible.

Summary

Terminal environments are experiencing a revival as a result of the COVID-19 crisis. Administrators can draw on a considerable range of commercial and free tools for solutions. However, companies must decide whether a commercial tool, such as ThinLinc, is worthwhile or whether it would be better to use a free remote desktop tool like UltraVNC. Compared directly, they are not significantly different. A ThinLinc license makes sense primarily where companies are dependent on timely support. ■

Info
[1] ThinLinc: [\[https://www.cendio.com/thinlinc/what-is-thinlinc\]](https://www.cendio.com/thinlinc/what-is-thinlinc)
[2] noVNC: [\[https://novnc.com/info.html\]](https://novnc.com/info.html)

New features in Windows Terminal Preview

Quick Shell

Windows Terminal Preview comes with interesting new features, such as state-preserving quick windows access via shortcuts and configuration of almost all settings in a GUI, which is reason enough to take a look at the preview. By Rainer W. Gerling

You can pick up both the current version of Windows Terminal and Windows Terminal Preview

from the Microsoft Store [1]. Alternatively, a manual install from GitHub [2] is also possible; make sure you choose the version with the highest release number. At the end of the description, you will find the *Assets* item, which you should expand if you do not see any file links there. A click on the line with the file name ending with `msix-bundle` starts the download of the installation package – or the installation, depending on your browser settings. If you install from the Microsoft Store, the applications are updated automatically.

For this article, I looked at the Windows Terminal Preview, which at the time was version 1.10.1933.0 with the then current Stable version at 1.9.1942.0. Both the Preview and Stable versions can be used in parallel with separate settings.

Configuration (Almost) Without an Editor

One important innovation is the graphical interface for configuring the tool. It was already included in the current version, although with significantly reduced functionality. Most settings can be made in the graphical

interface (Figure 1). Depending on the Preview version you use, some very detailed settings might still need to be edited directly in the JSON file. Whether this will change by the time the final 2.0 version is released remains to be seen. Note that the key names are localized if you change the interface language, but the JSON file always uses the English expressions. The profiles for the command prompt, PowerShell, Windows Subsystem for Linux, and Azure Cloud Shell are always generated automatically if the respective feature is available. Deleting these profiles will not

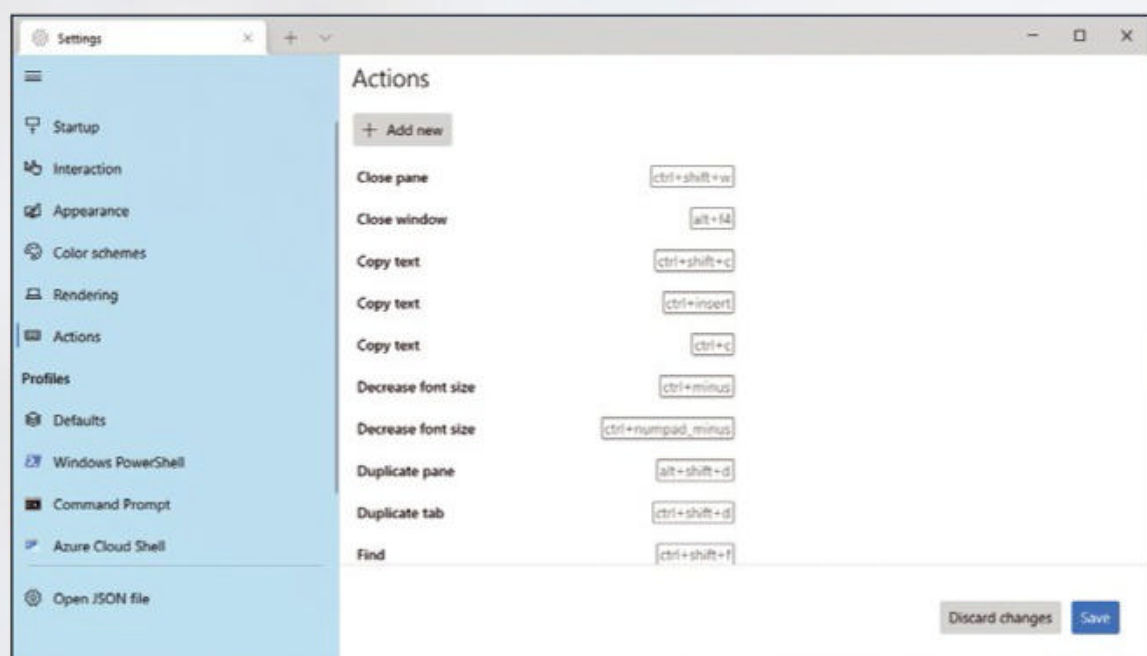


Figure 1: The graphical user interface for configuring the Windows Terminal Preview.

Listing 1: globalSummon Examples

```

01 {
02   "command":
03   {
04     "action": "globalSummon",
05     "desktop": "onCurrent",
06     "monitor": "any"
07   },
08   "keys": "ctrl+1"
09 },
10 {
11   "command":
12   {
13     "action": "globalSummon",
14     "desktop": "onCurrent",
15     "monitor": "toMouse",
16     "name": "Debian"
17   },
18   "keys": "ctrl+2"
19 }

```

do you any good because they will just be created again. However, you can disable them by enabling the *Hide profile from dropdown* option. The settings in *Defaults* apply to all profiles. Exceptions are configured in the individual *Profiles* panes. The interface is most likely not final and could still change by version 2.0, but one thing that should remain is that right-clicking on a window tab opens a menu for customizing the tab.

Quake Mode

One very practical function is the new Quake window. A keyboard shortcut can be used to start the Windows terminal with the default profile in the upper half of the screen and immediately move it to the foreground. Pressing the same shortcut causes it to disappear again. Between calls, the state remains. The default shortcut for this is Win + ` , which can cause collisions with the default key combination of the FancyZones editor from Microsoft.

As a prerequisite for the Quake window, an instance of the Windows Terminal must be running. In this context, the setting to start Windows Terminal with the user's login is very practical. The current Stable version already

supports Quake mode. In the Preview version, an icon is visible in the taskbar, but not in the Stable version.

Key Bindings Become Actions

In the *Actions* sidebar item, dubbed *Key bindings* in earlier versions, you can configure the shortcuts between the key combinations and the Windows terminal commands. They can be adjusted conveniently in the graphical configuration interface, which also gives you access to the JSON file (bottom menu item) that stores all settings. Therefore, you can easily transfer the customized actions to another computer. To define actions, select *Settings | Actions*. A click on the pencil icon (which appears when you hover over the line in question) lets you change the keyboard shortcut. After making changes, don't forget to press the *Save* button. Note that the selection accessible from the command palette in the standard menu is used for executing the commands and not for configuring the settings. You can also reach the command palette with the Ctrl + Shift + P shortcut and then select an action from the menu or type the command in a kind of internal command line.

Great Window Management

Quake mode is a special case of the `globalSummon` action that also has an effect outside the terminal because it can be used to move the last terminal window you used to the foreground. Parameters like `desktop`, `monitor`, and `name` let you control exactly what happens when you use multiple monitors, multiple virtual desktops, or both.

The `desktop` parameter supports the `any`, `toCurrent`, and `onCurrent` options. Use of the `any` option means that the shortcut changes to the desktop on which the terminal window is open; the `toCurrent` option moves the terminal window to the current desktop, no matter which desktop it is on right now; and

`onCurrent` moves the terminal window to the foreground if it is on the current desktop. Otherwise, a new window opens with the default profile on the current desktop.

The `monitor` parameter has the `any`, `toCurrent`, and `toMouse` options, where `any` keeps the terminal window on the display on which it is currently open, but moves it to the foreground; `toCurrent` drags the terminal window to the screen where the focused Windows window is, regardless on which monitor it is currently displayed; and `toMouse` moves the terminal window to the monitor that holds the mouse pointer.

If you do not specify the `name` parameter, the action refers to the last terminal window you used. Otherwise, the parameter moves the terminal window with the specified name to the screen. [Listing 1](#) shows two examples. The parameters can only be entered directly in the JSON file.

Clickable Links and Default Shell

In the interface under *Interaction*, you can use the *Automatically detect URLs and make them clickable* action to determine whether or not identified URLs will be clickable. If so, the URL can then be opened directly by control-clicking.

Up to now, the default shell in Windows is the command prompt. However, you can also set Windows Terminal Preview as the default shell. Microsoft still refers to this feature as *alpha*, and it requires a Windows 10 Developer version (22000.65 or higher) because the default shell is defined in the operating system itself and not in Windows Terminal.

If you are using a suitable Developer version, start the command prompt or Windows PowerShell and call *Command Prompt Properties* or *Windows PowerShell Properties* in the menu. When you get there, you will find a new *Terminal* tab where you can configure the terminal colors and cursor shape.

If your Windows version is not up to date, you will see an empty space in

the lower right corner (the red frame in [Figure 2](#) shows the space for the setting option). If your Windows

version is up to date, you can configure *Windows Terminal Preview* as the default shell.

Conclusions

The command prompt in the form of the terminal is far from dead and is under active development. In Windows Terminal, Microsoft has created a state-of-the-art command prompt with a wide range of configuration options. The additional integration of PowerShell, Windows Subsystem for Linux, and Azure Cloud Shell is a success. The customization options leave little to be desired. It is evident that Microsoft is looking to encourage professional users who prefer to use Linux to bind more strongly to Windows. The complete documentation for Windows Terminal (Preview) can be found online [\[3\]](#). ■

Info

- [1] Windows Terminal Preview in Microsoft Store: [\[https://www.microsoft.com/en-us/p/windows-terminal-preview/9n8g5rfz9xk3\]](https://www.microsoft.com/en-us/p/windows-terminal-preview/9n8g5rfz9xk3)
- [2] Windows Terminal Preview on GitHub: [\[https://github.com/microsoft/terminal/releases\]](https://github.com/microsoft/terminal/releases)
- [3] More information on Windows Terminal: [\[https://docs.microsoft.com/en-us/windows/terminal/\]](https://docs.microsoft.com/en-us/windows/terminal/)

The Author

Rainer W. Gerling is a theoretical physicist and a data privacy and IT security expert. He has been professionally involved in information technology for more than 40 years, and in 1986 he published one of the first articles in Germany on computer viruses. From 1993 until his retirement in 2020, he was the data privacy officer and CISO of the Max-Planck-Gesellschaft (Max Planck Society). Today he teaches Information Security at the Munich University of Applied Sciences.

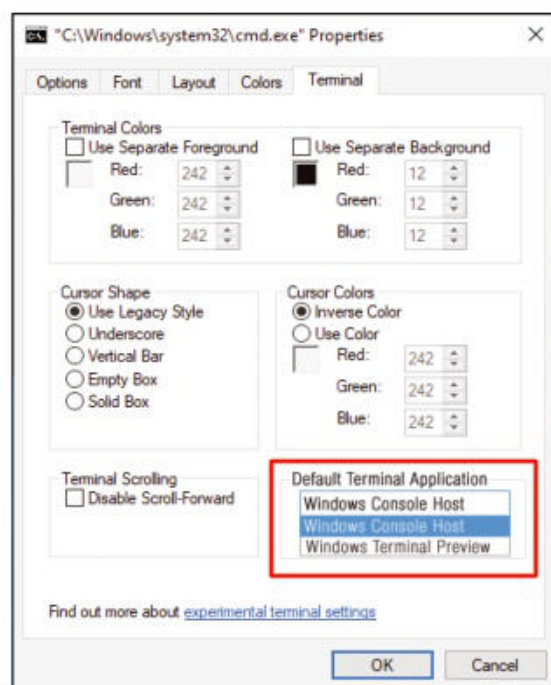
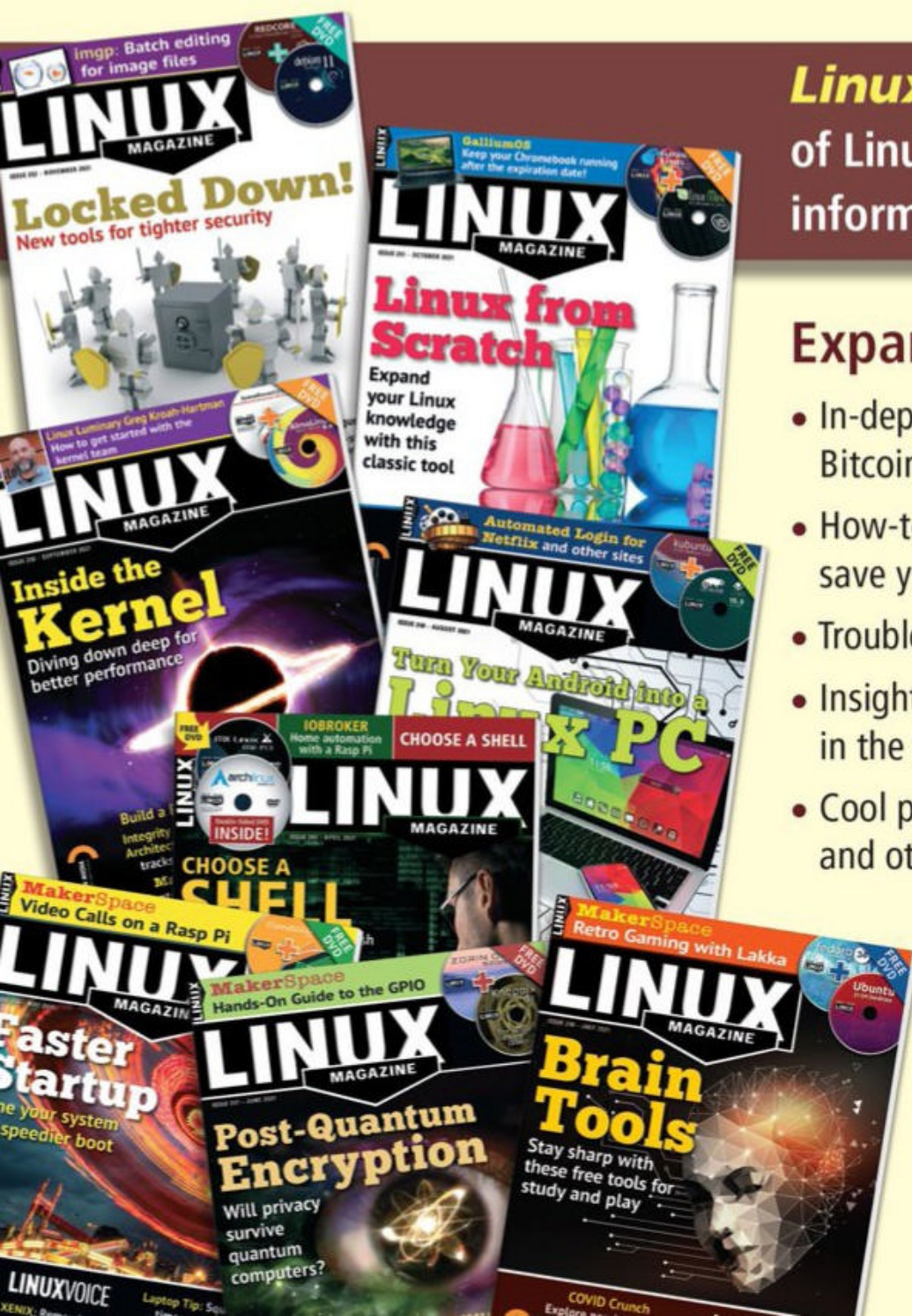


Figure 2: Although the current Windows 11 already has an option to set the default shell, it is still missing in Windows 10.



Linux Magazine is your guide to the world of Linux. Look inside for advanced technical information you won't find anywhere else!

Expand your Linux skills with:

- In-depth articles on trending topics, including Bitcoin, ransomware, cloud computing, and more!
- How-tos and tutorials on useful tools that will save you time and protect your data
- Troubleshooting and optimization tips
- Insightful news on crucial developments in the world of open source
- Cool projects for Raspberry Pi, Arduino, and other maker-board systems



If you want to go farther and do more with Linux, subscribe today and never miss another issue!

Subscribe now!
shop.linuxnewmedia.com/subs





CPU affinity in OpenMP and MPI applications

Bindings

Get better performance from your nodes by binding processes and associating memory to specific cores. Jeff Layton

It's called high-performance computing (HPC), not low-performance computing (LPC), not medium-performance computing (MPC), and not even really awful-performance computing (RAPC). The focus is doing everything possible to get the highest performance possible for your applications.

Needless to say, but I will say it anyway, processors and systems have gotten very complicated. Individual CPUs can have 64+ cores, and this number is growing. They are being packaged in different ways, including multichip modules [1] with memory controllers connected in various locations, multiple memory channels, multiple caches sometimes shared across cores, chip and module interconnections, network connections, Peripheral Component Interconnect Express (PCIe) switches, and more. These elements are connected in various ways, resulting in a complex non-uniform memory access (NUMA) [2] architecture.

To get the best possible performance, you want the best bandwidth and least latency between the processing elements and between the memory and processors. You

want the best performance from the interconnect between processing elements, the interconnect among processing and memory elements and accelerators, and the interconnect among the processors and accelerators to external networks. Understanding how these components are connected is a key step for improving application performance.

Compounding the challenge of finding the hardware path for best performance is the operating system. Periodically, the operating system runs services, and sometimes the kernel scheduler will move running processes from a particular process to another as a result. Then your carefully planned hardware path can be disrupted, resulting in poor performance. I have run all types of code on my workstation and various clusters, including serial, OpenMP, OpenACC, and MPI code. I carefully watch the load on each core with GkrellM [3], and I can see the scheduler move processes from one core to another. Even when I leave one to two cores free for system processes, with the hope that processes won't be moved, I still see the processes move from one core

to another. In my experience, when running serial code, it only stays on a particular core for a few seconds before being moved to another core. When a process move takes place, the application is "paused" while its state moves from one processor to another, which takes time and slows the application. After the process is moved, it could be accessing memory from another part of the system that requires traversing a number of internal interconnects, reducing the memory bandwidth, increasing the latency, and negatively affecting performance. Remember, it's not LPC, it's HPC. Fortunately, Linux has developed a set of tools and techniques for "pinning" or "binding" processes to specific cores while associating memory to these cores. With these tools, you can tell Linux to run your process on very specific cores or limit the movement of the processes, as well as control where memory is allocated for these cores.

In this article, I present tools you can use for binding processes. In "Processor Affinity for OpenMP and MPI" (online) [4], I show how they can be used with OpenMP and MPI applications.

Example Architecture

I'll use a simple example of a single-socket system with an AMD Ryzen Threadripper [5] 3970X CPU that has simultaneous multithreading (SMT) turned on.

A first step in understanding how the processors are configured is to use the command `lscpu`. The output of the command on the example system is shown in Listing 1. The output notes 64 CPUs and two threads per CPU, which indicates that SMT is turned on, which means 32 “real” cores and 32 SMT cores.

Also note the single socket and one NUMA node. The output also lists the L1d cache as 1MiB, the L1i cache as 1MiB, the L2 cache as 16MiB, and the L3 cache as 128MiB. However, it doesn't tell you how the caches are associated with cores.

One way to get most of this information in a more compact form is shown in Listing 2.

An important question to be answered is: Which cores are “real,” and which cores are SMT? One way is to look at the `/sys` filesystem for the CPUs:

```
-----text01 code
$ cat /sys/devices/system/cpu/cpu0/topology/thread_siblings_list
0,32
```

If the first number in the output [6] is equal to the CPU number in the command, then it's a real core. If not, it is an SMT core. For the example command, the CPU number in the command is 0 and the first number is also 0. This makes it a real core. Now try the command on a few other CPUs (Listing 3). The first command

Listing 1: `lscpu`

```
$ lscpu
Architecture:                x86_64
CPU op-mode(s):              32-bit, 64-bit
Byte Order:                  Little Endian
Address sizes:               43 bits physical, 48 bits virtual
CPU(s):                      64
On-line CPU(s) list:        0-63
Thread(s) per core:         2
Core(s) per socket:         32
Socket(s):                   1
NUMA node(s):               1
Vendor ID:                   AuthenticAMD
CPU family:                  23
Model:                      49
Model name:                  AMD Ryzen Threadripper 3970X 32-Core Processor
Stepping:                    0
Frequency boost:             enabled
CPU MHz:                     2198.266
CPU max MHz:                 3700.0000
CPU min MHz:                 2200.0000
BogoMIPS:                    7400.61
Virtualization:              AMD-V
L1d cache:                   1 MiB
L1i cache:                   1 MiB
L2 cache:                    16 MiB
L3 cache:                    128 MiB
NUMA node0 CPU(s):          0-63
Vulnerability Itlb multihit: Not affected
Vulnerability L1tf:         Not affected
Vulnerability Mds:          Not affected
Vulnerability Meltdown:     Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1:   Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:   Mitigation; Full AMD retpoline, IBPB conditional, STIBP conditional, RSB filling
Vulnerability Srbds:        Not affected
Vulnerability Tsx async abort: Not affected
Flags:                       fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good nopl nonstop_tsc c cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignss e 3dnowprefetch osvw ibs skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3 cdp_l3 hw_pstate sme ssbd mba sev ibpb stibp vmmcall fsgsbase bmi1 avx2 sme p bmi2 cqm rdt_a rdseed adx smap clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local clzero irperf xsaveerptr wbnoinvd arat npt lbr v svm_lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_omsave_vmload vgif umip rdpid overflow_recov succor smca
```

Listing 2: Compact `lscpu`

```
$ lscpu | egrep 'Model name|Socket|Thread|NUMA|CPU(s)\|
CPU(s):                64
On-line CPU(s) list:   0-63
Thread(s) per core:    2
Socket(s):             1
NUMA node(s):          1
Model name:            AMD Ryzen Threadripper 3970X 32-Core Processor
NUMA node0 CPU(s):     0-63
```

Listing 3: Real or SMT? Method 1

```
$ cat /sys/devices/system/cpu/cpu1/topology/thread_siblings_list
1,33
$ cat /sys/devices/system/cpu/cpu30/topology/thread_siblings_list
30,62
$ cat /sys/devices/system/cpu/cpu31/topology/thread_siblings_list
31,63
$ cat /sys/devices/system/cpu/cpu32/topology/thread_siblings_list
0,32
$ cat /sys/devices/system/cpu/cpu33/topology/thread_siblings_list
1,33
```


looks at CPU 1, and it's a real core (the CPU number is 1, and the first number in the output is 1, which matches). CPU 30 and 31 are also both real cores. However, when the command is run on CPU 32, the first

Listing 4: Real or SMT? Method 2

```
$ cat $(find /sys/devices/system/cpu -regex
".*cpu[0-9]+/topology/thread_siblings_list") | sort
-n | uniq
0,32
1,33
2,34
3,35
4,36
5,37
6,38
7,39
8,40
9,41
10,42
11,43
12,44
13,45
14,46
15,47
16,48
17,49
18,50
19,51
20,52
21,53
22,54
23,55
24,56
25,57
26,58
27,59
28,60
29,61
30,62
31,63
```

number in the output is 0. Because 0 does not match 32, it is an SMT core. The same is also true on CPU 33. You can also use the first number in the output for the SMT cores as the real core with which it is associated. For example, CPU 32 is associated with CPU 0 (the first number in the output). So CPU 0 is the real core and CPU 32 is the SMT core in the pair. Understanding the numbering of the real and SMT cores is important, but you have another way to check whether the CPU is real or SMT. Again, it involves examining the /sys filesystem (Listing 4). The output from the command is in pairs, listing the real CPU number first and the associated SMT CPU number last. The first line of the output says that CPU 0 is the real core and CPU 32 is the SMT CPU. Really it's the same as the previous command, except it lists all of the cores at once. The lstopo tool can give you a visual layout of the hardware along with a more detailed view of the cache layout (Figure 1). This very useful command returns the hardware layout of your system. Although it can include PCIe connections as well, I've chosen not to display that output. Notice in the figure that each 16MB L3 cache has four groups of two cores. The first core in each pair is the real core and the second is the SMT core. For example, Core L#0 has two processing units (PUs), where PU L#0 is a real core listed as P#0 and PU L#1 is the SMT core listed as P#32. Each group of two cores has an L2

cache of 512KB, an L1d (data) cache of 32KB, and a L1i (instruction) cache of 32KB. The eight L3 cache "groups" make a total of 64 cores with SMT turned on.

Affinity Tools

In this article, I discuss two Linux tools that allow you to set and control application threads (processes), giving you great flexibility to achieve the performance you want. For example, a great many applications need memory bandwidth. The tools allow you to make sure that each thread gets the largest amount of memory bandwidth possible. If network performance is critical to application performance (think MPI applications), with these tools, you can bind threads to cores that are close to a network interface card (NIC), perhaps not crossing a PCIe switch. Alternatively, you can bind processes to cores that are as close as possible to accelerators to get the maximum possible PCIe bandwidth. The Linux tools presented here allow you to bind processes and memory to cores; you have to find the best way to use these tools for the best possible application performance.

taskset

The taskset command [7] is considered the most portable Linux way of setting or retrieving the CPU affinity (binding) of a running process (thread). According to the taskset man

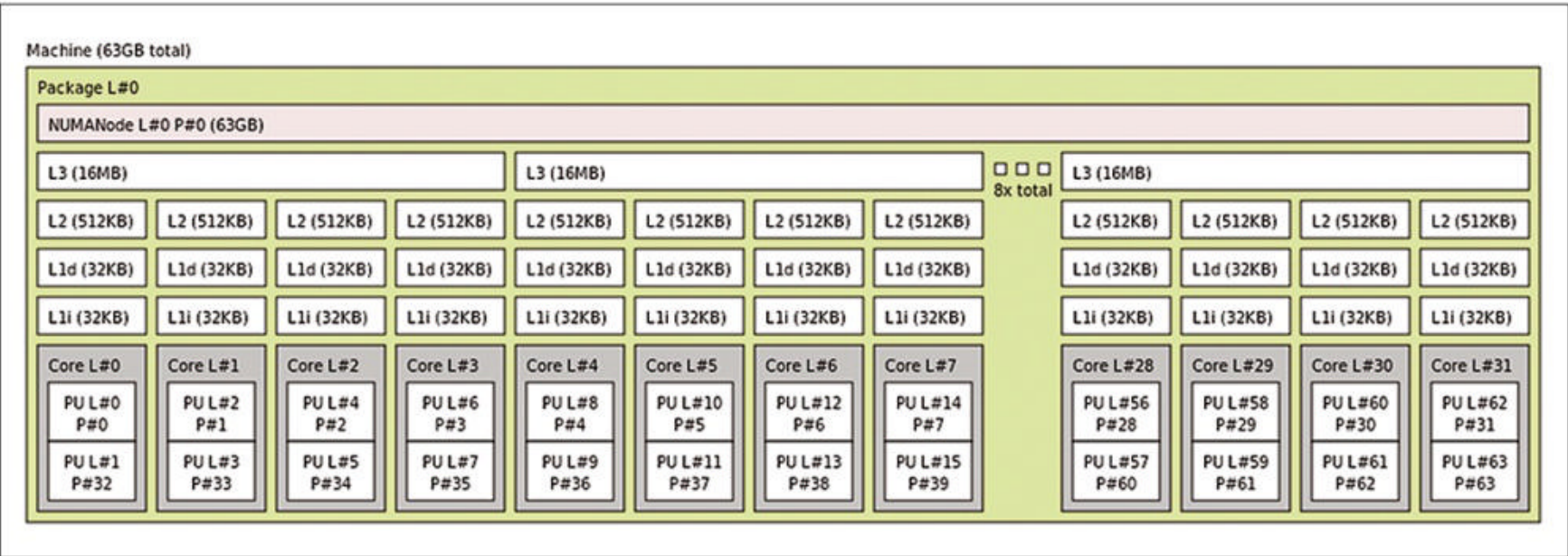


Figure 1: lstopo output for sample systems.

page, “The Linux scheduler will honor the given CPU affinity and the process will not run on any other CPUs.”

An example of executing a process with the `taskset` command is:

```
-----text02
$ taskset --cpu_list 0,2 application.exe
```

This command sets the affinity of `application.exe` to cores 0 and 2 and then executes it. You can also use the short version of the `--cpu_list` option, `-c`. If you want to change the affinity of a running process, you need to get the process ID (PID) of the processes with the `--pid (-p)` option. For example, if you have an application with four processes (or four individual processes), you get the PIDs of each process and then run the following command to move them to cores 10, 12, 14, and 16:

```
-----text03
$ taskset --pid --cpu_list 10 [pid1]
$ taskset --pid --cpu_list 12 [pid2]
$ taskset --pid --cpu_list 14 [pid3]
$ taskset --pid --cpu_list 16 [pid4]
```

numactl

One key tool for pinning processes is `numactl` [8], which can be used to control the NUMA policy for processes, shared memory, or both. One key thing about `numactl` is that, unlike `taskset`, you can't use it to change the policy of a running application. However, you can use it to display information about your NUMA hardware and the current policy (Listing 5). Note for this system, SMT is turned on, so the output shows 64 CPUs.

The system has one NUMA node (available: 1 nodes), and all 64 cores are associated with that NUMA node. Because there is only NUMA node, the node distances from NUMA node 0 to NUMA node 0 is listed as 10, which indicates it's the same NUMA node. The output from the command also indicates it has 64GB of memory (node 0 size: 64251 MB).

The advantages of `numactl` come from its ability to place and bind processes,

particularly in relation to where memory is allocated, for which it has several “policies” that are implemented as options to the command:

- **The `--interleave=<nodes>` policy** has the application allocate memory in a round-robin fashion on “nodes.” With only two NUMA nodes, this means memory will be allocated first on node 0, followed by node 1, node 0, node 1, and so on. If the memory allocation cannot work on the current interleave target node (node x), it falls back to other nodes but in the same round-robin fashion. You can control which nodes are used for memory interleaving or use them all:

```
-----text04
$ numactl --interleave=all application.exe
```

This example command interleaves memory allocation on all nodes for `application.exe`. Note that the sample system in this article has only one node, node 0, so all memory allocation uses it.

- **The `--membind=<nodes>` policy** forces memory to be allocated from the list of provided nodes (including the `all` option):

```
-----text05
$ numactl --membind=0,1 application.exe
```

This policy causes `application.exe` to use memory from node 0 and node 1. Note that a memory allocation can fail if no more memory is available on the specified node.

- **The `cpunodebind=<nodes>` option** causes processes to run only on the CPUs of the specified node(s):

```
-----text06
$ numactl --cpunodebind=0 --membind=0,1 application.exe
```

This policy runs `application.exe` on the CPUs associated with node 0 and allocates memory on node 0 and node 1. Note that the Linux scheduler is

free to move the processes to CPUs as long as the policy is met.

- **The `--physcpubind=<CPUs>` policy** executes the process(es) on the list of CPUs provided:

```
-----text07
$ numactl --physcpubind=+0-4,8-12 application.exe
```

You can also say `all`, and it will use all of the CPUs. This policy runs `application.exe` on CPUs 0-4 and 8-12.

- **The `--localalloc` policy** forces allocation of memory on the current node:

```
-----text08
$ numactl --physcpubind=+0-4,8-12 --localalloc application.exe
```

This policy runs `application.exe` on CPUs 0-4 and 8-12, while allocating memory on the current node.

- **The `--preferred=<node>` policy** causes memory allocation on the node you specify, but if it can't, it will fall back to using memory from other nodes. To set the preferred node for memory allocation to node 1, use:

```
-----text09
$ numactl --physcpubind=+0-4,8-12 --preferred=1 application.exe
```

This policy can be useful if you want to keep `application.exe` running, even if no more memory is available on the current node.

Listing 5: numactl

```
$ numactl --hardware
available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
              18 19 20 21 22 23 24 25 26 27 28 29 30 31
              32 33 34 35 36 37 38 39 40 41 42 43 44 45
              46 47 48 49 50 51 52 53 54 55 56 57 58 59
              60 61 62 63
node 0 size: 64251 MB
node 0 free: 60218 MB
node distances:
node 0
0: 10
```


Listing 6: numactl --show

```
$ numactl --show
policy: default
preferred node: current
physcpubind: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
              19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
              34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
              49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
cpubind: 0
nodebind: 0
membind: 0
```

To show the NUMA policy setting for the current process, use the `--show` (`-s`) option:

```
$ numactl --show
```

Running this command on the sample system produces the output in [Listing 6](#).

The output is fairly self-explanatory. The policy is *default*. The preferred

NUMA node is the current one (this system only has one node). It then lists the physical cores (`physcpubind`) that are associated with the current node, the bound CPU cores (`node 0`), and to which node memory allocation is bound (again, `node 0`). The next examples show some `numactl` options that define commonly used policies. The first example focuses on running a serial

```
c506-022[skx](1009)$ show_affinity all
pid      Exe_Name      tid      Affinity
233493    slurm_script      233493    0-95
233500    sleep              233500    0-95
233591    sshd               233591    0-95
233592    bash               233592    0-95
233824    ibrun              233824    0-95
233874    mpiexec.hydra      233874    0-95
233875    pmi_proxy          233875    0-95
233879    namd2_skx          233879    0
                233900    0,2,4,6,8,10,12,14,16,18,20,22,48,50,52,54,56,58,60,62,64,66,68,70
                233906    2
                233912    4
                233916    6
                233918    8
                233921    10
                233922    12
                233926    14
                233929    16
                233932    18
                233935    20
                233938    22
                233947    2
233880    namd2_skx          233880    24
                233899    24,26,28,30,32,34,36,38,40,42,44,46,72,74,76,78,80,82,84,86,88,90,92,94
                233905    26
                233909    28
                233910    30
                233914    32
                233917    34
                233919    36
                233923    38
                233930    40
                233933    42
                233936    44
                233939    46
                233950    26
233881    namd2_skx          233881    1
                233902    1,3,5,7,9,11,13,15,17,19,21,23,49,51,53,55,57,59,61,63,65,67,69,71
                233903    3
                233908    5
                233911    7
                233915    9
                233925    11
                233927    13
                233942    15
                233943    17
                233944    19
                233945    21
                233946    23
                233951    3
233882    namd2_skx          233882    25
                233901    25,27,29,31,33,35,37,39,41,43,45,47,73,75,77,79,81,83,85,87,89,91,93,95
                233904    27
                233907    29
                233913    31
                233920    33
                233924    35
                233928    37
                233931    39
                233934    41
                233937    43
                233940    45
                233941    47
                233949    27
```

Figure 2: Output of TACC `show_affinity` tool (used with permission from the GitHub repository owner).

application – in particular, running the application on CPU 2 (a non-SMT core) and allocating memory locally:

```
-----text10
$ numactl --physcpubind=2 --localalloc
application.exe
```

The kernel scheduler will not move `application.exe` from core 2 and will allocate memory using the local node (node 0 for the sample system). To give the kernel scheduler a bit more freedom, yet keep memory allocation local to provide the opportunity for maximum memory bandwidth, use:

```
-----text11
$ numactl --cpunodebind=0 --membind=0
application.exe
```

The kernel scheduler can move the process to CPU cores associated with node 0 while allocating memory on node 0. This policy helps the kernel adjust processes as it needs, without sacrificing memory performance too much. Personally, I find the kernel scheduler tends to move things around quite often, so I like binding my serial application to a specific core; then, the scheduler can put processes on other cores as needed, eliminating any latency in moving the processes around.

Tool for Monitoring CPU Affinity

Both `taskset` and `numactl` allow you to check on any core or memory bindings. However, sometimes they aren't enough, which creates an opportunity for new tools. A good affinity

monitoring tool, `show_affinity` [9], comes from the Texas Advanced Computing Center (TACC).

The tool shows "... the core binding affinity of running processes/threads of the current user." The GitHub site has a simple, but long, output example from running the command (Figure 2).

Summary

Today's HPC nodes are complicated, with huge core counts, distributed caches, various memory connections, PCIe switches with connections to accelerators, and NICs, making it difficult to clearly understand where your processes are running and how they are interacting with the operating system. This understanding is extremely critical to getting the best possible performance, so you have HPC and not RAPC.

If you don't pay attention to where your code is running, the Linux process scheduler will move them around, introducing latency and reducing performance. The scheduler can move processes into non-optimal situations, where memory is used from a different part of the system, resulting in much-reduced memory bandwidth. It can also cause processes to communicate with NICs across PCIe switches and internal system connections, again resulting in increased latency and reduced bandwidth. This is also true for accelerators communicating with each other, with NICs, and with CPUs.

Fortunately, Linux provides a couple of tools that allow you to pin (also called binding or setting the affinity of) processes to specific cores along

with specific directions on where to allocate memory. In this way, you can prevent the kernel process scheduler from moving the processes or at least control where the scheduler can move them. If you understand how the systems are laid out, you can use these tools to get the best possible performance from your application(s).

In this article, I briefly introduced two tools along with some very simple examples of how you might use them, primarily on serial applications. ■

Info

- [1] Multichip Modules: [\[https://en.wikipedia.org/wiki/Multi-chip_module\]](https://en.wikipedia.org/wiki/Multi-chip_module)
 - [2] Non-Uniform Memory Access (NUMA): [\[https://en.wikipedia.org/wiki/Non-uniform_memory_access\]](https://en.wikipedia.org/wiki/Non-uniform_memory_access)
 - [3] GkrellM: [\[http://gkrellm.srcbox.net/\]](http://gkrellm.srcbox.net/)
 - [4] "Processor Affinity for OpenMP and MPI" by Jeff Layton: [\[https://www.admin-magazine.com/HPC/Articles/Processor-Affinity-for-OpenMP-and-MPI\]](https://www.admin-magazine.com/HPC/Articles/Processor-Affinity-for-OpenMP-and-MPI)
 - [5] AMD Ryzen Threadripper: [\[https://www.amd.com/en/products/cpu/amd-ryzen-threadripper-3970x\]](https://www.amd.com/en/products/cpu/amd-ryzen-threadripper-3970x)
 - [6] First number in the output: [\[https://stackoverflow.com/questions/7274585/linux-find-out-hyper-threaded-core-id\]](https://stackoverflow.com/questions/7274585/linux-find-out-hyper-threaded-core-id)
 - [7] Taskset command: [\[https://man7.org/linux/man-pages/man1/taskset.1.html\]](https://man7.org/linux/man-pages/man1/taskset.1.html)
 - [8] `numactl`: [\[https://linux.die.net/man/8/numactl\]](https://linux.die.net/man/8/numactl)
 - [9] `show_affinity`: [\[https://github.com/TACC/show_affinity\]](https://github.com/TACC/show_affinity)
-

The Author

Jeff Layton has been in the HPC business for almost 25 years (starting when he was 4 years old). He can be found lounging around at a nearby Frys enjoying the coffee and waiting for sales.

Clustering with the Nutanix Community Edition

The Right Track

The free Community Edition of the Nutanix hyperconverged infrastructure, Nutanix on-premises cloud, is offered alongside its commercial product for those looking to take their first steps in the environment. By Günter Baumgart

To be clear, the Community Edition of Nutanix was developed for testing purposes only; it is not a replacement for the production version. The Community Edition does not give you all the possibilities that you have with the commercial version. For example, the Community Edition only supports two hypervisors: Acropolis (AHV) by Nutanix and ESXi by VMware. The basic setup of a private enterprise cloud from Nutanix built on the Community Edition includes the hypervisor, the Controller Virtual Machine (CVM) and associated cloud management system, the Prism element for single-cluster management, and Prism Central for higher level multicluster management (**Figure 1**).

With the Community Edition, you can set up a one-, three-, or four-node cluster. All other conceivable cluster combinations are reserved exclusively for the commercial version. The individual components of the Community Edition, such as AHV; the AOS cloud operating system, which is based on the individual CVMs in the cluster; and the cloud management system, cannot be mixed with components of the production version. Therefore,

you cannot manage a Community Edition cluster with Prism Central from the production version. Conversely, you cannot use Prism Central Community Edition to manage a production cluster.

If you want to use VMware's ESXi in the Community Edition as your hypervisor, also remember that you will

then not be able to use the Nutanix Flow microsegmentation functionality because it can only be used in conjunction with AHV.

Everything's Connected

During the installation and subsequent testing of the Community

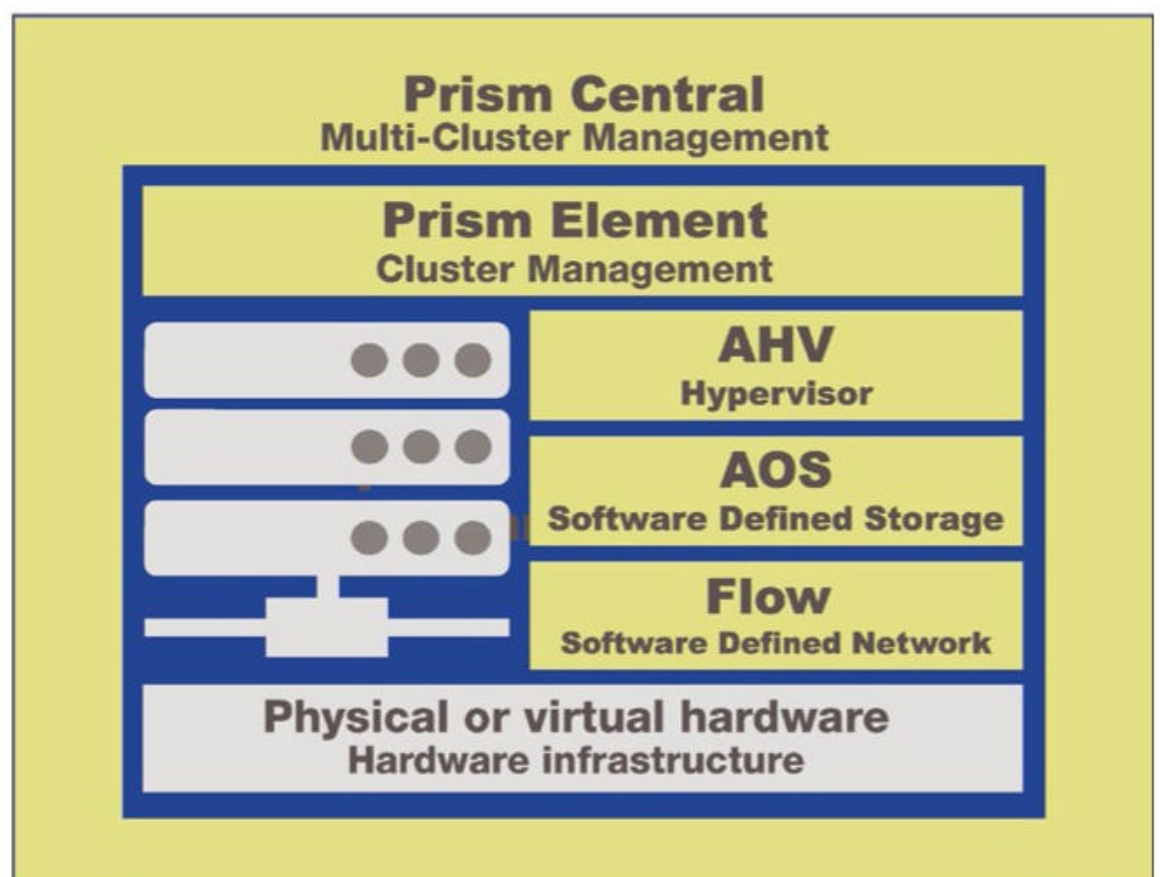


Figure 1: A schematic representation of the components of the management environment.

Photo by Beau Runsten on Unsplash

Edition, it can be quite useful to switch to the command line from time to time. To do this, you need to know how to find your way around the network and which module and service you can reach. Regardless of whether you are using AHV or ESXi, you always have at least two networks: an internal network that is not connected to a physical network adapter and an external network to which the existing physical adapters are connected. The internal network is used to support communication between the CVM and the hypervisor. The 192.168.5.0 network is used for this purpose.

The hypervisor always has the IP address 192.168.5.1 and the CVM the IP address 192.168.5.2, which means the installation process always creates two virtual bridges or virtual switches for each node in the cluster. If you use AHV, you will find *vir br0* and *br0* in the node, which for ESXi are *vSwitchNutanix* and *vSwitch0*.

You assign external IP addresses to the CVM and the hypervisor during the install. If you now want to access the console of the AHV, you can either address it on the external network or the internal network. The same applies to the console of the CVM: You can access the CVM console from the external or internal network (**Figure 2**).

Table 1 provides an overview of the accounts you can use to access the system, including the root login name for accessing the console on the hypervisor and the *nutanix* login name for the CVM console, along with the matching password *nutanix/4u*, which you also need to log on to the respective consoles.

Installation Media

In the first step, you need to create an account with Nutanix [1] and register your email address by following the *Get Access Today!* link. After you have completed the registration process, you have a personal Nutanix account and are now authorized to log in to the portal [2].

On the Nutanix Community Portal site you will see the *Download Nutanix Community Edition* block. After clicking on this, the Community Edition download site pops up immediately, and you are treated to an initial overview of the binaries available to you there. At press time, version CE-2020.09.16 was available. Because a new production version was recently released (AOS LTS 5.20 and AOS STS 6.0), it can be assumed that a new Community version will soon follow. To install the Community Edition (CE), you need to download the corresponding ISO file (CE-202y.mm.dd.iso). You can use this image to install the CVM and AHV on your nodes in a fully automated process. If you would rather use ESXi as the hypervisor in

your Nutanix Lab cluster, you also need the image of the vSphere hypervisor (ESXi ISO).

If you want to install and set up Prism Central after your cluster has been installed and configured, you need the matching binary (i.e., the *Prism Central Deployment* file) in the form of a TAR archive and the *Metadata for AOS upgrade and PC deploy/upgrade* file as a ZIP file. In addition to the JSON files for upgrades, the latter also contains the *ce-pc-deploy-202y.mm.dd-metadata.json* file, which you need to install Prism Central.

Next, download the VirtIO drivers and, if you want to try out End User Computing (EUC) or Virtual Desktops (VDIs) on the Community Edition,

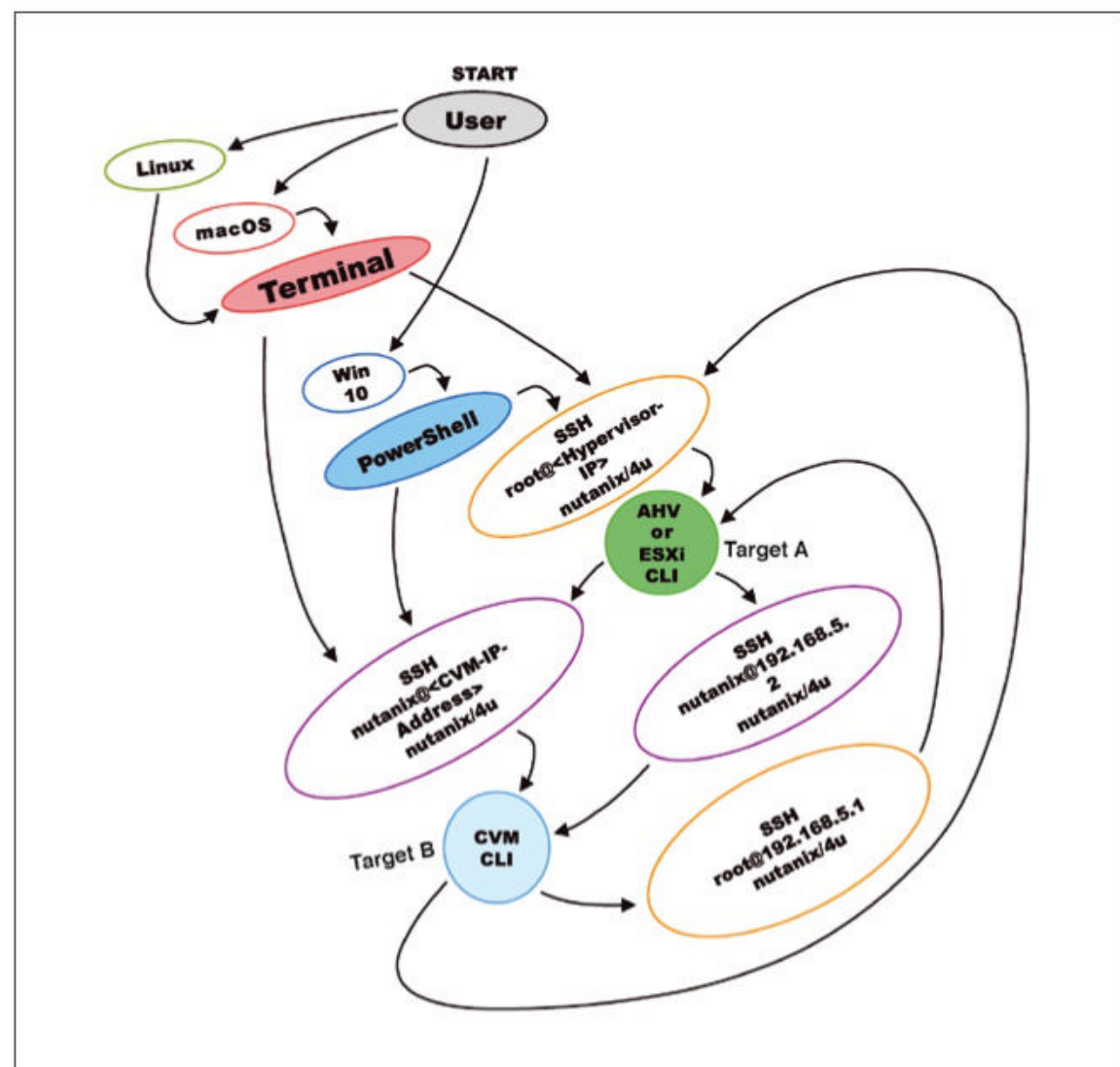


Figure 2: Different approaches lead to the CVM and hypervisor consoles.

Table 1: Nutanix Usernames

Component	Protocol	Password	Username
Controller VM	SSH	<i>nutanix</i>	<i>nutanix/4u</i>
AHV	SSH	<i>root</i>	<i>nutanix/4u</i>
ESXi	SSH	<i>root</i>	<i>nutanix/4u</i>
Prism Element	HTTPS (port 9440)	<i>admin</i>	<i>nutanix/4u</i>
Prism Central	HTTPS (port 9440)	<i>admin</i>	<i>nutanix/4u</i>

the matching plugins. In the *Documentation and Guides* section on this site you will also find bundles of additional documentation on the Community Edition in the form of PDFs and video files.

Installation Preparation

You have now downloaded all the software you need. The question that remains is how to install the Community Edition: physically or virtually (i.e., in a nested setup)? You also need to decide whether you want a one-, three-, or four-node cluster. No matter what you ultimately decide, the installation procedure is always the same. In the first iteration I take a look at creating a one-node cluster lab based on the Community Edition with Nutanix AHV as the hypervisor, Prism Central for multicluster management, and an Intel NUC (Next Unit of Computing, a small-form-factor barebone computer) mini-PC as the hardware platform. The NUC used in our lab is the NUC8 i7 BEH model. It comes with two

32GB DDR4-2666 SO-DIMMs (i.e., a total of 64GB of RAM). The computer has two disk drives – one 512GB and one 1TB SSD – and an eighth generation quad-core Intel Core i7 8559U processor running at 2.7GHz. Therefore, the machine is not totally up-to-date but is still perfectly adequate for the lab. If you do not have a machine like this at hand for your installation, use something with similar hardware, or if you are going for a nested setup in your lab, use something with similar specs for the VM.

To avoid wasting time while installing your lab setup, you should have all the necessary information ready in advance: a DNS server, a default gateway, at least two Network Time Protocol (NTP) servers, and – if you want to connect your lab to Active Directory – access credentials. You also need an IP address from your lab network for the CVM, another for the one-node-cluster itself, one in case you want to provide an iSCSI target with Nutanix volumes, another IP address (for the hypervisor, of course), and yet another IP address for Prism Central. (See the “IP Addresses for Larger Clusters” box.) Additionally, you need unique names for the Nutanix cluster and for Prism Central.

IP Addresses for Larger Clusters

If you are more interested in installing a three- or four-node cluster, remember that you will need separate IP addresses for each individual hypervisor and CVM that resides on your cluster's nodes.

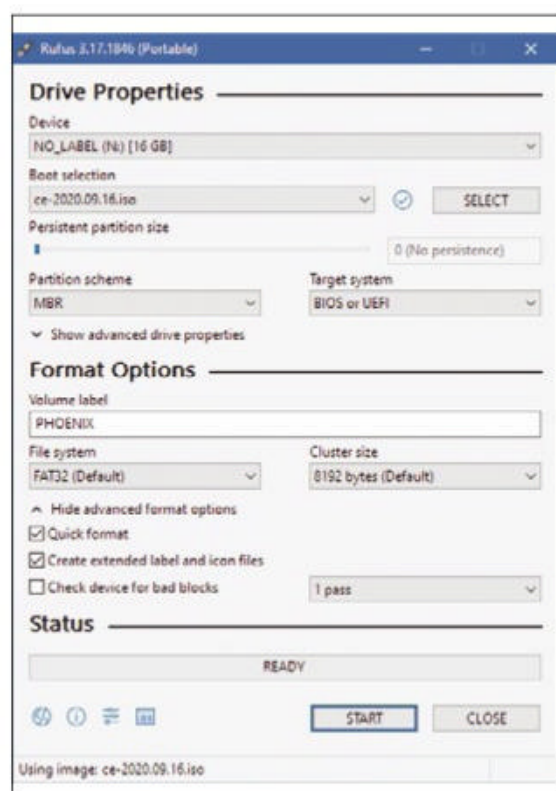


Figure 3: Generate a USB CE installation device with the `ce-2020.09.16.iso` file.

virtual disk with a capacity of 32GB as the installation target for the VM.

Setting up a Local Nutanix Cloud

After the preparatory work is done, you can move on to installing the Community Edition on the Intel NUC. Insert the two USB sticks into the corresponding server ports and switch the server on. You can now follow the installation process on the monitor connected to the NUC and, depending on the type of hardware you are using in your lab, the *Nutanix Community Edition Installer* configuration front end appears, sooner or later, where you set up your Nutanix one-node cluster.

First, select the hypervisor you want to use in your cluster. If you go for Nutanix AHV (**Figure 4**, step 1), you can continue directly with the disk assignments (**Figure 4**, step 2) because AHV is an integral part of the CE image. If you decide to use ESXi as your hypervisor, you need to provide your ESXi installation image over HTTP (e.g., in the form `http://<webserver>/iso/esxi.iso`). Because I am using AHV as the hypervisor for this workshop, I check AHV in the selection box in step 1.

You can now see all the storage devices found on the server. As you can see, `sdd` was selected as the USB installation target for the hypervisor, and the CVM will be installed on `sda`. In the fields selected in step 3, you now need to assign the address data for the hypervisor's external network. In step 4, you can enter the external address data for the controller virtual machine (VM), and in step 5, you are given the option of having the cluster created automatically by the installation process. This step is fine if you want to create a one-node cluster, but if you want to create a three-node cluster, for example, it might make more sense to create the cluster manually after successfully completing the installation of all the nodes by typing the following on the command line of a controller VM:

Creating a USB Installation Stick

To install the server, you need two USB sticks. One USB stick is used for the installation, and the second is used as an installation target or boot device. As far as the capacity of the sticks is concerned, 32GB will do nicely. To create a bootable USB stick, you can use the USB installer of your choice (e.g., Rufus [3]).

Now take the downloaded `ce-2020.09.16.iso` file and create a bootable USB CE installation stick with one of the two USB sticks (**Figure 3**). If you plan to install the Community Edition nested, this step is not necessary, of course, because you can mount the CE image directly on the virtual hardware. The second USB stick would not be necessary either because you can simply add another

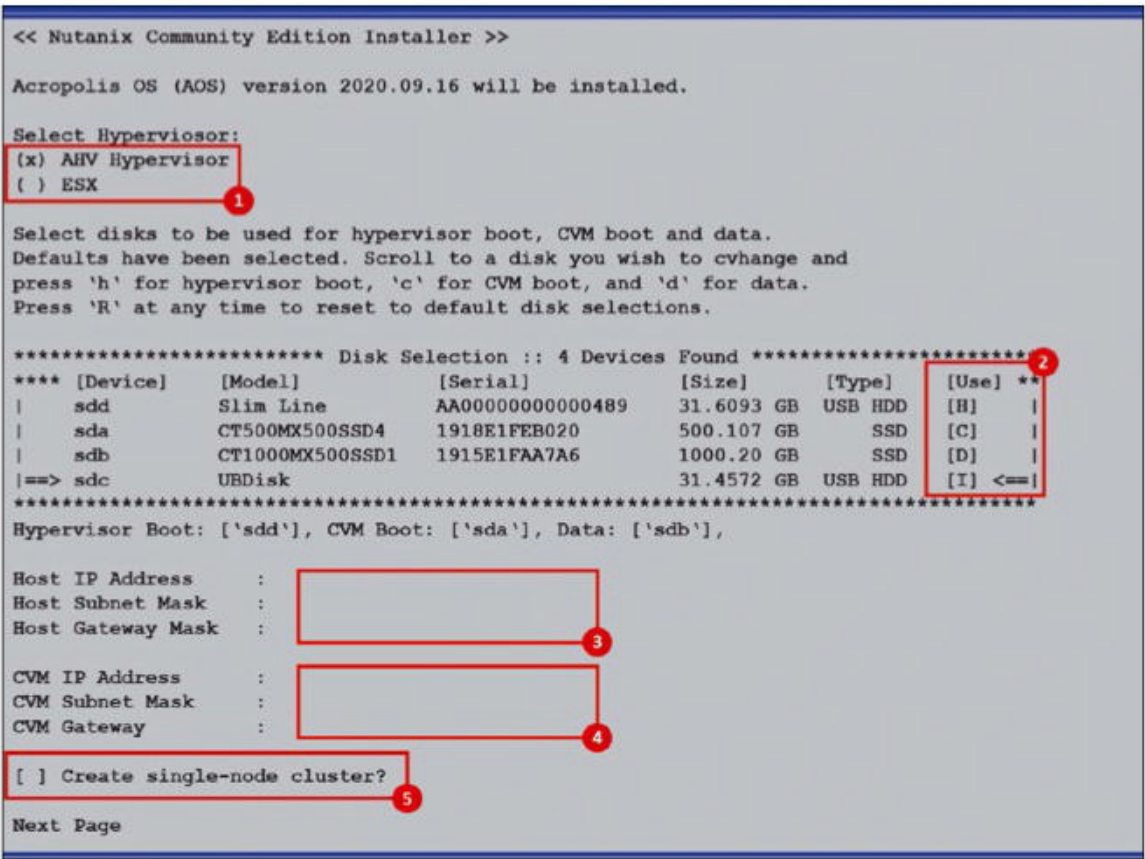


Figure 4: Entering the parameters for the cluster configuration.

```
cluster -s <cvm_ip-1>, <cvm_ip-2>, 2
<cvm_ip-3> [--redundancy_factor=2] 2
create
```

You do not need to start the cluster when you are creating a new cluster; the cluster starts up automatically after creation. However, the reason you should not rely on this automation, but build the cluster retroactively, is that if the installation of just one node fails for some reason, you need to create the whole cluster again, and this is overhead you would like to avoid. Once you have completed the entries and set up everything, click *Next* to be taken to the End User Licence Agreement (EULA). After accepting the terms of use, press *Start* and the installation begins. You can now follow the entire installation process onscreen from the console output. My entire installation and the subsequent startup of the service took a good 30 minutes with the hardware used in this example, which is actually quite quick if you consider that you are installing and configuring the complete Nutanix CE one-node hyperconverged infrastructure (HCI). Now remove the USB installation stick when the system prompts you to do so, and press *Y* at the console to reboot, and you

can proceed to create the cluster. Because the option *Create single-node cluster?* was not selected for this installation, this is the next and final step after the restart. To use SSH to connect to the CVM and create the cluster, enter:

```
ssh nutanix@<IP address of CVM>
cluster -s <cvm_ip> create
```

You have to wait until the system has created the cluster and started up all the services. Again, some patience is required. To check the progress, type:

```
cluster status
```

After all cluster processes have started up and you have seen a

Success message in the output, log on to the Prism Element UI process of the CVM on [https:// < IP address of the CVM > :9440/](https://<IP address of the CVM>:9440/). For the first login, use the *admin* account with the *nutanix/4u* password. The user interface now expects you to enter a new password. After doing so, log on again with the new password. In the next step, Prism Element then expects the NEXT account to be set up. The Prism Element splash page then appears. Now click on the cogwheel (Settings) at the top in the right-hand corner of the user interface (UI) and select *Network Configuration | Create Network* to create a new network. I used *vlan.0* for the network name and *0* as the VLAN ID.

You could now start installing your VMs on the platform. A short live demo introducing Prism Element can be found online [4]. For now, you should do some fine tuning such as assigning cluster names, cluster IP addresses, and the IP address for the iSCSI target *Nutanix-Volumes*. You can also do all of this at the command line (Table 2). More commands and scripts are located online [5], or simply enter *nccli* or *accli* at the command line of the CVM and press the Tab key to delve more deeply into the individual command references. Next, click on the *Unnamed* item in the UI (Figure 5) and enter a name for your cluster and the corresponding IP addresses in the individual fields of the form. Once that's done, you can explore and use your cloud in your lab.

Table 2: Useful Commands	
Function	Command
Start the cluster	cluster start
Stop the cluster	cluster stop
Delete the cluster	cluster destroy
Display the cluster status	cluster status
Create a one-node cluster	cluster -s <CVM_IP_adress> create
Enter a DNS server	cluster -dns_servers=<DNS-IP-1>, <DNS-IP-2> create
Enter an NTP server	cluster -ntp_servers=<NTP_server> create
Define the cluster name	cluster -cluster_name=<cluster_name> create
Assign the cluster IP address	cluster -cluster_external_ip=<cluster_IP_address> create

Prism Central and Other Features

Now that you have reached this point of your installation, you have your first Nutanix test cluster. The creators of the Community Edition promise that new improvements are continually being incorporated into the test platform. You can check out its progress by updating the system through the Life Cycle Manager (LCM). Additional information on LCM can be found online [6], or simply go to the Prism Element UI and click on *Home | LCM*. You will then be guided by the system and provided the necessary information, such as the Nutanix knowledge base (KB) articles. Once you have familiarized yourself with the platform and tested your own workloads extensively with the Community Edition, take the next step and test the other Nutanix products on your CE HCI cluster. First, install Prism Central, which is the basis for many other products. Deploying Prism Central requires only a few

steps. To begin, log on to the Prism Element UI. Top left in the browser you will then see a box labeled *Prism Central*. To open a form where you can upload the Prism Central binaries `ce-pc-deploy-2020.09.16-metadata.json` and `ce-pc-deploy-2020.09.16.tar` that you previously downloaded from the CE Community site, click on *Register or create new*.

After the upload completes, click *Install*, then select whether you want a clustered installation and whether you want to roll out a *LARGE* or a *SMALL* environment. Next, enter the IP address, the gateway, and at least one DNS server and click on *Deploy* to roll out the Prism Central VM in the cluster. Once the installation process is complete, as shown by the task display in Prism Element, register your Prism Central with your NEXT account.

To do so, log on to your new Prism Central at `https:// <IP address Prism Central> :9440` with the *admin* account and *nutanix/4u* as the password. You will recall the initial login

to the Prism Element: It's exactly the same procedure here. Now move on to the CE Cluster registration in Prism Central by going to your Prism Element (`https:// <IP address of CVM> :9440`), clicking on *Register or create new*, and selecting *Connect*. Here, you enter the IP address, login name, and password of Prism Central and click on *Connect*; hey, presto, the cluster is registered (Figure 6).

Now that Prism Central is available, you can move on to test the scalable file server, the similarly scalable object store, or S3 storage from Nutanix, or you can take a closer look at the micro-segmentation solution, Flow. If you want to familiarize yourself with automating workloads or work processes, Calm is certainly a must for you, or you can go one step further and test Karbon.

Karbon lets you roll out complete Kubernetes clusters within the Nutanix platform in an automated process. If you are also interested in DIY automation, you have massive opportunities for programming with `accli`, `nccli`,

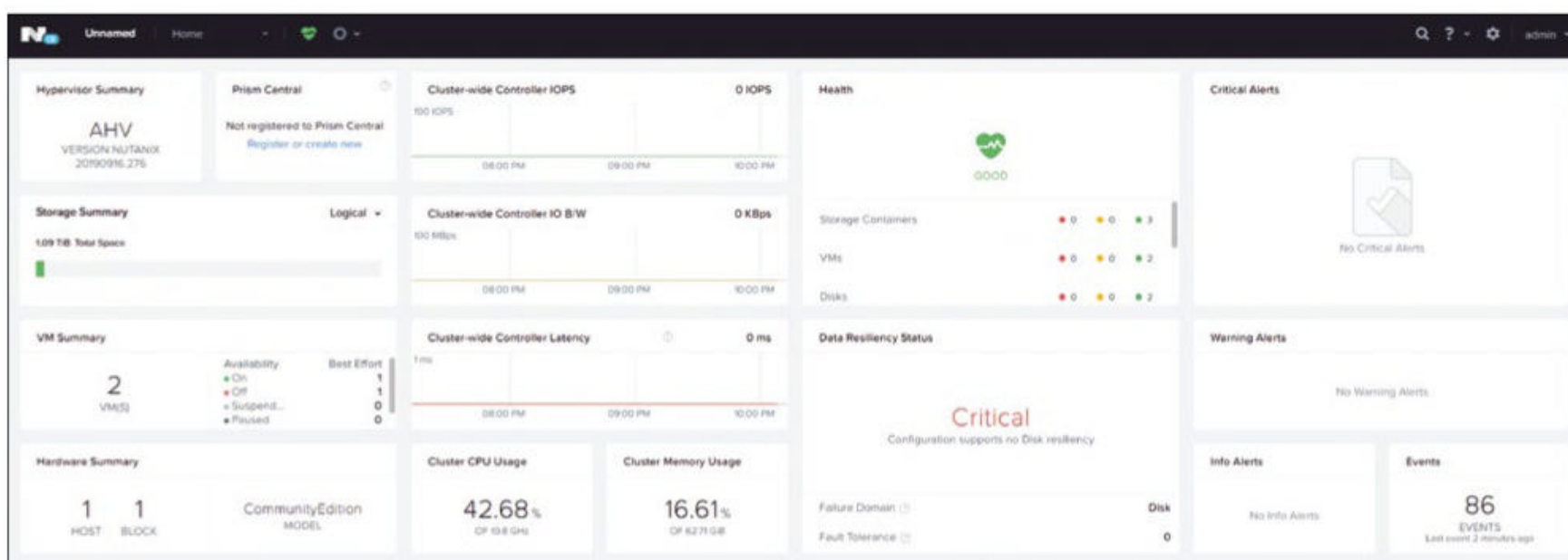


Figure 5: The main dashboard of Prism Element with the home site.

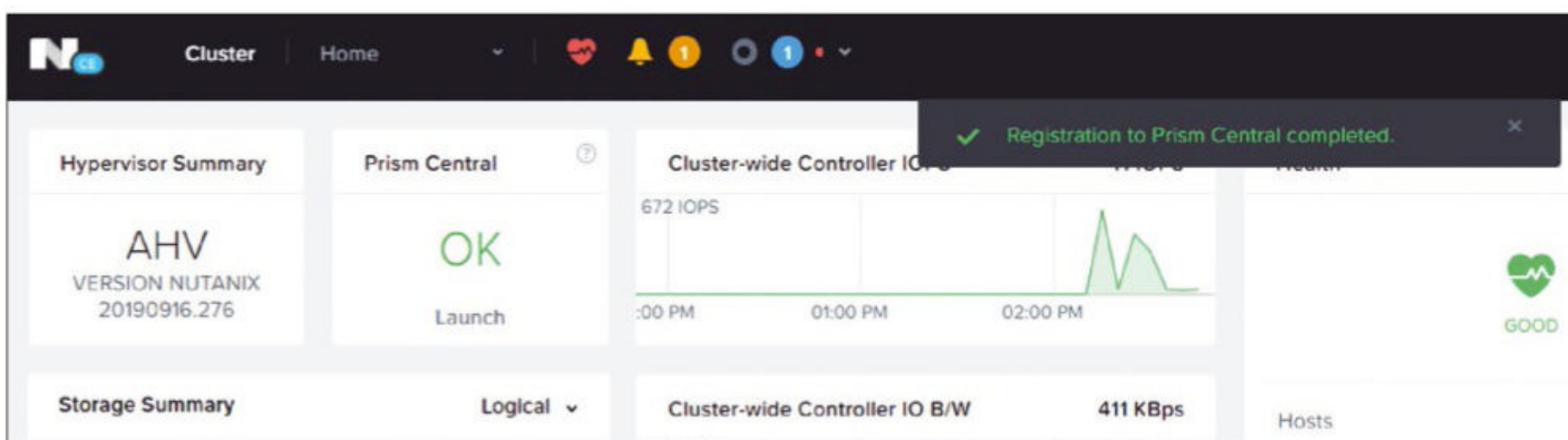


Figure 6: Cluster management is now possible in Prism Central.

the Nutanix REST API, or PowerShell, for which, of course, the corresponding Nutanix commandlets are also available. As you can see, you can get a huge amount of experience with the Nutanix Community Edition and gain insight into the manufacturer's solutions.

Startup and Shutdown

After the install, the cluster is running; however, you might want to shut it down and start it up again later. First, shut down all your workloads and Prism Central. If you are unable to initiate a shutdown from your workload itself, use the Prism Item UI under *Home | VM*, for example, to change the power state of a VM. Second, shut down the CVM by going to the CVM's command line and entering:

```
cvm_shutdown
```

The CVM shuts down after you are automatically logged off. Third, go to the AHV's CLI and enter:

```
poweroff
```

Wait until the NUC has been completely switched off by the system. To start up again, switch on the NUC and wait until you can access the Prism Element UI on [https:// <IP address of CVM>:9440](https://<IP address of CVM>:9440). Depending on the hardware, this may take a few minutes.

Conclusions

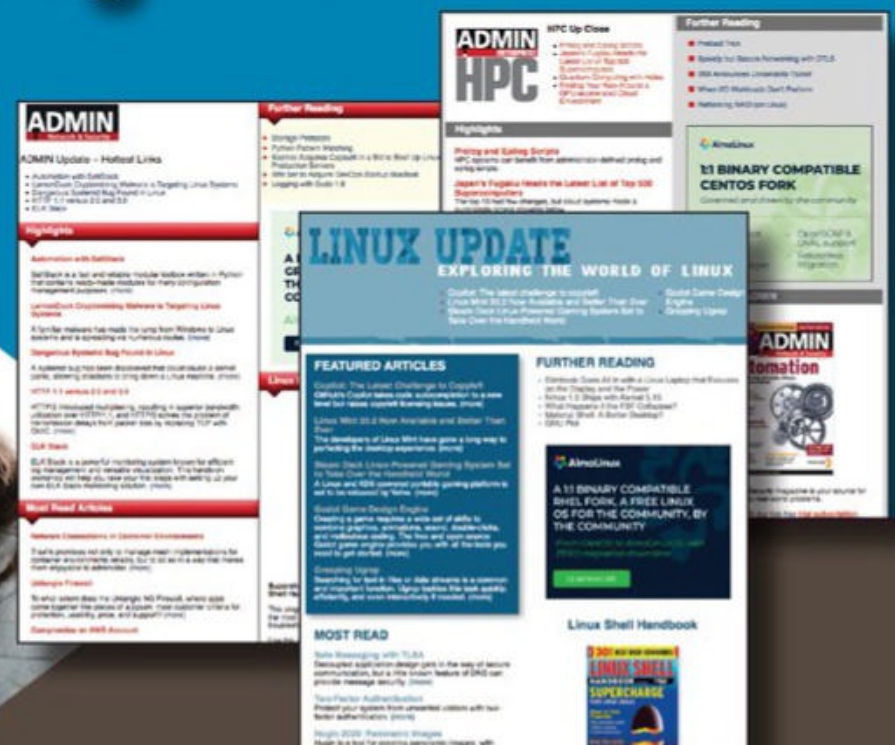
If you want to set up a cluster without too much overhead, you will find it fairly easy to do with Nutanix Community Edition. This free version is primarily intended for testing

the environment and exploring the platform. If you want to deploy the cluster in a production environment, you will need the full version, but both versions of Nutanix are easy to use and flexible. ■

Info

- [1] Creating a Nutanix account: [\[https://www.nutanix.com/products/register\]](https://www.nutanix.com/products/register)
- [2] Nutanix portal: [\[https://next.nutanix.com\]](https://next.nutanix.com)
- [3] Rufus: [\[https://rufus.ie/en/\]](https://rufus.ie/en/)
- [4] Overview of Prism Element: [\[https://www.youtube.com/watch?v=zQkSKix3qWs\]](https://www.youtube.com/watch?v=zQkSKix3qWs)
- [5] Commands and scripts: [\[https://portal.nutanix.com/page/documents/details?targetId=Command-Ref-AOS-v5_20\]](https://portal.nutanix.com/page/documents/details?targetId=Command-Ref-AOS-v5_20)
- [6] Information on Life Cycle Manager: [\[https://portal.nutanix.com/page/documents/details/?targetId=Life-Cycle-Manager-Guide-v2_4:Life-Cycle-Manager-Guide-v2_4\]](https://portal.nutanix.com/page/documents/details/?targetId=Life-Cycle-Manager-Guide-v2_4:Life-Cycle-Manager-Guide-v2_4)

IT Highlights at a Glance



Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your Inbox.

[Linux Update](#) • [ADMIN Update](#) • [ADMIN HPC](#)

Keep your finger on the pulse of the IT industry.

ADMIN and HPC: bit.ly/HPC-ADMIN-Update

Linux Update: bit.ly/Linux-Update

Workspace ONE for endpoint management

Empowered

VMware Workspace ONE provides a secure and user-friendly digital workplace. We look at the features, components, and architecture of Workspace ONE, as well as application management and simplification of the integration of end devices through user self-enrollment. By Jens-Henrik Söldner

The trend is moving away from stationary and toward mobile workplaces. That said, it is still important for the IT department to manage all end devices efficiently. Because of all the different operating systems, a clear-cut tool for endpoint management is essential. VMware's Workspace ONE [1] aims to integrate all of a company's devices, including devices belonging to employees

(bring-your-own-devices, BYODs), uniformly and centrally into the existing infrastructure with the help of a secure platform. The overriding goal is to organize application lifecycle management. The overall strategy is known as Unified Endpoint Management (UEM) and is one of the two core components of Workspace ONE. In addition to UEM, the second core component, Workspace ONE Access, combines the administration of users and user groups, the assignment of access authorizations to the applications, and the setup and provisioning of virtual desktops with the respective applications in a catalog (Figure 1). The Access component provides a connector to other identity (ID) providers such as Ping, Okta, and Microsoft Azure, providing a common catalog interface for all applications.

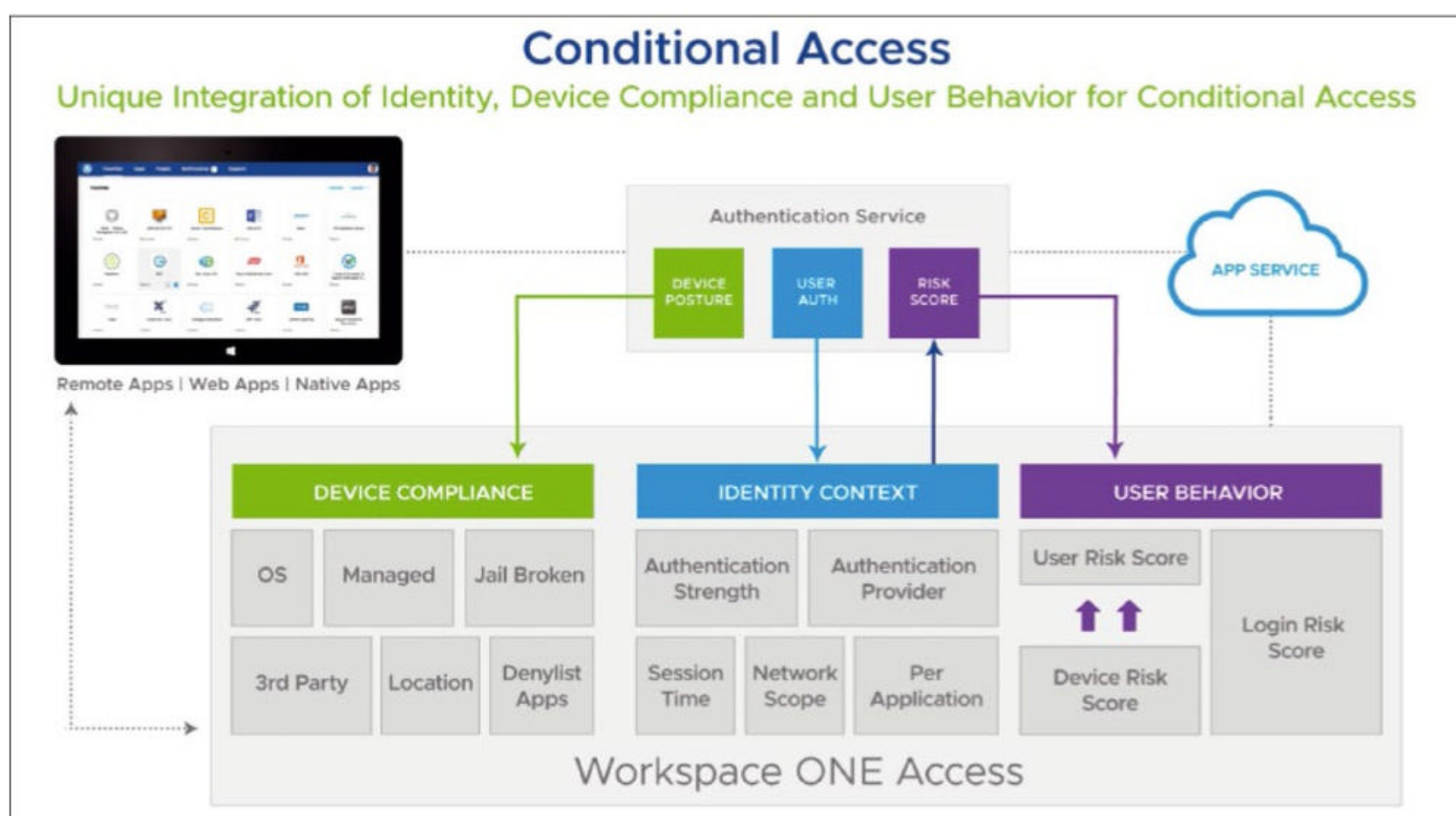


Figure 1: Managing user access with Workspace ONE Access involves various modules. Image courtesy VMware. [2]

Table 1: Workspace ONE Components

Component	Function
VMware Workspace ONE UEM	Enterprise mobility management
VMware Workspace ONE Access	Identity platform
VMware Workspace ONE Intelligence	App analysis and automation
Workspace ONE App	User access to apps
VMware Horizon	Virtual desktops and remote desktop services (RDS)
VMware Workspace ONE Boxer	Secure email client
VMware Workspace ONE Browser	Secure web browser
VMware Workspace ONE Content	Mobile repository for content
VMware Workspace ONE Tunnel	App VPN access to enterprise resources
VMware AirWatch Cloud Connector and VMware Identity Manager Connector	Synchronization with enterprise directories
VMware Unified Access Gateway	Gateway for secure Edge services
VMware Workspace ONE Secure Email Gateway	Email proxy server
Certificate Authority Integration	Lifecycle management of provided certificates
VMware Email Notification Service	Email messaging for Workspace ONE Boxer on iOS

The AirWatch Cloud Connector, in turn, securely transmits requests from Workspace ONE UEM to the back-end infrastructure. Table 1 provides an overview of all components of Workspace ONE.

Architecture and Services

Administrators use Workspace ONE to define user groups, policy settings, and device configurations. Users then

access their applications through Workspace ONE according to the defined settings and configurations. Figure 2 illustrates the interaction of the various components and interfaces in Workspace ONE.

Workspace ONE services are based on the integration of VMware Workspace ONE UEM, Workspace ONE Access, and VMware Horizon. Deployment can be achieved in a variety of configurations, including:

- On-premises deployment of Workspace ONE Access and Workspace ONE UEM.
- Cloud-based deployments of Workspace ONE Access and Workspace ONE UEM.
- Hybrid deployments with different components available either on-premises or in the cloud.

Basic EMM Features

As mentioned earlier, administrators can deploy, manage, and secure applications with Workspace ONE. In doing so, IT can leverage interfaces for diverse operating systems to configure mobile devices such as notebooks, smartphones, and tablets in line with corporate policies. Workspace ONE's UEM accesses enterprise mobile management (EMM) interfaces to provision, configure, and secure applications and devices. This level of control enables IT to implement a flexible BYOD program that lets users choose their devices.

In this context, EMM is an umbrella term for systems that manage mobile devices in the enterprise. It is subdivided into the following subareas:

- Mobile device management (MDM): Manages mobile devices

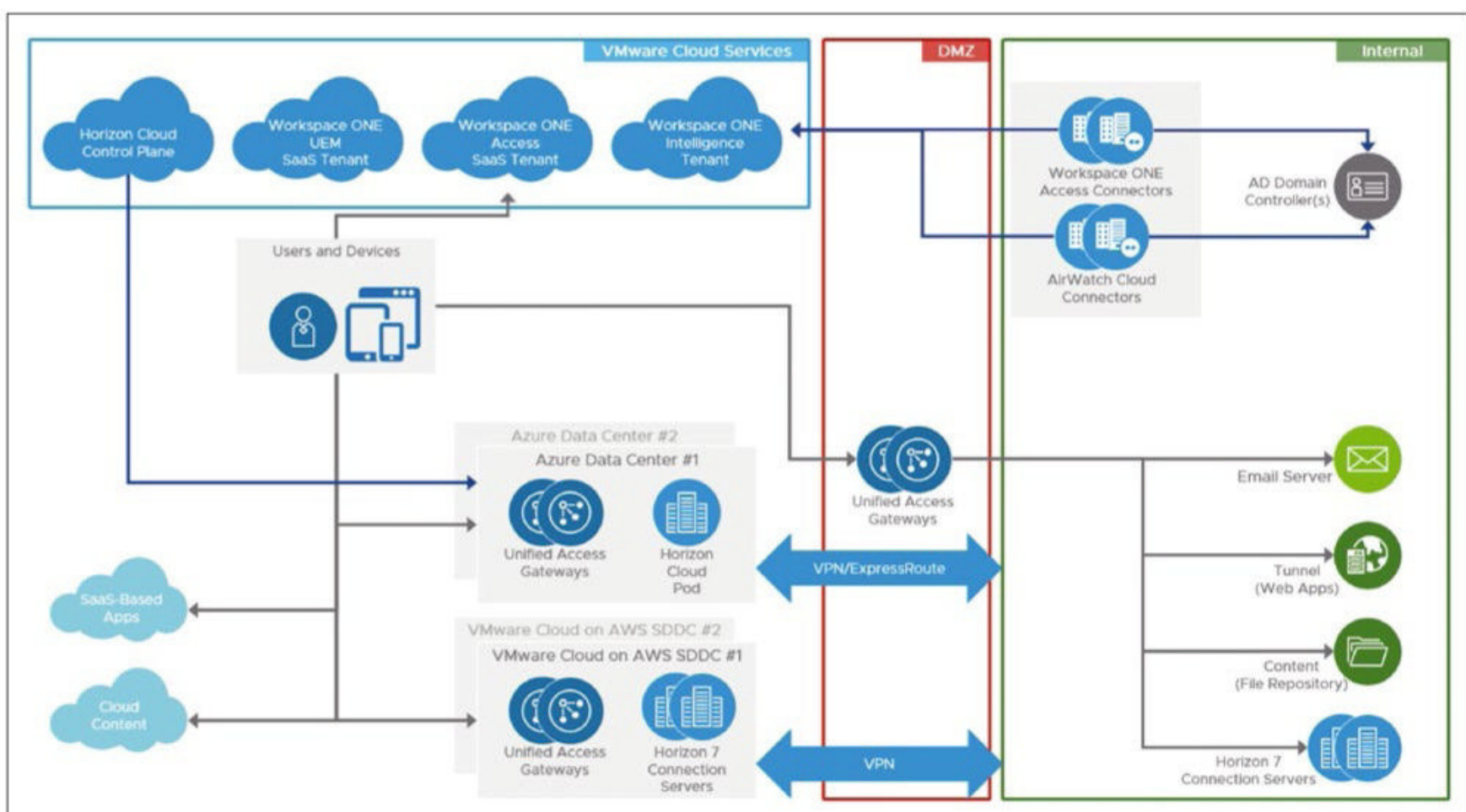


Figure 2: A schematic overview of Workspace ONE shows the interplay of the numerous components and interfaces. Image courtesy VMware. [3]

with software and hardware as specified by the enterprise.

- **Mobile application management (MAM):** Manages the software installed on mobile devices through an internal company app store.
- **Mobile content management (MCM):** Provisions the company's own applications and enables files and documents to be passed on to colleagues, business partners, and customers by the mobile route.

Profiles and Software Distribution

The use of device profiles changes the behavior of enrolled devices. Device profiles, combined with compliance policies, help enforce corporate rules and procedures. For example, Workspace ONE UEM device profiles can be created on the basis of criteria such as users, groups, platforms, and operating systems. You can also create smart groups, which are customizable groups that allow you to filter dynamically. User groups can control the assignment of access authorizations individually on the respective devices. Workspace ONE UEM lets you create device-specific profiles for the individual operating systems. The functions are adjustable to suit requirements, which means you can activate or deactivate one or more functions, such as switching off the Siri voice assistant on Apple devices. However, profiles need to be set up categorically and in a granular way. This approach simplifies the management of individual profiles. As a result, they are not valid for all devices but, instead, are specifically for the selected operating system in the matching smart group. Device

registration establishes the initial communication with Workspace ONE UEM to enable EMM.

Another component of UEM is data leakage protection. Examples of potential data leaks include saving work documents to a public storage device such as Dropbox or receiving work email with an unmanaged email client. Other security measures include encrypting and restricting email traffic, such as editing and sharing attached files. The company can also require that only applications provided by the enterprise are used instead of OS-native applications. The choice of browser can also be restricted for users to ensure secure browsing.

Workspace ONE lets you integrate existing ID providers, such as Microsoft's Active Directory or another LDAP-based directory, to implement user synchronization, authentication, and application access. Finally, the tool enables simplified software distribution. Deploying, updating, and deleting software packages can all be automated. Moreover, you can distribute the packages in predefined time windows and as a function of the network infrastructure load, and notifications for upcoming software updates lets users prepare for them accordingly.

Provisioning an Application Catalog

A wizard that acts as a checklist guides you through the individual settings in the UEM console, step by step. The wizard is divided into four modules: *Workspace ONE*, *Devices*, *Content*, and *Application*. Each module contains instructions on how to achieve specific goals. Because some steps overlap in some modules, the

wizard tracks progress across all four modules, ensuring that the same step never needs to be performed twice. Workspace ONE provides users with access to

cloud, mobile, Mac, and Windows applications through the unified Intelligent Hub application catalog, which includes applications for deployment in the main Workspace ONE Access and Workspace ONE UEM components. Therefore, when configuring Intelligent Hub, you need to connect Workspace ONE UEM with Workspace ONE Access. **Table 2** shows which type of deployment Workspace ONE provides for each operating system. In addition to the application types in **Table 2**, supported applications also include VMware ThinApp, VMware Horizon 7, VMware Horizon Cloud Service, and applications published by Citrix. Furthermore, the catalog supports virtualized desktops.

SSO and Two-Factor Authentication

Users install the Workspace ONE app on a mobile device and gain single sign-on (SSO) access to enterprise, cloud, and mobile apps with their corporate credentials. The Workspace ONE application leverages native operating system features to protect application access (e.g., biometric fingerprint readers on Android, Touch ID on iOS, and Windows Hello on Windows 10). Mobile SSO establishes trust between the user, device, application, and enterprise and enables one-touch mobile application logins. To protect more sensitive applications, you can enable biometric or other multifactor authentication methods. Mobile SSO is available for Android, iOS, and Windows 10 devices.

In combination with the VMware Verify mobile app, strong, multifactor authentication is also possible, simplifying access across devices. When a user attempts to access the Workspace ONE application store or any other application that requires strong authentication, Verify sends a notification to the user's cell phone, which provides additional security for applications that do not inherently offer multilevel authentication. For authentication, Workspace ONE provides multiple options to configure network-, platform-, and

Table 2: Deployment Types by Operating System

Platform/Application	Internal	Public	Web	Purchased
iOS	✓	✓	✓	✓
macOS	✓	-	✓	✓
Android	✓	✓	✓	-
Google Chromebook	-	-	✓	-
Windows Phone	✓	✓	-	-
Windows Desktop	✓	✓	✓	-

application-specific criteria. Once it has been configured and security rules have been created, compliance with security rules is mandatory. On the device, access to applications is not allowed until the security rules are accepted and proof of compliance is maintained. Compliance rules protect against rooted devices or devices with jailbreak. These rules can also be used to allow or prohibit applications.

Adaptive Management

With adaptive management, users do not log their devices into Workspace ONE UEM to access apps that require only a basic level of security. Instead, users download the Workspace ONE mobile app from the appropriate app store and log in with their credentials. From here, they can access their authorized applications. For apps that require a higher level of security, user access may only be possible after device registration.

According to the assigned device profile, the catalog displays all authorized mobile, software as a service (SaaS), virtual, and desktop applications. Applications that require re-registration are marked with a lock icon. When users try to download software with this icon, a registration process begins. For example, users could obtain a conferencing application such as WebEx without registering. However, downloading enterprise applications such as Salesforce requires registration.

Secure Integration with the Enterprise Network

Workspace ONE UEM leverages the company's existing network infrastructure to provide its own high availability, redundancy, and scalability for the applications and desktops that are ultimately delivered to end users. To this end, local load balancing is integrated on the back end of the SaaS environment. The backbone security infrastructure includes redundant Ethernet switches, LAN separation, firewalls, intrusion detection, and monitoring. Redundant firewalls are located

between the Internet and the Air-Watch environment. An intrusion detection system (IDS) monitors all internal network traffic, logs it, and sounds the alert when suspicious network activity is detected.

Other security features include:

- isolation of all Workspace ONE UEM web servers with a demilitarized zone (DMZ),
- antivirus clients to protect all servers, and
- spam filtering and spam reporting for email.

From a web-based HTML5 management console, you can control Workspace ONE UEM. All data transferred between the web console and mobile devices is encrypted. To ensure the environment meets the latest security standards, the cloud-based Workspace ONE components automatically update and patch themselves.

The approach to data center security is multilayered. Primary data centers have onsite backups for rapid recovery and replicated offsite backups for disaster recovery. Production systems are hosted in two primary data centers, with cross-site replication of nightly backups to support performance, growth, and security requirements.

Self-Enrollment of Private Devices

Finally, I look at self-enrollment as one of the many options for registering a device in Workspace ONE. For an iPhone, for example, you need to install the Intelligent Hub application from the App Store for the target device. To establish a connection from the target device to the UEM, the server address and the respective group ID must be entered. After entering and establishing the connection, the user logs in with credentials provided by the enterprise. The successful login is followed by a manual installation routine, which is used to complete mobile device management registration.

The previous steps are virtually the same for Android devices, the difference being that they can be used after a successful connection. That is, you can make the resources available to

the user, and the user can access the deployed applications within the Intelligent Hub catalog.

Unlike Android devices, however, Apple devices require an additional step before going live: an installation of one or more profiles. Once installed and connected to the Workspace ONE server, a window opens in the application asking you to create a profile. This step takes place outside of Workspace ONE, in the settings of the Apple device. After successfully creating a profile, the device is finally registered, and the user can access the application provided on the Intelligent Hub catalog. Likewise, you can view the registered device from the UEM platform and manage it as needed.

At all times you have the option to de-register the registered devices with what is known as an “enterprise wipe,” which deletes the changes made to the device by Workspace ONE. Another action is known as a “device wipe,” which resets the device back to the factory settings and deletes all data.

Conclusions

Thanks to Workspace ONE UEM, enterprises can easily integrate a fleet of mobile devices. Additionally, Workspace ONE Access enables the implementation of corporate policies and the unified deployment of applications. Other access components, such as the Airwatch Cloud Connector (ACC), support the integration of local and cloud-based LDAP directories. Workspace ONE can therefore be seen as a comprehensive tool for centralized and uniform management of end devices and mobile work. ■

Info

- [1] VMware Workspace ONE: [\[https://www.vmware.com/products/workspace-one.html\]](https://www.vmware.com/products/workspace-one.html)
- [2] Workspace ONE Access: [\[https://techzone.vmware.com/resource/business-continuity-vmware-solutions-remote-work#existing-workspace-one-uem-and-access\]](https://techzone.vmware.com/resource/business-continuity-vmware-solutions-remote-work#existing-workspace-one-uem-and-access)
- [3] Cloud-based logical architecture: [\[https://techzone.vmware.com/resource/vmware-workspace-one-and-horizon-reference-architecture-overview#cloud-based-logical-architecture\]](https://techzone.vmware.com/resource/vmware-workspace-one-and-horizon-reference-architecture-overview#cloud-based-logical-architecture)



Secure email communication

Trustworthy

DMARC combines the abilities of SPF and DKIM to safeguard and protect against spam and phishing and allows targeted configuration according to company policy. By Matthias Wübbeling

IT administrators, no matter their level of experience, agree that managing mail servers is one of the supreme disciplines. Too many fragile system settings, too many pitfalls, and above all, public pillories in the form of blacklists if something goes wrong during configuration. All the more respect goes to the admins who successfully manage mail servers, keep them permanently available, and ensure that outgoing email reaches the intended recipient reliably.

Microsoft in particular and the email service providers they control are considered particularly strict when it comes to accepting messages from third-party servers. Although some people claim that Microsoft is deliberately filtering third-party providers to boost the number of customers for its own services, this prejudice cannot be confirmed on closer inspection. Microsoft offers comprehensive

information about blocking mail servers and recommendations for action on a website set up specifically for this purpose [1]. If you have the mail server under strict control, you do not need to worry about problems with delivery to Live, Hotmail, or Outlook servers.

In this article, I look at how to secure email with the Domain-based Message Authentication, Reporting, and Conformance (DMARC) email authentication protocol.

Secure Mail Dispatch

Even though spam filtering and malware detection when receiving email play a major role in corporate security, in this article, I only look at how to secure email transmission. Of course, the measures I will be looking at also indirectly contribute to the security of enterprise email accounts, as long as the other

mail servers also use them. Because fake senders can be blocked even before email is received, the incoming spam volume is automatically reduced. The integration of corresponding tests for receiving mail servers is described in the documentation for the Exim and Postfix mail transfer agents.

Probably the most important step is to set up a Sender Policy Framework (SPF) record in the domain name system (DNS) that lets admins specify authorized outbound mail servers. Although designed in 2004, SPF only became the standard recommended by the Internet Engineering Task Force (IETF) in 2014 [2]. From a sample of more than 3,000 domains belonging to German companies, I examined the DNS records and determined the number of valid SPF records. About 25 percent of these domains do not have an SPF record stored, which

Photo by Simon Moog on Unsplash

means that recipients cannot check whether the delivering mail server is allowed to send messages for the sender domain.

This situation opens the door for phishing and CEO fraudsters; therefore, you should get an overview of your company's sending mail servers and enter them as SPF records in the DNS. If you can't easily determine this list, at least limit sending to your company's subnets and contracted mail sending service providers. Taking this step is still better than having no SPF entry at all. Note that the entry must be created for the domain that is used as the domain part of the email, not for the (incoming) mail server.

Signatures with DKIM

Domain Key Identified Mail (DKIM) [3] lets users sign outgoing email with a private key. The server's public key is stored in the domain's DNS. Here, too, the entry must be created for the domain part of the email. In contrast to SPF, however, multiple keys with different selectors can be managed, which results in DNS entries for different subdomains. For a selector with the name *admin-mag*, the appropriate DNS entry would be created in *admin-mag._domainkey.admin-magazine.com*.

The selectors are included with the email signature so that the receiving mail server can select the correct domain to receive the public key. Because the selector can vary from server to server, and the recommendation is even to change it regularly, it is not possible to make a comprehensive statement about the implementation of DKIM on the Internet within the scope of this article. As a lower boundary, however, it can at least be stated that around 13 percent of the domains tested have entered one of the standard selectors from the documentation and tutorials. However, it is not possible to say reliably whether this actually signs outbound messages, if the incoming email message was not signed.

Guidelines with DMARC

SPF entries come with instructions on how to deal with senders not mentioned in the entry, but email without a DKIM signature should not be rejected without further ado. Because of the different selectors, no uniform DNS entry exists to check the existence of a DKIM key.

DMARC lets you define appropriate policies for domains, which means you can specify that email from a domain always needs to be signed with DKIM, even independently of specific DKIM selectors. With the associated policy, you can then determine whether non-signed email will be ignored by the recipient, quarantined, or directly rejected.

Additionally, you can specify the relative proportion of your email messages for which the DKIM signature will be verified. To be on the safe side, you should, of course, have 100 percent verified. In the implementation phase, but also permanently for information purposes, you can store URIs to which forensic and summary reports of the DMARC check are sent. You also need to ensure that you can receive and process the reports sent once or several times a day by the various mail servers.

To receive the reports by email, enter a *mailto* URI with your email address. However, make sure it matches the domain being checked; otherwise, other mail servers will initially refuse to send the statistics for security reasons. This behavior can be further customized with additional DNS entries and use other domains for receiving DMARC reports.

Silent Mail with SRS

Another aspect of mail server operation – especially if you maintain mailing lists or allow forwarding to external email addresses – is the Sender Rewriting Scheme (SRS). Suppose you set up forwarding of an email address, as in *distributionlist@admin-magazine.com*, to external recipients. Your incoming

mail server then becomes a relay for external senders and sends email on their behalf to all recipients specified in the forwarding list. This is also how mailing lists, such as those belonging to the popular Mailman software [4], work by default.

The mail servers of these recipients should reject the messages from your server because your mail server is probably not registered in the SPF record of the original sender. What can help in this case is to set up an SRS service on the mail server that rewrites the sender addresses of forwarded email to a temporarily valid email address on your server that has a filter set up. With this new sender address, the email then passes the SPF check – your server is posted in the SPF entry for the temporary address – and the email can be forwarded successfully.

Conclusions

For the secure use of email as a medium, servers also need to be configured comprehensively to check outgoing messages. The configuration itself is not at all complicated and can be implemented for most mail servers in just a few hours. As an administrator, you can prevent third-party mail servers from sending email on behalf of your own domain and causing damage. If you consistently use the techniques presented here, you can secure your mail server against this scenario, which means you also can reliably send messages from your own mail server to particularly restrictive mail servers of large enterprises, such as Microsoft or Google. ■

Info

- [1] Outlook page on outbound mail problems: [\[https://sendersupport.olc.protection.outlook.com/pm/troubleshooting.aspx\]](https://sendersupport.olc.protection.outlook.com/pm/troubleshooting.aspx)
- [2] RFC 7208 on SPF: [\[https://www.rfc-editor.org/rfc/rfc7208.txt\]](https://www.rfc-editor.org/rfc/rfc7208.txt)
- [3] RFC 4870 on DKIM: [\[https://www.rfc-editor.org/rfc/rfc4870.txt\]](https://www.rfc-editor.org/rfc/rfc4870.txt)
- [4] Mailman: [\[https://list.org\]](https://list.org)



Hardware-enhanced security

Key to Security

Nitrokey hardware authentication devices aim to raise data encryption, key management, and user authentication security to the next level. By Rubén Llorente

Consumer-grade security implemented by software is cost effective for most users, who may install and run password managers, encryption tools, and other privacy programs for virtually no cost; however, software comes with limitations.

Most users can live with these limitations, but if you are exceptionally conscious of your privacy or handle top secret data, you do not want to take any chances. Thankfully, you

have alternatives to running your security applications in software only: Nitrokey.

Enter the Key

Nitrokey GmbH [1] is a German hardware company focused on security products. Its main product line is a variety of Nitrokeys, which are hardware authentication tokens in the form of pen-drive-sized units

that connect to your computer or server over the universal serial bus (USB). The manufacturer offers a whole range of models with different capabilities, with prices ranging from EUR29 to 109 (VAT not included; or about \$25–\$88). They also produce a line of Qubes OS-certified laptops and Nextcloud appliances and provide a free Matrix instance for those who want a secure chat platform.

The Nitrokey website claims that their hardware can be used to enhance the security of certain web logins, email encryption, hard disk encryption, and SSH access. Whether this is true, and whether a Nitrokey is worth the cost, is what I intend to determine in this article.

I am reviewing Nitrokey Storage 2, the most featureful Nitrokey available. For EUR109 plus taxes and shipping, you get a Nitrokey with 16GB of encrypted, tamper-resistant storage, a password manager, and a sticker (Figure 1). What it lacks is a manual, which is disappointing. For the price, they should have included at least a quickstart guide. Instead, a label in the packaging instructs you to check the online documentation [2] for instructions.

The Nitrokey looks solid enough, but I have seen Kingston pen drives that



Figure 1: A Nitrokey Storage 2 comes with a keychain hole and a sticker but no instruction manual.



Figure 2: An inexpensive OTG adapter would let you use a Nitrokey with an Android phone.

looked hardier. The cap that covers the USB plug might be easy to lose because you don't have a way to attach it to the body of the Nitrokey while the device is in use.

Getting Started

Your computer will need a software application to interface with the Nitrokey. Thankfully, you are provided a free, open source program [3] that supports FreeBSD, macOS, Windows, and many Linux distributions. AppImages, Flatpaks, and Snaps are also available. Nitrokeys appear to be partially supported on Android smartphones over USB On-the-Go (OTG) [4], although it would require an adapter (Figure 2). In any case, Android is not listed as an officially supported platform. For testing, I decided to use Knoppix 9.1. The software necessary to

initialize a new Nitrokey can be installed by issuing the commands:

```
sudo apt update
sudo apt install nitrokey-app
```

The Nitrokey App ends up under the *Accessories* category of the Start menu, from which the program may be launched and the Nitrokey plugged in for initialization. The documentation instructs you to change the default

user and admin PINs and then load either OpenPGP or S/MIME keys into the Nitrokey (see the "What is OpenPGP?" box). The steps for changing the PINs are very intuitive (Figure 3), but key initialization is not.

I decided to generate an OpenPGP set of keys for testing, but the steps suggested by the documentation failed to function. As an alternative, I issued the commands

```
$ gpg --edit-card
gpg/card> admin
gpg/card> generate
```

to start an interactive menu (Figure 4) from which I could generate a set of keys. The strength of this method is that the keys are generated by an internal Nitrokey chip and are never accessible to the operating system. Therefore, the private keys cannot be

retrieved by any means, even if the operating system is compromised. The Nitrokey boasts tamper-resistant storage, which in theory makes it impossible to access the private keys by prying the Nitrokey open. Although not emphasized in the documentation, it is very important that the public keys are exported right away, using

```
gpg --armor --export <yourID> > key.asc
```

because they must be imported manually in every other computer on which you intend to use the Nitrokey. The pen drive is not capable of generating them on its own. If the computer on which you generated the keys is lost (which actually happened during my tests) and you have no copy of the public keys, the keys in the Nitrokey will be, for all intents and purposes, unusable, and you will need to generate new ones.

GNU Privacy Guard (GPG) or software that leverages GPG for encryption (e.g., Pass or Mutt) can then use the Nitrokey for decrypting sensitive data with an additional layer of protection. Because the private keys can never leave the Nitrokey, GPG sends any material that needs to be processed by the private key (e.g., an encrypted message) into the Nitrokey

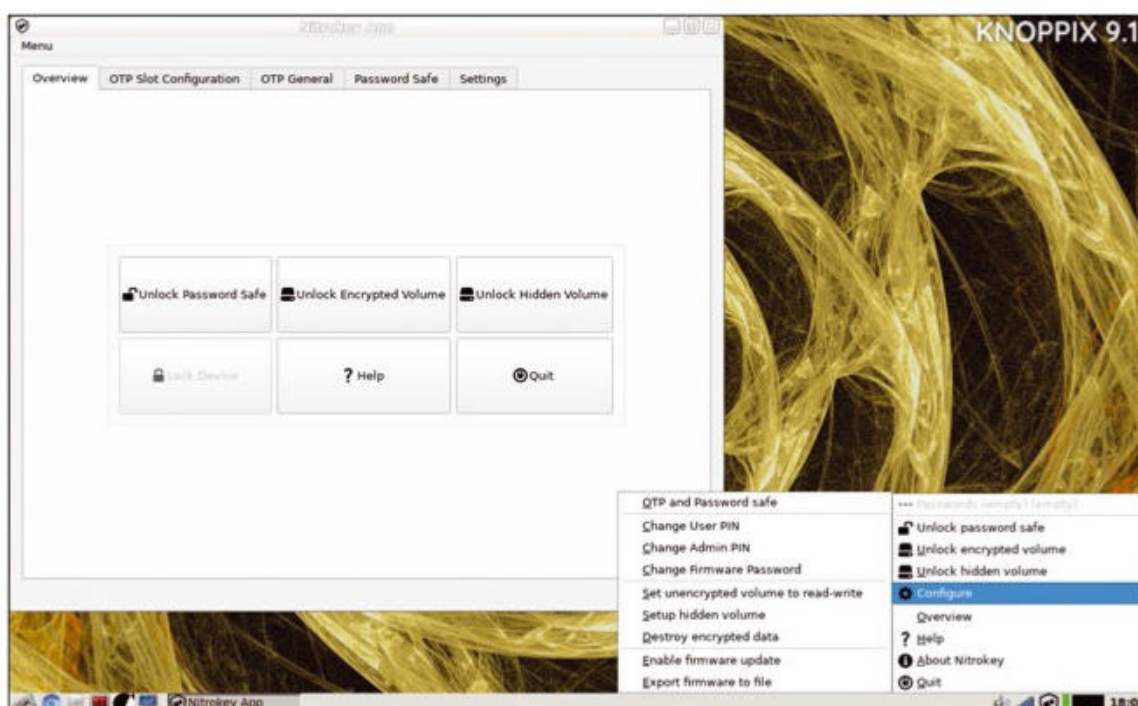


Figure 3: The default PINs are easily guessable and must be changed from the systray applet.

What is OpenPGP?

OpenPGP [5] is a standard for encrypting messages, particularly email. The theory behind it is complex, but in essence, each user creates a private key and a public key, which are big chunks of cryptographic material. Briefly, encryption uses the public key and signing uses the private key, whereas decryption uses the private key, and signature verification uses the public key. When you want to send an encrypted email, you place your private key in a safe place (e.g., a Nitrokey) and give a copy of the public key to each person from whom you want to receive encrypted email.

Despite being regarded primarily as an email encryption protocol – it is commonly integrated with email clients, such as Thunderbird or Claws Mail, to perform encryption and decryption automatically – OpenPGP has many other applications. For example, the Pass password manager uses OpenPGP to store passwords in encrypted form.

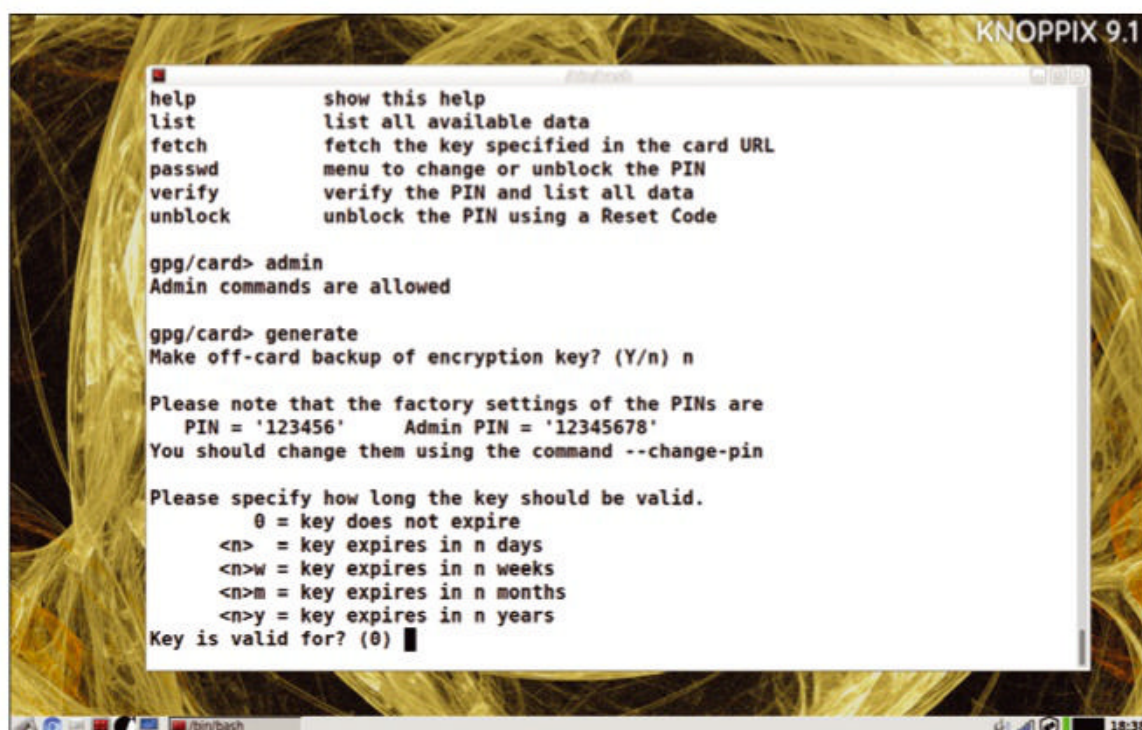


Figure 4: The steps described by the official documentation didn't work, but `gpg --card-edit` made it possible to generate a set of OpenPGP keys on the Nitrokey.

itself, which returns the processed data back to GPG (in my example, a stream of unencrypted data).

Therefore, for performing tasks like decrypting OpenPGP messages addressed to a given user, both the Nitrokey and the PIN used to unlock it must be under the control of the person performing the decryption. If the Nitrokey is not plugged in, it is impossible to decrypt any message addressed to the user. Should the Nitrokey be stolen, the keys within it will be useless to the thief because they can only be used by the person who knows the PIN. The Nitrokey provides a limited number of attempts for guessing the PIN: Once the limit is reached, the keys become unavailable.

The advantages of using a Nitrokey for storing OpenPGP keys are, thus, substantial.

Protected Computer Login

Nitrokeys may be used as access tokens for privileged access management (PAM)-enabled distributions. You may, for example, configure your operating system to allow only you to login if you plug in a Nitrokey that contains your set of keys. Once this is done, only a person controlling both the Nitrokey and its PIN may login, invoke `sudo`, or unlock the screensaver.

This method is a big upgrade from regular password protection because obtaining the password is no longer enough for the attacker.

The steps for using the Nitrokey in such a way are documented [6], but setting up the configuration involves messing with PAM configuration files and carries the possibility of breaking your operating system and rendering it unusable. Additionally, the documentation is not very helpful.

I used Ubuntu 21.04 to test the Nitrokey as a login token. The first step was to install `poldi`, which is a PAM component designed to work with OpenPGP tokens:

```
sudo apt-get install libpam-poldi
```

Next, you need to obtain the application ID of the Nitrokey. After plugging it in, issue the command

```
gpg --card-status | Application
```

to see the identifier (e.g., D0060001240102000000000000xxxxxxx).

Now, `poldi` must be informed that it is supposed to manage the authentication of the system user. For example, if your login name is `linux`, you would place the following line in `/etc/poldi/localedb/users`:

```
$application_identifier linux
```

and replace `$application_identifier` with the string you got in the previous step. This operation requires root privileges.

The last part is dangerous; mistakes may render your operating system unusable, so ensure you have a rescue CD around just in case you break something. You must modify PAM to accept the Nitrokey. The easiest way to do this in Ubuntu 21.04 is to open the file `/etc/pam.d/common-auth` and replace the line

```
auth [success=2 default=ignore] \n    pam_unix.so nulloc.secure
```

with the line:

```
auth [success=2 default=ignore] \n    pam_poldi.so
```

From now on, you will need to plug the Nitrokey into a USB port and enter the PIN for login into a graphical session, unlocking the screensaver or even invoking `sudo`. The bad news is that integration seems a bit rough around the edges. For example, the appearance of the Gnome display manager (GDM) is slightly mangled when asking for the Nitrokey PIN instead of a user-password combination.

One must wonder whether going through this hassle is worth the trouble. It certainly works as advertised, but the only threat this setup protects against is hardware keyloggers intended to steal your login password, because the person who planted the keylogger will need your Nitrokey to log in, even if your PIN is discovered. It is a neat concept but not extremely useful. Somebody capable of planting a keylogger to steal your password probably can boot the computer with a Live CD while nobody is watching to perform all sorts of horrible deeds without your credentials.

Password Management

The Nitrokey Storage 2 comes with an integrated password manager. You may use your Nitrokey alongside the Nitrokey App to store your passwords

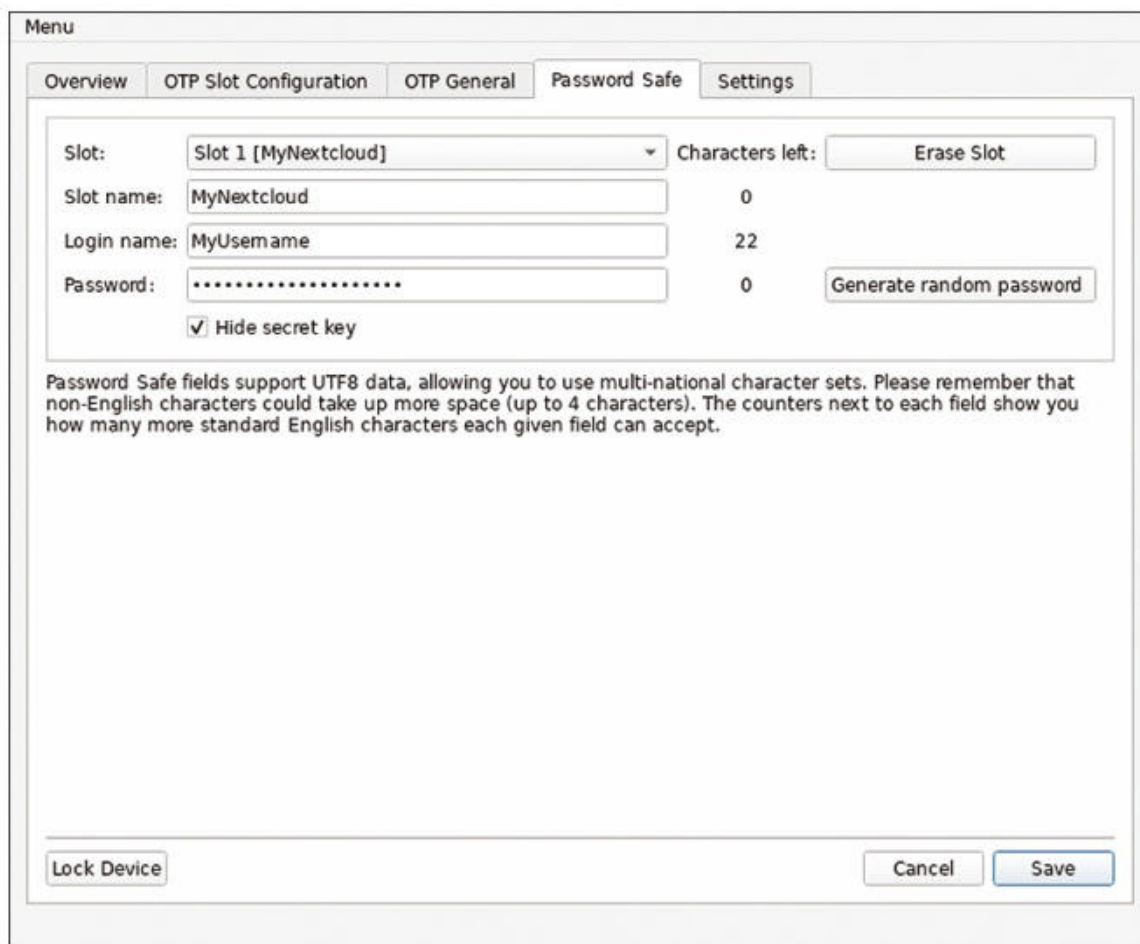


Figure 5: The User PIN may be used to retrieve a password stored within the Nitrokey. The drawback of this approach is that, once the Nitrokey gives the password over to the operating system, it could be stolen if the operating system is compromised.

within the Nitrokey secure storage. The passwords stored in such a way will only be accessible by using the correct PIN (Figure 5).

The main limitation of the integrated password manager is that it has room for only 16 passwords. My personal collection of passwords has more than 150 entries in it, so the Nitrokey falls short. Moreover, the integrated password manager does not offer a high level of protection because it does not conceal the passwords themselves from the operating system as well as it does with private OpenPGP keys.

The Pass password manager can take advantage of the Nitrokey and be configured to use the keys stored within to manage stored credentials. The Nitrokey must be plugged in and the PIN entered to access the passwords managed by Pass. The documentation also points out that KeePass can be configured in a similar way [7].

One-Time Passwords

Although the Nitrokey is not much better than a software password

manager, it hides an ace up its sleeve: one-time passwords (OTPs), which are useful for services that support two-factor authentication (2FA). Websites that support OTP-based 2FA work by generating an OTP secret key, which you must store in a secure device (e.g., the Nitrokey); then, each time you attempt to access the website, it asks for both your regular password and a one-time password (Figure 6).

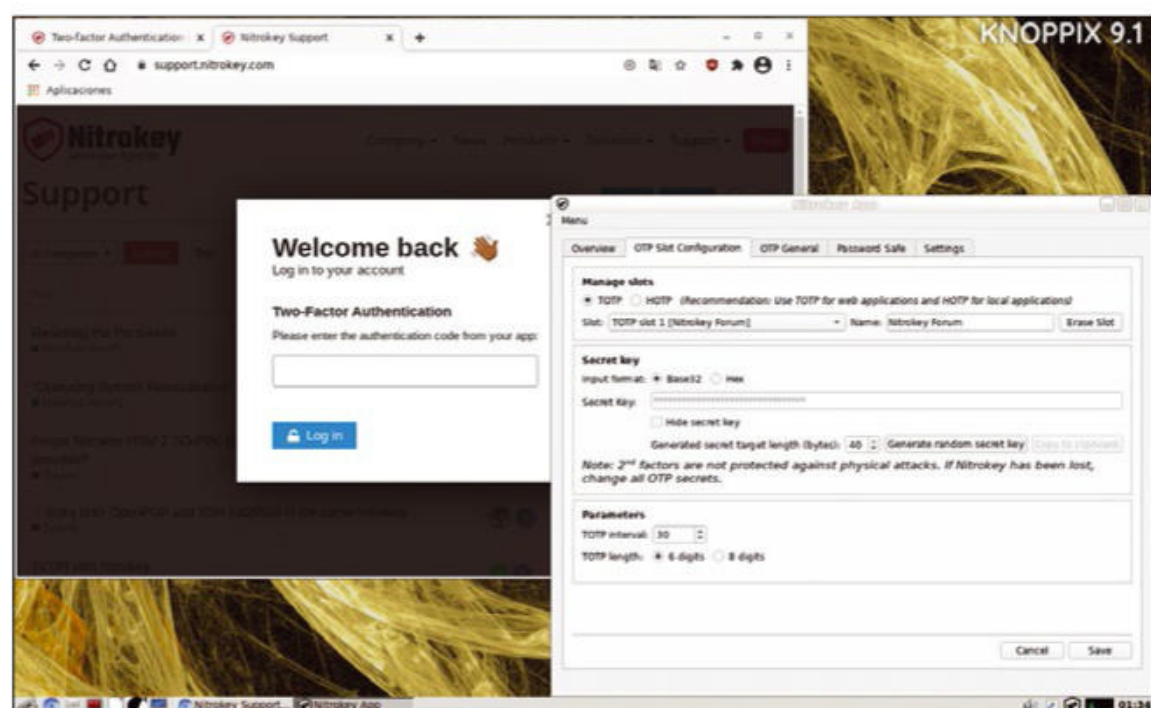


Figure 6: The Nitrokey adds security to certain web accounts by enabling 2FA. Once enabled, 2FA requires the appropriate Nitrokey and the password for logging in.

One-time passwords are strings generated as a function of the OTP secret and a predictable factor (e.g., time). The user is expected to use the Nitrokey to generate a one-time password, which will only be valid for a short amount of time (typically less than a minute), that the website will recognize.

One-time passwords are very strong because sniffing them is useless. They are ephemeral, and a given OTP will never be used again. To break 2FA, an attacker would need both your regular password and your OTP secret. Because the OTP secret resides only within the Nitrokey, the attacker would need to get the Nitrokey itself and the PIN to access the OTP secret and generate one-time passwords that a website will recognize as yours.

Breaking 2FA is not trivial. Then again, the Nitrokey represents a massive step up from regular password authentication. However, a multitude of free software implementations exist for OTP. Most users of OTP-based 2FA just install an application such as mobile OTP (mOTP) in their smartphones for the same effect. Although it might be argued that a smartphone is less secure than the Nitrokey, because a smartphone can suffer malware attacks more easily than a Nitrokey, I am not convinced the security gain is high enough to justify the

Why Not 2FA?

Two-factor authentication is great when it works, but my experience is that it is awful when it doesn't.

I have seen some forums, Cryptocoin exchanges, and businesses implement 2FA because of the security gain. Somebody who has 2FA enabled won't have their account compromised unless both the password and the OTP device (Nitrokey, smartphone, or similar token) are compromised. However, small sites short of personnel tend to crumble under an avalanche of support requests from users soon after 2FA is implemented.

The reason is that if the device that contains the OTP secret is lost, the user can no longer access the service. Most users don't make backups of their OTP secret, which means most users will eventually lose it and send a support request to the website operator.

On serious websites there is no such a thing as an automated 2FA reset. The banks I

work with will have the user walk into an office to prove their identity before access is restored. After all, if something is valuable enough to protect with 2FA, it is probably too valuable to consider sending a reset link to an email address. Email accounts are usually protected by a traditional password: If your bank was willing to send you a credentials reset link by email, an attacker would only need to break into your email account (which is protected by a single factor of authentication) and request a reset to the bank. This would downgrade 2FA to "1FA" and render the whole system pointless.

Therefore, users who want to protect their digital life ought to make a backup of their 2FA device, and administrators of sites that want to deploy this technology need to be aware of the costs. The resources to be spent in a user support department will probably increase more than expected.

expense. (See the "Why Not 2FA?" box.)

Encrypted File Storage

The Nitrokey Storage 2 works as an encrypted pen drive. Files stored within the Nitrokey Storage 2 unit won't be accessible unless the correct PIN is entered – and the number of tries is limited to protect your data against theft. The Nitrokey is much more safe than a regular pen drive for storing sensitive files.

Most home users who want to encrypt the content of their USB drives opt for software solutions such as VeraCrypt or dm-crypt, which are free of cost, quite convenient, and very safe. The drawback to these tools is that they are vulnerable to keyloggers and malware: If you enter a VeraCrypt password in a computer that has a keylogger reporting your activity to an evil entity, your security will be broken. The Nitrokey fails to counter this vulnerability because a PIN can be sniffed as easily as a regular password. Although the feature worked as intended during my tests, I think it does not add much security in this regard.

On the other hand, hidden volumes are supported (see the systray applet

menu in [Figure 3](#)). Hidden volumes are concealed filesystems that provide plausible deniability, whose existence cannot be proven, and are hidden within the regular encrypted storage. (See the "Is Plausible Deniability Safe?" box.) The intention is for the user to have innocuous files in the regular encrypted storage and the top secret data kept within a hidden volume. Were the Nitrokey to be

Is Plausible Deniability Safe?

The main problem with plausible deniability is that, although it is impossible to prove the existence of hidden information within the device, it is impossible (in practice) to prove that a hidden volume does not exist.

Imagine a terrorist organization steals your Nitrokey and kidnaps your hamster. When they phone you and tell you they are going to torture your hamster until you reveal the password of the hidden volume, if the Nitrokey does not have a hidden volume, the terrorists will torture your hamster forever because they have no way of knowing whether it has concealed data or you are lying.

Even if you reveal the password of a hidden volume, the terrorists will threaten to keep torturing your hamster unless you reveal the password of the hidden volume stored within the hidden volume. (Yes, some cryptosystems can stack multiple hidden volumes.)

stolen and its owner captured and tortured for the password, the user could provide the PIN for the regular encrypted storage and convince the captors that its contents are the only thing stored in the encrypted vault. The hidden data would remain concealed and thus safe.

Live DVDs and Sensitive Secrets

Modern operating systems are quite messy when it comes to handling secret files. When you open a file with a program, there is always the chance that the program will put pieces of the file in caches, temporary folders, or even the swap partition. When working with a secret file, this is problematic.

If you mount an encrypted volume in a directory and open one of its files (e.g., with LibreOffice) pieces of the file might end up in unencrypted places of the hard drive. At the very least, the path of the file would be added to the Recent Documents list, which is easily retrievable and therefore vulnerable.

Live DVDs are thus a great tool for working with sensitive files: You can load the Live operating system and mount the encrypted volume

Game theory, then, suggests that plausible deniability might be a liability (assuming you care for your hamster) because your enemies have no reason to stop pressing on. The possibility always exists that more information is hidden that you don't want to reveal, so the incentive is to keep torturing your hamster. Meanwhile, your best alternative is not to reveal the passphrase because you know they are not going to stop the torture, no matter what you give them.

One idea is that plausible deniability systems of this sort are extremely valuable to the owner of the data as long as that owner is not the person who has the passphrases. For example, if you are an executive and you hand a Nitrokey to your assistant to store some top secret files on it, you may rest assured she won't reveal the password of the Nitrokey, even if they threaten to torture her hamster. She would know that talking would not save her pet.

within it. Once you are finished working with the encrypted files, turning the computer off erases any trace of activity from the machine. Should the computer be stolen, the thief would not be able to retrieve the information.

Conclusions

Although encrypted storage worked well in the tests, it is arguably the most expensive feature the Nitrokey has to offer. The price difference between the Nitrokey Storage 2 and the Nitrokey Pro 2 is EUR60 (VAT excluded), and the only meaningful feature the latter lacks in comparison is encrypted storage. Software implementations that offer similar features, including hidden volumes, cost nothing.

The main advantage the Nitrokey Storage 2 has over software implementations is that the number of times an attacker can try a password is limited by the hardware chip, so in theory, the Nitrokey is much safer because it cannot be brute forced. However, the equivalent software implementations are considered unbreakable in practice, as long as good passphrases are used. The increase of security brought by the Nitrokey is significant, but whether a home user can justify the expense is a different question.

The Nitrokey Storage 2 works as advertised for the most part. Keeping

a set of OpenPGP keys within a Nitrokey is just safer than storing them on your hard drive, as is the usual practice. The hardware-enhanced encrypted storage is a good upgrade from common software encryption tools, as long as the files to be protected are important enough to justify the expense.

Its password management capabilities, alongside its ability to function with 2FA, are quite handy, but they don't add much security when compared with software solutions.

The Nitrokey is a portable solution that might help you move files between a heterogeneous group of computers. A cool feature of the Nitrokey Storage 2 is that it includes a nonencrypted partition that can be set as read-only. Out of the factory, it comes with a version of the Nitrokey App for Windows, Linux, and macOS, which is convenient for using the Nitrokey on computers without an Internet connection or that don't make it easy to install third-party software.

Finally, the Nitrokey is partially supported under Android. The only feature that works on such a platform through the OpenKeyChain application is the smartcard functionality for managing OpenPGP keys, which means no encrypted data storage or password management on Android. However, email signing and encryption-decryption are available.

The list of functions Nitrokeys [8] can address is amazing. In addition

to the features described in this article, the Nitrokey Storage 2 can be used as an SSH authentication token, perform certificate-based authentication with websites, or authenticate into virtual private networks (VPNs). The documentation is barely sufficient, though, and although hobbyists might benefit from using the Nitrokey, leveraging the full power of this device is only within the reach of power users and professionals. ■

Info

- [1] Nitrokey: <https://www.nitrokey.com/>
 - [2] Docs: <https://www.nitrokey.com/start>
 - [3] Platform support: <https://www.nitrokey.com/download>
 - [4] Nitrokey on Android: <https://www.nitrokey.com/news/2017/using-nitrokey-android-phones>
 - [5] OpenPGP: <https://www.openpgp.org/>
 - [6] Nitrokeys for computer login: <https://www.nitrokey.com/documentation/applications#computer-login>
 - [7] Pairing with KeePass: <https://www.nitrokey.com/documentation/applications#password-manager>
 - [8] Documented applications: <https://www.nitrokey.com/documentation/applications>
-

The Author

Rubén Llorente is a mechanical engineer whose job is to ensure that the security measures of the IT infrastructure of a small clinic are both legally compliant and safe. He is also an OpenBSD enthusiast and a weapons collector.



Detecting and analyzing man-in-the-middle attacks

Cuckoo's Egg

Wireshark and a combination of tools comprehensively analyze your security architecture. By Thomas Joos

In man-in-the-middle (MITM) attacks, attackers place themselves between the victim and the targeted resources, putting them in a position to intercept, read, and possibly even manipulate communications. In doing so, the attacker does not have to redirect the traffic completely or impersonate the data target. Instead, they can sniff the data on the network and then let it continue to the intended target without interference. In other words, the attacker is in the middle of the data flow. As a result, many users and administrators do not identify these attacks until it is too late, because in most cases, network services are not disrupted by the attack. Services continue to run normally while the attacker accesses the traffic between the client

and the server. Identity theft, faked transactions, or stealing intellectual property are just a few possible results. These attacks can just as easily be performed on cable-based networks as on WiFi, although they are particularly common on WiFi networks because public WiFi is often virtually unprotected. Before I look at possible defense mechanisms and tools such as Wireshark, I'll first look into how an MITM attack takes place, with techniques such as Address Resolution Protocol (ARP) poisoning, and how you can detect and analyze attacks, which in turn can help you protect your own network against MITM attacks and optimize your internal security structure accordingly.

ARP Gateway

MITM attacks often rely on the ARP cache, which is the local cache with IP to MAC address assignments. Its content can be displayed at the Windows command line by typing

```
arp -a
```

(Figure 1). On Linux computers

```
ip n s
```

does the same thing. This information can help detect MITM attacks because the command shows whether a MAC address is stored on a computer for two or more different IP addresses, which can be indicative of ARP spoofing. However, any attacker can also read and manipulate the data because the cache has no authentication and protection. Anyone can view and change IP addresses and the associated MAC addresses and use them for attacks. The example of ARP spoofing used here plays out as follows: PC1 belongs to the victim. It receives ARP responses from the attacker's PC, which pretends to be a router with a route to the Internet. As a result, PC1 redirects its ARP requests to the attacker's PC, which forwards them to the real router and vice versa. At the same time, the router receives ARP responses from the attacker impersonating PC1, which results in all ARP information intended

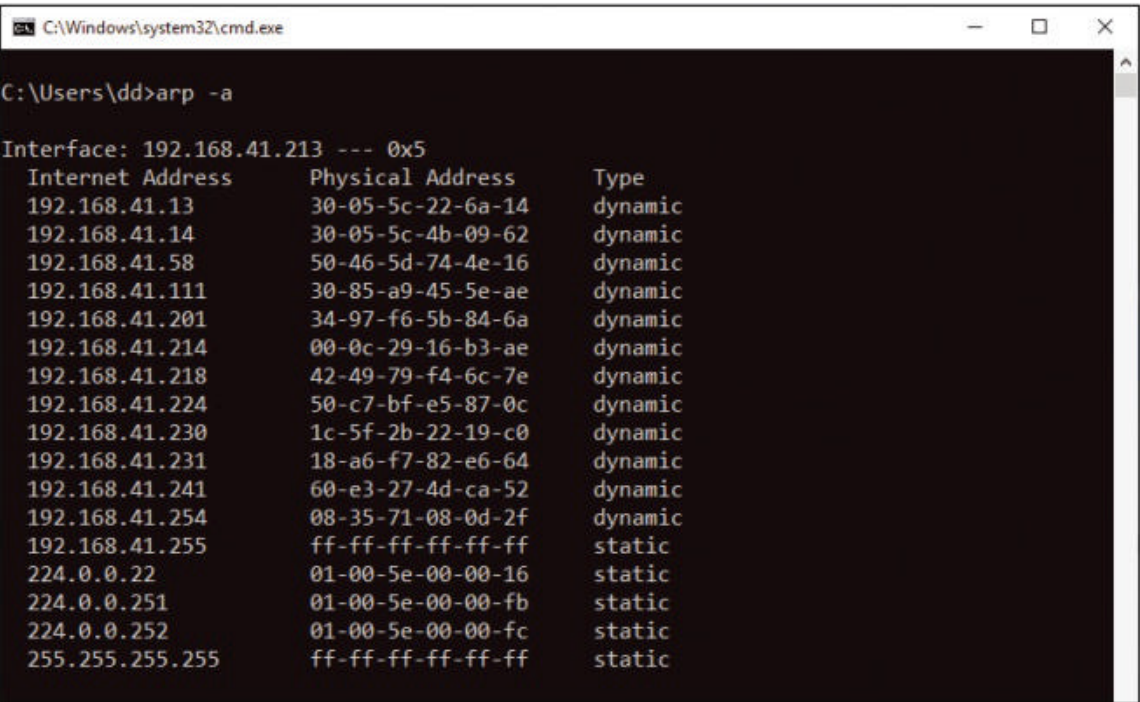


Figure 1: The ARP cache can be displayed from the Windows command line.

Photo by Soner Eker on Unsplash

for PC1 reaching the attacker's PC, which redirects the responses to PC1. The attacker can now view and modify all the packets received. If the data traffic is not encrypted, intruders can grab login data for HTTP websites or the content of documents with this approach. Additionally DNS spoofing, phishing, keylogging, and many other attacks are possible in this way. The best way of preventing ARP spoofing and thus MITM attacks is end-to-end encryption. In this case, all the data traffic between the devices involved is encrypted, and attackers cannot use the intercepted data. End-to-end encryption is made possible by the use of protocols such as HTTPS, POP3S, or IMAP4S. Managed switches also offer the option of preventing such attacks, although the security function first needs to be enabled on the switch. On Cisco switches, for example, this function is known as Dynamic ARP Inspection (DAI).

Setting up Wireshark

Wireshark [1] sits on the network like an MITM attacker and captures data traffic, allowing you to detect patterns that could indicate an MITM attack.

However, Wireshark is also frequently used by attackers because it analyzes network packets unobtrusively. If an intruder uses ARP spoofing to route packets to their own computer, Wireshark can analyze the packets in the same way – and you can detect these packets on the network. In other words, Wireshark can help you carry out, prevent, or log MITM attacks. On Linux, the *libpcap* library

is a prerequisite for using Wireshark. An installation on Windows 10 or 11 is possible, as well. As part of the installation, Wireshark can import the latest version of Npcap. On Windows, Npcap or WinPcap has to be in place to capture live network traffic. Wireshark includes Npcap as of version 3.x; the older versions use WinPcap. On Windows 10/11 and Windows Server 2016/2019, Npcap is better suited for analyzing data on the network in combination with Wireshark. After starting Wireshark, the first step is to prepare the program for the test (as is true for other tasks you perform with Wireshark, not just for analyzing MITM attacks). The most important functions can be found in *Capture | Options*. Clicking the *Manage Interfaces* button opens a dialog with the local interfaces that you can use for monitoring. The *Input* tab provides the network interfaces that Wireshark uses for sniffing.

Equally important is *Edit | Preferences | Capture*, which is where you select the default network interface you want Wireshark to monitor. The *Update list of packets in real time* and *Automatic scrolling in live capture* options ensure that the currently captured packets are

always displayed in the window. The *Name Resolution* sidebar option in the Preferences dialog is also important. You will want to enable the *Resolve network (IP) addresses* option. Wireshark will then attempt to display the names of the devices for the IP addresses it displays.

It is crucial to enable promiscuous mode to ensure that Wireshark records all packets on the network, and not just those addressed to its own host system. The corresponding settings are also available under *Capture | Options*, where promiscuous mode is normally enabled, unless you have disabled it.

On the *Output* tab, you can specify the file in which Wireshark will save the capture. To ensure that the files do not fill up your whole disk, you can enable automatic overwriting of older files with *Use a ring buffer with n files*. Saving the captures makes it easier to analyze MITM attacks later, but it does not stop you capturing the attacks during live analysis.

Sniffing with Wireshark

To start sniffing scans with Wireshark, just click on the icon with the shark

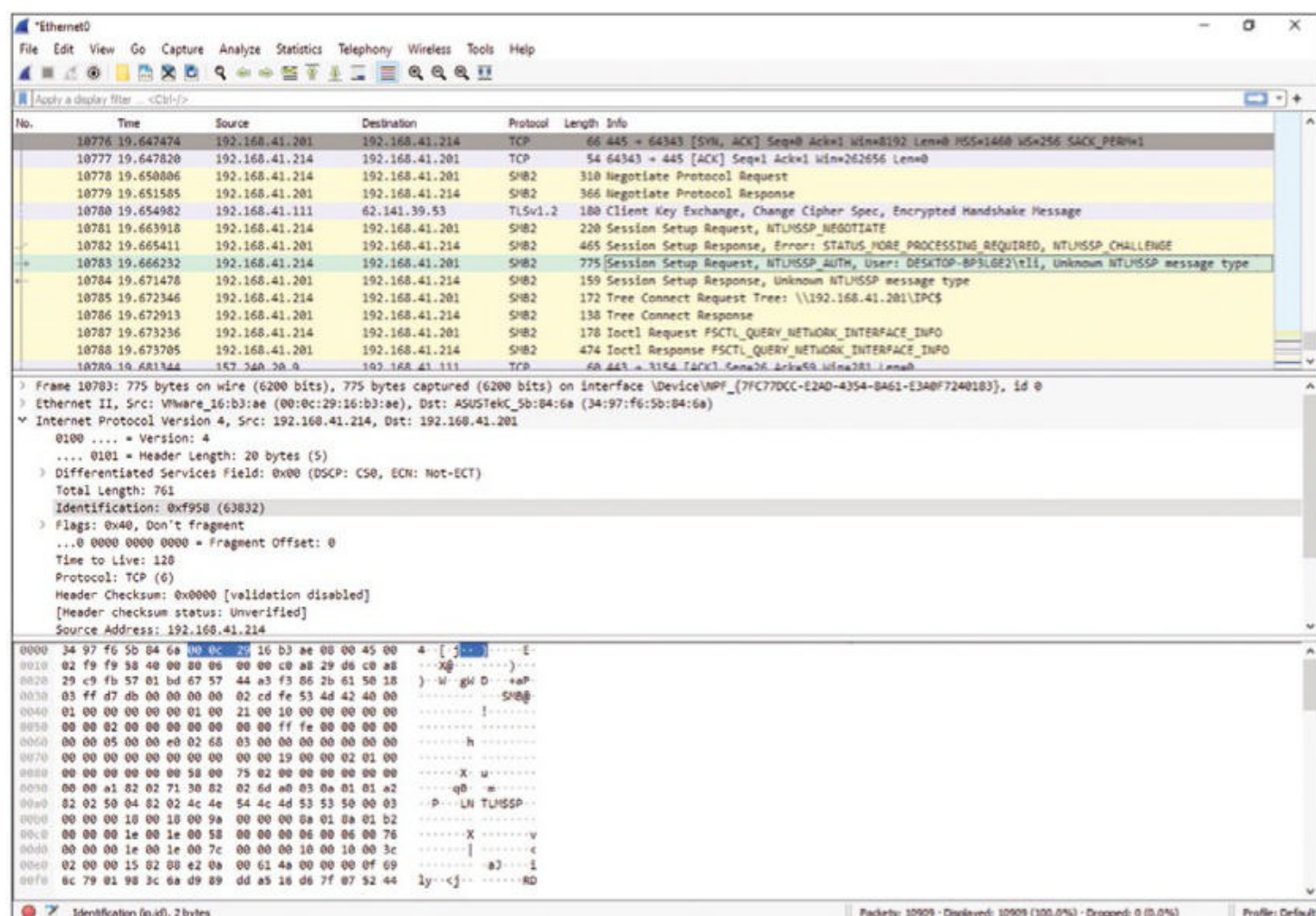


Figure 2: Wireshark can sniff data packets on the network to detect and carry out man-in-the-middle attacks.

fin. Alternatively, double-click on the interface from which you want to record data. Use the Stop icon to stop the scan and the File menu item to save it, unless you have automated this action with the settings as explained above. In the upper window, Wireshark shows the incoming packets and, after selecting a packet, their content in the lower window (Figure 2). You can enable filters in the upper section to ensure that Wireshark lists only the data that interests you. Initial tests for detecting MITM attacks and understanding the corresponding processes can consist of monitoring how a new IP address is queried by DHCP, which is how clients communicate with the network and DHCP servers communicate with clients. The data can be displayed on-screen thanks to Wireshark. Sniffing other information that clients send to servers or to other endpoints basically works this way, as well. Once you have started sniffing the traffic, you can renew the IP address on a computer that needs an IP address while running in Windows with:

```
ipconfig /release
ipconfig /renew
```

The data traffic triggered by this can then be captured. Clicking on the *Protocol* column lets you sort, even without saving files or setting filters. Selecting the *DHCP* protocol helps you find the individual messages between the client and DHCP server.

Wireshark’s bottom window shows the IP addresses and names of the computers involved. Other data packets can be read in a similar way.

Simulated Ettercap MITM Attack

To understand and ultimately defend yourself against an MITM attack, it can be helpful first to simulate an MITM attack yourself. Always keep in mind that this kind of experiment on a third-party network – including public WiFi – is likely to be punishable by law. On your own network, however, the security functions and barriers on the managed switches can easily be tested. Tools such as Wireshark in combination with Ettercap [2] help to flood the network traffic with fake ARP data. The tool is available for Linux and is included in the Kali Linux distribution, as is Wireshark. On Ubuntu, install Ettercap with the commands:

```
sudo apt update
sudo apt install ettercap-common
```

After starting Ettercap, you can start the sniffing process and display the list of local network hosts (Figure 3). Special settings are not necessary. Ettercap then displays the network devices it has found, which you can use for attacks. To start an MITM attack, click on a computer in the host list and select the *Add to Target 1* tab. For an effective test, create a share and a text file with arbitrary content on the

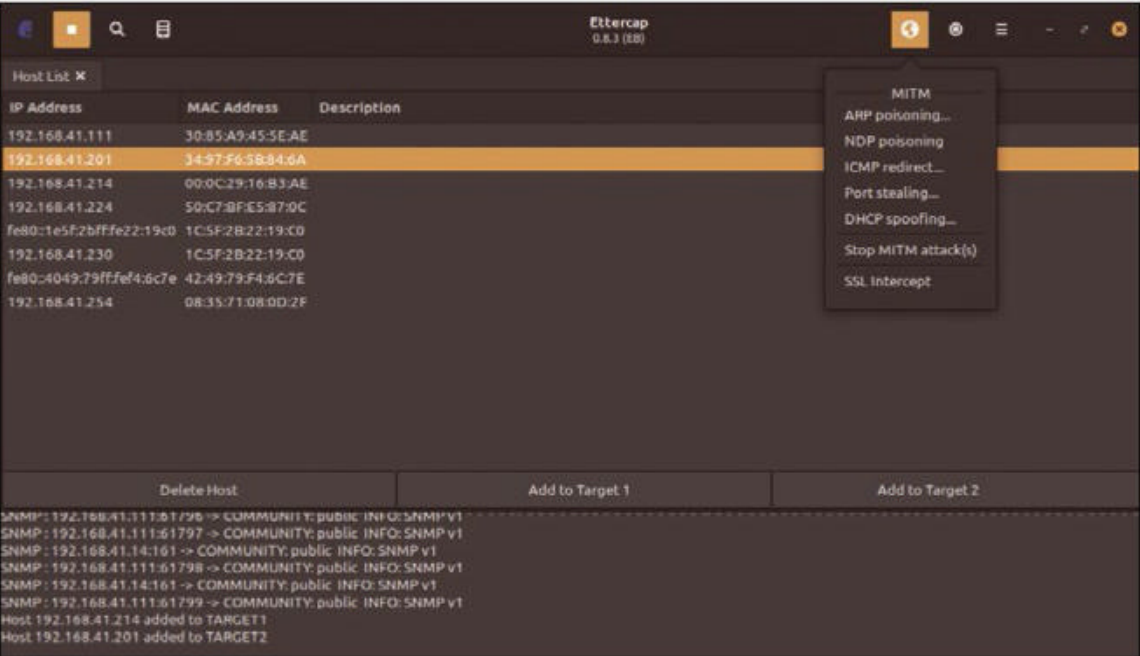


Figure 3: Ettercap helps perform MITM attacks, which you can then analyze with Wireshark.

Another Analysis Tool: XArp

In addition to Wireshark, tools such as XArp help detect fake entries in ARP tables. A combination of different tools can be useful, which together perform a comprehensive analysis or stress test on your own security architecture. One way to detect this kind of attack is to keep a close eye on the ARP table on the victim’s computer. XArp does just that, effectively helping to detect ARP spoofing. Unfortunately, XArp is no longer being maintained [3], although it might persist in distribution repositories, or someone might eventually revive the project.

computer. You can then track access to the share, see when the file is opened, and view its content in Wireshark. Check the ARP cache on the computer beforehand with `arp -a` and make a note of the original MAC address of the computer with the share (see the “Another Analysis Tool: XArp” box). In the attack, swap the MAC address of the original computer with the MAC address of another computer, in this case the one on which you launched Ettercap. Up to this point, Ettercap has not performed any actions but has only read data on the network, just as an attacker would do. Next, select another computer that you want to sniff for the test and click *Add to Target 2*. The target definitions can also be seen at the bottom of the window. The Ettercap computer can now sniff the data between the two devices, and you can, in turn, analyze the operations with Wireshark. In Ettercap’s upper right menubar is an icon with a globe. If you click on it, you can choose from different MITM attacks. To test an attack, it is best to select *ARP poisoning* and confirm that you want to start. The attack is now active and can be observed with Wireshark. Ideally, you will want to launch Wireshark on the computer that is running Ettercap. This attack can also be done with Kali Linux, as mentioned before; both tools are integrated. At any time, you can stop ARP poisoning in Ettercap or define other targets. After stopping the attack, the selected target systems again have the correct MAC address assignments after a short time.

Laughing Third Party

Launching Wireshark in parallel on the computer that you have defined as Target 2 is the easiest way to trace the attack. Open the share you created earlier and the file on the Target 2 computer, which is exactly what users would do when accessing data on the network. The two Wireshark instances capture the actions performed in the background.

If you again query the ARP cache on the Target 2 computer by typing `arp -a`, you will see that during an active MITM attack courtesy of Ettercap, the MAC addresses for Target 1 are identical to those of the Ettercap computer. The MITM computer has succeeded with its ARP attack and can spoof another computer. The client you defined as Target 2 assumes that the Kali computer with Ettercap is the Target 1 computer with the active share, allowing traffic to be recorded on the Kali computer, even though the data is running back and forth between Target 2 and Target 1 and the Kali computer is not involved – a typical MITM case. Other computers will not notice this activity because the attack does not disturb the network.

The entries you have made let the computer with Ettercap and its active Wireshark instance read data that is exchanged between Target 1 and Target 2. If the data is not encrypted, the Wireshark instance on the Ettercap/Kali client will help you extract the content of the data packets. You will find the corresponding captures on the Kali/ Ettercap computer. Closing Ettercap on the MITM

machine also ends ARP poisoning, and the attack is no longer visible.

Filters

Wireshark is as useful a tool for performing MITM attacks as it is for analyzing them. For this reason, it makes sense to take a close look at the tool's capabilities. One important feature is the filters: If you enable an ARP filter in Wireshark, using the example of the attack described previously, you can focus on the ARP-related network traffic (**Figure 4**). If you then use the SMB or SMB2 filter, you will also see the SMB traffic between the clients. With the SMB filter, all exchanges between Target 1 and Target 2 show up, including the content of the text file created and opened for this test. Wireshark also has the filters *arp*, *duplicate-address-frame* and *arp*, *duplicate-address-detected*, which tell Wireshark to display, from a saved or live capture, the packets that have duplicate MAC addresses for different IP addresses. Precisely this information can be seen in the *Info* column. If you find such packets on the network, you can assume that an attacker is

trying to duplicate MAC addresses. If you click on such packets, the original MAC address of the respective systems can also be found during the analysis.

Conclusions

Wireshark is the ideal tool when it comes to capturing network traffic and can also be used to analyze MITM attacks. With its various filters, you can determine whether data on the network shows signs of such attacks. Together with software such as XArp and Ettercap, you can perform stress tests on your own network to check your internal security situation. ■

Info

- [1] Wireshark: [\[https://www.wireshark.org\]](https://www.wireshark.org)
- [2] Ettercap: [\[https://www.ettercap-project.org\]](https://www.ettercap-project.org)
- [3] XArp: [\[http://www.xarp.net\]](http://www.xarp.net)

The Author

Thomas Joos is a freelance IT consultant and has been working in IT for more than 20 years. In addition, he writes hands-on books and papers on Windows and other Microsoft topics. You can meet him online at [\[http://thomasjoos.spaces.live.com\]](http://thomasjoos.spaces.live.com).

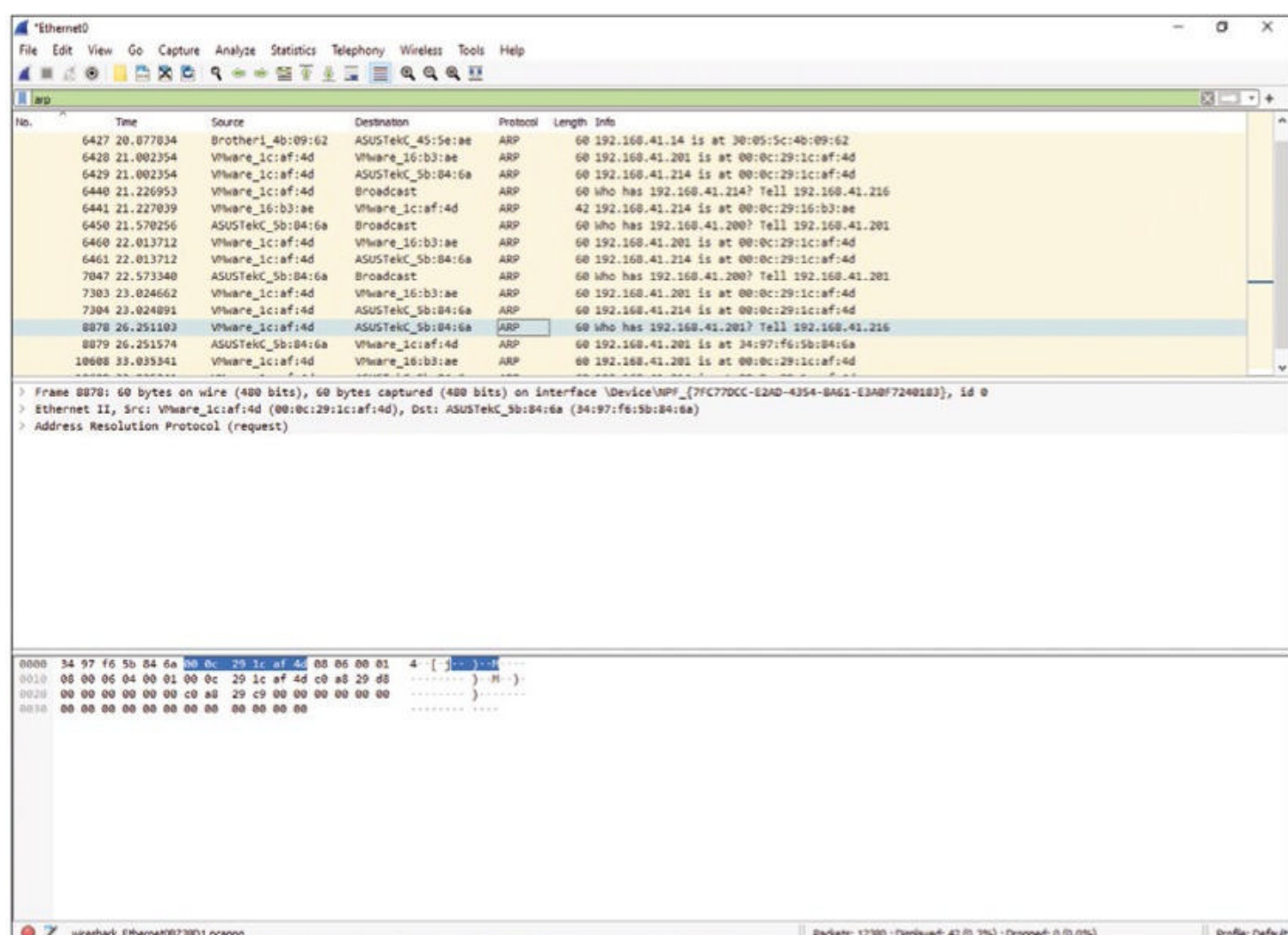


Figure 4: An MITM attack can be detected quite quickly by changing the display filters for ARP and SMB.

Endpoint Security for Windows 10

Well-Tempered Computer



Windows 10, build 21H1, has numerous protection mechanisms out of the box. We look at the option for delaying updates, the components and features of Microsoft Defender, and recommendations for hardening the operating system. By Marc Grote

Microsoft introduced a number of new security features in Windows 10, but they are not available in all variants of the operating system. For example, features such as Windows Defender Device Guard – now Microsoft Defender Application Control – or Microsoft Defender Credential Guard are only available in Windows 10 Enterprise E3/E5; Microsoft Defender for Endpoint – formerly Advanced Threat Protection – is only available with Windows 10 Enterprise E3/E5, Microsoft 365 E5 Security, and Microsoft 365 E5. Also not to be ignored is that Microsoft only allows the Enterprise version to use group policies that can configure the Windows Store.

Windows Update for Business

The monthly patch day still causes excitement among many administrators, as does the question as to whether everything will continue to work as it did before the update. Microsoft has changed the update cycle for Windows 10. Apart from the monthly critical updates, the company releases optional updates at

different times in the second half of the month. Therefore, you can concentrate on installing the critical updates and install the optional updates at a later point in time, once their compatibility with the IT infrastructure has been successfully checked. Windows Update for Business [1], the update process for business customers, includes what are known as update rings, which you can use to specify the order in which you want to patch end devices and servers. These rings let you, for example, patch only unimportant computers or special test machines in an initial update wave. Update rings also allow systems to be patched as a function of how they interact. For example, a domain controller can be patched first, followed by an Exchange server that requires the Active Directory (AD) services to work properly. Windows Update for Business also lets you define maintenance windows during which computers receive updates, so you can select the time windows when the service interruptions associated with the update installation, in the form of computer or service restarts, will have little or no

effect on your operation. IT managers can use local settings on the client or group policy to delay updates.

Authentication Options

In addition to the classic username and password option to authenticate the system, Windows 10 provides other options (Figure 1). In work-group environments, for example, a picture password can be used. You can choose a picture for logging in or define various gestures that are known only to you and use them for authentication. Microsoft equates picture passwords with the PIN entry method in terms of security. Windows Hello [2] is a facial recognition feature that automatically logs the user into the operating system when a known face is detected. As an alternative to facial recognition, the eyes (iris) or fingers (fingerprint) can be scanned for identification. Microsoft decided on this additional authentication option because passwords have long since ceased to provide sufficient security if users do not implement all the requirements for their secure use. You need the right

kind of device to run Microsoft Hello, such as an integrated iris or finger scanner. In AD environments, Windows Hello can be implemented with the help of group policies.

Microsoft Passport is a multifactor authentication (MFA) system that uses a PIN or biometrics (provided by Windows Hello) in conjunction with encoded keys from a device for authentication. Users can use it to authenticate against a local AD, Azure Active Directory (AAD), or non-Microsoft LDAP service.

Windows 10 clients can also join an AAD and use it as their exclusive authentication source. A client's membership in the AAD then enables single sign-on (SSO) to various services, such as Office 365 in the Microsoft cloud. In environments with an on-premises AD and AAD, a synchronization instance ensures that SSO is still guaranteed for on-premises and cloud resources.

Endpoint Protection

Microsoft Defender is an integral part of Windows 10 and helps protect the computer against malware in two ways:

- **Real-time protection:** Microsoft Defender blocks malware that tries to install or run on the PC and notifies the user. The user is also notified if apps try to change important settings.
- **Various scanning options:** Microsoft Defender automatically checks at regular intervals for whether malware is installed on the PC. The scan can be started at a different time, if desired. Microsoft Defender automatically removes or quarantines all suspicious objects detected during a scan. Users and administrators can manually remove objects from the quarantine, or the objects located there are automatically deleted after a definable period of time.

The Microsoft Defender interface largely matches that of older Windows Defender versions. However, the configuration of Defender options is now done in the Windows 10 settings menus and not in the Defender application itself.

In Windows 10, Microsoft Defender has become a strategic product and has been extended with numerous features, which I will describe in more detail in the following sections of this article. The components include Defender Security Center, Defender for Endpoint, Defender Application Control, Defender Credential Guard, Defender Exploit Guard, and Defender SmartScreen.

Defender Security Center

In version 1703, Microsoft has combined various Windows security functions in the Defender Security Center [3] and made them easily accessible in a central interface. The Defender Security Center combines the following functions:

- **Defender features:** virus and threat protection.
- **Device performance and health:** information about device drivers and Windows update states, as well as battery status on mobile devices.
- **Firewall and network protection:** Defender firewall state with advanced security.
- **App & browser control:** configuration in Defender SmartScreen for apps, files, and the Edge browser.
- **Family options:** controlled access to web pages, time control for applications, and allowed access to applications for children.

Defender for Endpoint

Microsoft Defender for Endpoint (formerly Windows Defender Advanced Threat Protection) [4] service for Windows 10, as of version 1607, helps you detect attacks on a network and initiate appropriate countermeasures. Microsoft Defender for Endpoint combines Windows 10 protection

measures and cloud technologies in a single tool.

Features include anomaly detection capabilities (registry, filesystem, and network access) and security analysis capabilities in the Microsoft cloud (Bing and SmartScreen reputation, Microsoft Malicious Software Removal Tool, and threat intelligence). Microsoft Defender for Endpoint also supports security features such as AppLocker or Device Guard. The configuration is handled by group policies, System Center Endpoint Configuration Manager, scripts, mobile device management (MDM) tools, or Microsoft Intune.

Whereas Microsoft Defender for Endpoint is a cloud-based service, Microsoft Advanced Threat Analytics (ATA) is a local service installed on servers in the IT infrastructure to detect suspicious activities on the network in real time. ATA comprises the ATA Gateway or ATA Lightweight Gateway, ATA Center, and other components, some of which are optional. The core functionality involves all domain controllers in the enterprise mirroring their network traffic to the ATA gateway. ATA records the data in a database, clearly displays the findings in the ATA Center, and points out threats and possible countermeasures.

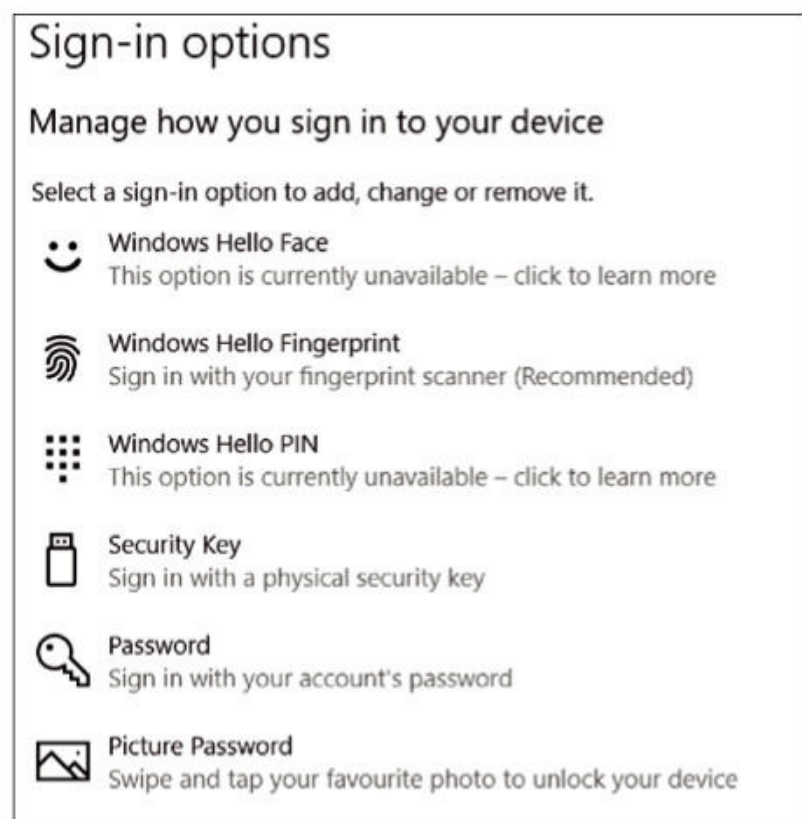


Figure 1: Logon procedures are numerous and vary in usefulness depending on the end device.

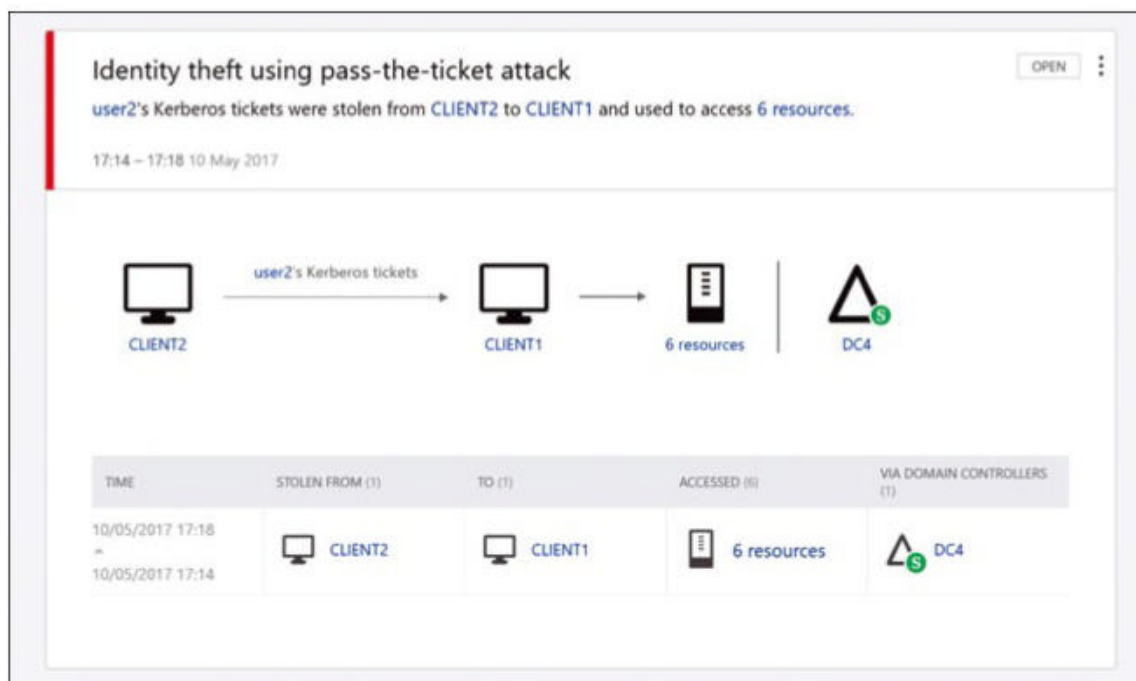


Figure 2: The Advanced Threat Analytics console quickly identifies security incidents in the directory service.

Defender Application Control

The technology behind Defender Application Control (WDAC) [5] is also intended to prevent malware from running on and thus infiltrating the system. The tool is primarily intended to protect against new and unknown malware and Advanced Persistent Threats (APTs). WDAC thus provides increased protection in Windows 10 because it prevents any untrusted or non-digitially-signed app from running, including portable apps that run off a USB stick without a local installation.

Administrators can specify the source from which apps are considered trusted. Both universal apps and Win32 apps can be protected with WDAC in this way. When an application is executed, WDAC checks its trustworthiness. An application is considered secure if it has a digital signature from the manufacturer or the Windows Store, and organizations can define their own applications as secure. Administrators use central policies to determine which apps are trusted and how WDAC should be configured in the enterprise. WDAC protects itself against tampering with hardware and virtualization technologies to isolate the process from all other components. Because WDAC uses Hyper-V as its base, client systems need to meet all the requirements for enabling the Hyper-V role. Compared with other Microsoft technologies (e.g., AppLocker), WDAC plays to its

strengths because it is not possible to infiltrate the check process itself.

Defender Credential Guard

Defender Credential Guard [6] is one of the most important newer features in Windows 10. It also uses Hyper-V virtualization-based security features to isolate secrets such as passwords, password hashes, and Kerberos ticket-granting tickets and gives access only to privileged system processes. NTLM (new technologies LAN manager) hashes or the well-known pass-the-hash or pass-the-ticket attacks are no longer possible in this case (Figure 2). Defender Credential Guard requires the use of Windows 10 Enterprise or Windows Server 2016 and has the following hardware requirements:

- Virtualization-based security (VBS): VBS requires a 64-bit CPU, enabled CPU

virtualization extensions, extended page tables, and a Windows hypervisor.

- Secure boot
 - Trusted Platform Module (TPM) 2.0 (recommended)
 - Unified Extensible Firmware Interface (UEFI) lock (recommended)
- Defender Credential Guard is configured by group policies.

Defender Exploit Guard

Defender Exploit Guard (WDEG) [7] is provided by Microsoft starting with Windows version 1709. WDEG provides a set of host intrusion prevention features to reduce the attack surface on user applications. It is the successor to the Enhanced Mitigation Experience Toolkit (EMET), which Microsoft also makes available for free download for older versions of Windows to protect against exploits in applications that use techniques such as data execution prevention, address space layout randomization, structured exception handling

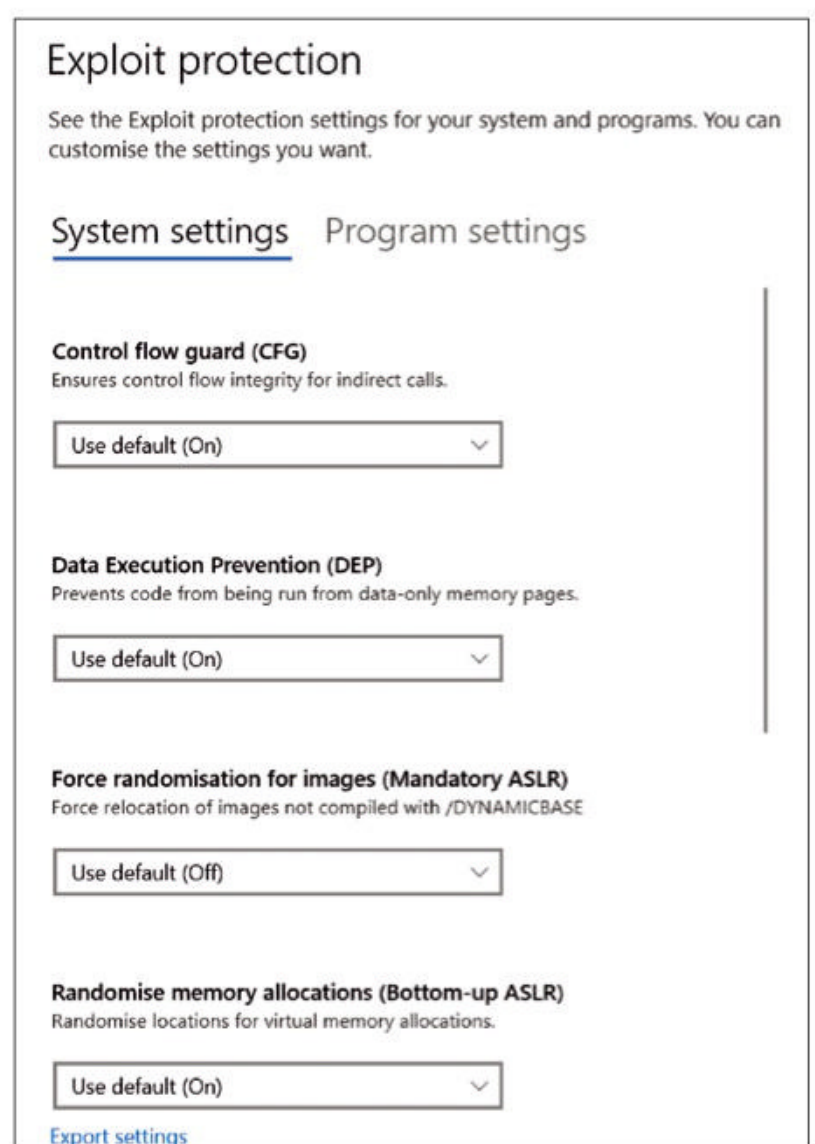


Figure 3: Defender Exploit Guard is configured in the Defender Security Center.

overwrite protection, and others. WDEG can be configured in the Defender Security Center (**Figure 3**). Defender Exploit Guard extends the security features provided with EMET to include Control Flow Guard, which provides protection against memory corruption by restricting where applications can execute program code and extends existing exploit prevention technologies.

Defender SmartScreen

SmartScreen has been part of the operating system on clients and servers for several Windows versions and was renamed Microsoft Defender SmartScreen [8] in Windows 10. The tool helps prevent access to websites that have been reported as phishing or malware spreaders and blocks the download of potentially dangerous files. Defender SmartScreen determines whether a web page is potentially dangerous as follows:

- Analyzes web pages visited and searches for indications of suspicious behavior. If this is the case, SmartScreen displays a warning page and advises caution.
- Compares web pages visited with a dynamic list of reported phishing and malware portals. In case of a match, SmartScreen also displays a warning that the website may be malicious.

SmartScreen tries to determine whether a downloaded app or app installer might be dangerous in the following ways:

- Compares downloaded files with a list of reported websites and programs with malware known to be unsafe. In case of a match, SmartScreen displays a warning to that effect.
- Compares downloaded files with a list of known files downloaded by many Windows users. If the file does not appear in this list, SmartScreen displays a warning and advises caution.

SmartScreen is configured by Active Directory group policies or a mobile device management tool such as Microsoft Intune.

Recommendations for Windows 10

The majority of successful attacks on systems with Windows 10 can already be detected or prevented with the on-board tools available in the operating system. To make it easier to configure the operating system appropriately, the German Federal Office for Information Security (BSI) recently published recommended actions for securing Windows systems: *SiSyPHuS Win10: Study on System Integrity, Logging, Hardening, and Security-Relevant Functionality in Windows 10*. [9] One focus in creating this was on ease of implementation and practical application. For this reason, the BSI makes the recommended configuration settings available for download as group policy objects that can be imported directly.

In the security analysis, BSI examines the security-critical functions of the operating system. The goal is to be able to evaluate the security and residual risks for using Windows 10, to identify framework conditions for secure use of the operating system, and to create practically applicable advice for hardening and secure use. The recommendations from SiSyPHuS are primarily aimed at federal and state authorities, as well as companies. However, technically savvy citizens can also implement the listed points, depending on the Windows 10 version they are using.

The recommendations, Group Policy objects (GPOs), and other partial results of the study that have already been published are available on the BSI website [9]. The BSI intends to publish further conclusions from other parts of the study successively. The analyses include components such as PowerShell, the application compatibility infrastructure, driver management, and PatchGuard. The subject of the study was Windows 10 Enterprise LTSC 2019, 64-bit, German-language version.

Conclusions

Microsoft has made an effort in Windows 10 to expand the list of

new security features in addition to those already built-in and, as a result, has achieved a better level of protection than in older operating system versions. The semiannual updates and the ever-increasing integration with Azure Cloud make Windows 10 one of the most secure operating systems on the market. Unfortunately, many features require the use of the Enterprise version and cloud integration. ■

Info

- [1] Windows Update for Business: [\[https://docs.microsoft.com/en-us/windows/deployment/update/waas-manage-updates-wufb\]](https://docs.microsoft.com/en-us/windows/deployment/update/waas-manage-updates-wufb)
- [2] Windows Hello: [\[https://support.microsoft.com/en-us/windows/learn-about-windows-hello-and-set-it-up-dae28983-8242-bb2a-d3d1-87c9d265a5f0\]](https://support.microsoft.com/en-us/windows/learn-about-windows-hello-and-set-it-up-dae28983-8242-bb2a-d3d1-87c9d265a5f0)
- [3] Defender Security Center: [\[https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-security-center/windows-defender-security-center\]](https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-security-center/windows-defender-security-center)
- [4] Defender for Endpoint: [\[https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/microsoft-defender-endpoint?view=o365-worldwide\]](https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/microsoft-defender-endpoint?view=o365-worldwide)
- [5] Defender Application Control: [\[https://docs.microsoft.com/en-us/windows/security/threat-protection/device-guard/introduction-to-device-guard-virtualization-based-security-and-windows-defender-application-control\]](https://docs.microsoft.com/en-us/windows/security/threat-protection/device-guard/introduction-to-device-guard-virtualization-based-security-and-windows-defender-application-control)
- [6] Defender Credential Guard: [\[https://docs.microsoft.com/en-us/windows/security/identity-protection/credential-guard/credential-guard\]](https://docs.microsoft.com/en-us/windows/security/identity-protection/credential-guard/credential-guard)
- [7] Defender Exploit Guard: [\[https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/enable-exploit-protection?view=o365-worldwide\]](https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/enable-exploit-protection?view=o365-worldwide)
- [8] Defender SmartScreen: [\[https://docs.microsoft.com/en-us/windows/security/threat-protection/microsoft-defender-smartscreen/microsoft-defender-smartscreen-overview\]](https://docs.microsoft.com/en-us/windows/security/threat-protection/microsoft-defender-smartscreen/microsoft-defender-smartscreen-overview)
- [9] BSI: SiSyPHuS project (in English): [\[https://www.bsi.bund.de/EN/Topics/Cyber-Security/Recommendations/SiSyPHuS_Win10/SiSyPHuS_node.html\]](https://www.bsi.bund.de/EN/Topics/Cyber-Security/Recommendations/SiSyPHuS_Win10/SiSyPHuS_node.html)



eSIMs in the enterprise

Card Change

In this overview, we look at the opportunities eSIM technology offers for the mobile workplace and what IT managers need to watch out for in deployment and management. By Marco Föllmer

The COVID-19 crisis and 2020 have changed the world of work. In response to the virus, many companies introduced long-term home office work models for their teams for the first time. Today, it is clear that the legacy 9-to-5 working model in the office is a thing of the past for many. Even when new people join the company, their first day at work often no longer takes place on the company premises but at home. Employees need to be able to commission their devices into operation as easily as possible while complying with corporate security standards. Devices need to be directly subordinate to a unified endpoint management (UEM) system to ensure that they are configured in line with corporate policies. IT support can't supervise employees while they are at home and would be happy to field as few requests for help as possible. The same applies when a device needs to be replaced. The exchange needs to take place with as little overhead as possible for both users and IT.

Technologies for remote commissioning of devices already exist, and corporations could do worse than deploy them in line with requirements. One such technology is the eSIM, which offers employees, companies, and their IT administration teams the opportunity to simplify mobile work.

eSIM-Enabled Devices on the Rise

The embedded SIM is a chip built into the mobile device that stores one or more profiles belonging to network operators. For this purpose, each eSIM has an eID: a unique number used to authenticate the user on the mobile network. eSIM technology replaces physical SIMs in mini, micro, or nano format and offers many advantages for users. The eSIM looks completely different from a physical SIM and solves some of its challenges, as well.

eSIM technology was first used in a smartwatch by manufacturer Samsung in 2016. In the following years, its use was enabled in smartwatches

by other manufacturers, as well. In 2018, the eSIM then gained further popularity when Google and Apple brought it to their smartphones. Today, increasing numbers of smartphones, tablets, and smartwatches can use an eSIM, and the numbers continue to increase. According to a study by Juniper Research, global usage is expected to rise from 1.3 billion in 2021 to 3.4 billion in 2025 [1]. In the long term, it's possible that these rising numbers could lead to the elimination of the slot for physical SIMs in mobile devices and to eSIMs becoming the sole standard. eSIM technology is already being used successfully outside of mobile communications (e.g., in the IoT and connected car sectors).

Fast Commissioning on the User Side

One advantage of the eSIM lies in simplified operations. A physical SIM first has to be handed over to the employee, either in person or by surface mail. Once received, it has to

Photo by Glen Carrie on Unsplash

be broken out of the packaging and inserted into the device. With the eSIM, these two steps are eliminated, and errors, misuse, and long shipping routes can be avoided.

Employees can enable eSIM profiles regardless of their time zone and location by scanning a QR code. All they need is an Internet connection. In this way, employees can be ready to start work within a few minutes without any intervention on the part of IT support. This advantage saves time and resources for both sides, especially in times of hybrid work models and decentralized IT structures.

Parallel Use of Plans

For employees who travel on business, the IT department can issue an eSIM with a plan for the appropriate country at short notice. The eSIM also often supplements a physical SIM card. For example, a business and a private plan can be used in parallel on the same device. In many cases, this would not be possible without the eSIM because not every device offers the option of inserting two physical SIM cards in one device.

Likewise, the use of multiple business plans is no longer a problem. Depending on the device, up to 10 eSIM profiles can be stored and selected in parallel, which is a major advantage in terms of costs, as well, for companies that have opted for Bring Your Own Device (BYOD) or for private use of business devices (corporate-owned, personally enabled; COPE). Employees also benefit because they don't have to carry two devices for personal and business use.

Benefits in Mobile Security

Compared with the physical SIM, the eSIM boosts security when working on the move. eSIMs come with various security measures implemented by default. The eSIM profile installation on a new device is encrypted to prevent data leaks. The same applies with end-to-end transmission between the eSIM provider's servers and the end device. Additional

security is offered by a feature that only allows an eSIM profile to be decrypted and installed on a device assigned to the user.

The absence of a physical SIM card also reduces the risk of misuse because the eSIM cannot be slotted into another (possibly private) device. If a device does happen to be stolen or lost, IT support can quickly deactivate the eSIM remotely and delete the data from the device within the UEM system.

Wary of Change

All told, the eSIM offers many advantages over the physical SIM, and its use in smartphones in Germany is on the rise, mainly in the private sector, whereas German corporations still mainly use physical SIMs. Organizations with a large number of employees and complex structures in particular are worried that switching from a physical card to an eSIM could mean massive overhead in terms of time, resources, and manual work to set up the new system. After all, administrators need to configure various data (e.g., the eID, email address, device model: information that can usually be viewed on the UEM system in use) to enable an eSIM profile, but the data first has to be retrieved from various sources.

Additional data such as the telephone number, the selected rate plan, or details of the existing card, such as the ICC (the SIM serial number), can usually be found in the provider's customer portal or even stored individually in Excel lists.

The changeover can therefore require many manual steps on the part of the corporation if the data cannot be merged automatically. These steps apply not only when switching from the physical SIM card to the eSIM but also if a device needs to be replaced later, is lost or stolen, or a non-domestic plan is required.

Simplified Management

New software (e.g., the eSIM Business Manager, which EBF developed

in cooperation with Deutsche Telekom [2]) can help in these scenarios. In conjunction with a UEM system, an eSIM manager makes it possible to link the required data from various sources and make the data clearly visible and usable in a portal. Using such portals, administrators can assign an eSIM to a device and remotely enable the eSIM (and also revoke it again if necessary) in a quick and easy way. The software therefore enables both the commissioning of a new eSIM profile and the migration of a physical SIM to an eSIM profile with just a few clicks.

The process is equally convenient on the user's side: Employees can define when they want to migrate to suit their needs and complete the move with just a few clicks. The plan can then be used directly. The software not only helps manage eSIMs in companies but also facilitates parallel management of physical SIM cards, because the portal also displays the details you need to manage physical cards, such as the PIN and PUK.

Conclusions

Corporations, administrators, and IT decision makers need to start looking at eSIM technology today. After all, it offers a huge amount of potential for today's, and tomorrow's, world of work. Employees want and need to work more flexibly: from their home offices, from a hotel room on business trips, or onsite with customers. eSIM technology supports this style of work by offering flexibility and convenience and saving administrative overhead and costs. ■

Info

[1] eSim device installations: [<https://www.juniperresearch.com/press/esim-device-installations-to-reach-3-4-billion>]

[2] eSIM Business Manager: [<https://ebf.com/en/esim-business-manager-taking-endpoint-enrollment-to-the-next-level/>]

Author

Marco Föllmer is a managing partner and founder of EBF GmbH.

Zero Trust as a security strategy

Beyond the Patch

Acceptance of zero trust models like BeyondCorp by Google or LISA by Netflix lags in Europe, where endpoint security is king. We examine why this situation must change by looking into the principles of modern zero trust concepts. By Martin Loschwitz

Even if you don't want to hear it, European IT is not necessarily known for being hyper-innovative – for a variety of reasons. Successful startups, for example, are far less likely to be launched in Europe than in the US, and not because no bright minds with smart ideas are on the east side of the Atlantic, but because of the structures of the industry as such. The much-quoted bon mot “we’ve always done things this way” contains more than a shred of truth. Anyone who has ever experienced a European IT company from the inside will know what I mean when I say that the impression is more of an archeological excavation site than a technology company. Wrapped up in this dilemma is enterprise devotion to endpoint security in a local network, which encounters problems when administering clients outside that network and necessarily feeds the VPN revenue stream. However, rebuilding your infrastructure to implement a zero trust concept will pay off in the long run with less complexity and higher effectiveness,

especially for employees outside the local network, which in today’s environment, can easily be the majority of a work force.

Standards from the Last Century

One area in which this can be seen more clearly than in almost any other is security. Partners from the US or Israel who regularly work with large German corporations (my milieu) are amazed at the standards of security and compliance that are still commonplace in this country. Stating that access to your own email on a smartphone is supposed to be linked to a mobile VPN “because of security” often leads to bewilderment among observers. Likewise, that many large European corporations still force employees to change their passwords on a regular basis raises an eyebrow among others. This confusion is understandable, because it has long been shown that users simply change their existing

password by just one character if worst comes to worst. Supposedly progressive companies have started the next round in the fight against such passwords and check the password against a dictionary or for certain character strings. For example, although “2021” is not allowed in a password and the password manager will protest, it will still accept “2o21” as an entry without any complaints. External observers will start looking for the candid camera when they see all these security measures from the last millennium and realize that two-factor authentication is not mandatory.

We Have Always Done It This Way

Many security and compliance measures in today’s companies are more apparent than real, and this phenomenon can be seen, for example, in the idea of the “secure local network,” which continues to be used unwaveringly by many corporations

Photo by Jack Dong on Unsplash

up and down the continent – even if the representative of an insurance company at the customer’s location cannot issue a contract because the software that communicates with the central systems in the enterprise cannot establish a VPN connection. VPN connections don’t need thick wires, but they do need reliable ones.

You can’t win a pot of gold with a connection hampered by a poor enhanced data GSM environment (EDGE) or because of a completely overloaded 4G network.

Nevertheless, many companies are forcing their employees to use these and other technical measures of dubious benefit. As if things weren’t bad enough, some managers are not afraid to cite data protection blatantly as the cause of the malaise. In the given context, it is not uncommon to hear that the European Union General Data Protection Regulation (GDPR) is to blame because it mandates secure communication in line with “state-of-the-art principles” and “especially because of COVID.” Even before COVID and long before the GDPR, it was simply ignored if 2,500 colleagues had to make their way through the same, way too narrow VPN gateway. The nonsense that is sometimes heard in the corporate security context would often be euphemistically described as “discouraging.”

Endpoint as the Central Building Block

While reading this article, you might be wondering what the rant about large corporations and their sometimes absurd security theater has to do with endpoint security. The answer to this question may be somewhat surprising to some, because it is very relevant. Consistently and correctly implemented security on end devices is a huge building block on the way to a modern security architecture in your company. However, for endpoint security to work effectively, it needs various other factors. Anyone who considers endpoint security to be the only factor in the fight against attackers

and follows the maxim that only the most secure client possible is the last word of wisdom is fatally mistaken. To explain this in more detail, however, I need to briefly digress into the subject of secure networks and explain why this concept has long since become obsolete.

The Secure Network Principle

Admittedly, in many companies, the idea of a secure local network dates back to a time when the threat scenario in IT was completely different from the current scenario. Moreover, anyone who had to plan security for a company at the end of the 1990s did not have even a fraction of the tools available to admins today. Accordingly, most companies took a crude approach: The IT department simply divided the world into good and evil, or, as it was usually implemented, into “them” (evil hackers) and “us” (corporate users).

The people in charge followed through with this segregation with total consistency at all levels of the company’s IT. Most admins still encounter this basic idea in the data center today (e.g., when there is an internal zone containing the most important systems that do not have a direct connection to the Internet). Also in the mix is the demilitarized zone (DMZ), to which access from outside is meticulously regulated by firewalls (Figure 1). The third network segment, the Internet, is evil, ugh, and to be avoided to the greatest extent possible. Larger enterprises and deployments have further refined this principle by segmenting their networks even further.

In this case, the servers with the particularly

important data are not located in the internal zone but in a “very internal” zone, which in turn is separated from what is already an internal zone by a DMZ with an integrated firewall. The division into segments has one flaw: Once you are on the “local network” with a client, you are given access to the other components of the setup, even if your intentions are dishonest. In environments of this type, a client gets access to infrastructure simply because it is in a certain physical or logical location.

What may not sound so dangerous at the data center becomes a real problem in the context of clients. Because the principle of a secure local network is so tempting at first glance, many companies have quickly applied it to their client landscapes, too. They are actively supported by many service providers who generate horrendous revenues with VPN solutions (Figure 2). The result is that some of a company’s services can no longer be used in any meaningful way because they are practically inaccessible. If the company’s own service laptop breaks down, an employee is practically unable to work until they are issued replacement hardware that is allowed to talk to the VPN.

Why the Secure Network Principle Fails

The big problem with the secure network idea is that it is patently false and didn’t work well even years and

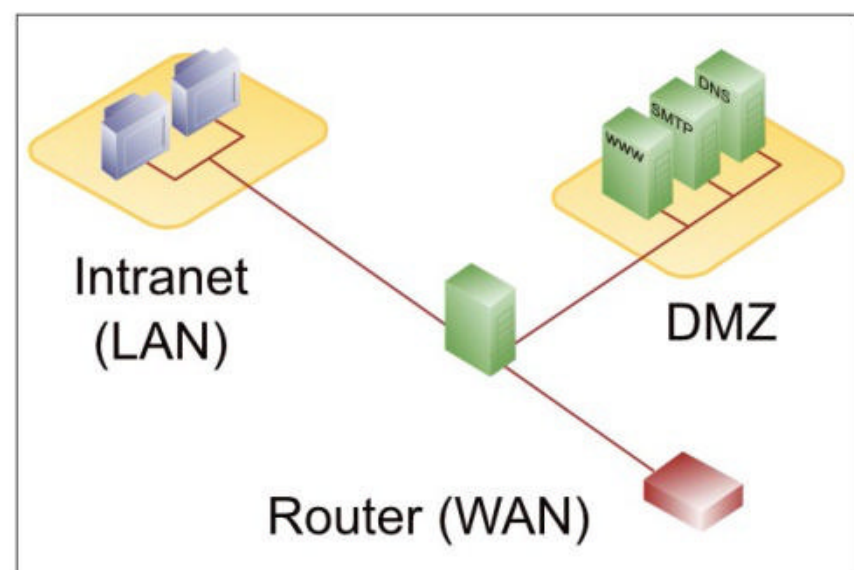


Figure 1: DMZ strategies are one of the oldest implementations of the secure local area network principle, but in today’s IT, this approach has had its day.

decades ago. The idea that a client of any kind is trustworthy simply because it has access to a certain network segment is nonsense. The rule implicitly assumes that the users who use the respective clients know what they are doing and can handle the privileges they have been granted. The opposite is the case, as various examples from the past prove.

The stories in which security researchers succeeded in gaining access to employees' systems with floppy disks labeled "porn" left in company restrooms are the stuff of legend – because the floppies naturally contained viruses. In this case, the entire security strategy is doomed, because once an attacker has access to a system inside the setup, most security precautions no longer work.

Other examples of the nonsensical nature of the secure network approach are the countless cases of hacked Docker containers and OS images that are now roaming the net. Admittedly, from the user's point of view, it is tedious to build your own OS image from the providers' sources. The temptation to organize a ready-made image off the web is great, even if the image might be of dubious provenance. Admins struggle with the results of such thoughtlessness every day. If the ready-made image contains

not only what it says on the box but also a virus or a cryptocurrency miner that generates traffic without end, security has already gone to pot.

To aggravate things, modern IT is extremely complex – companies that deal with IT forensics consequently make a lot of money, and for good reason. After all, in a software-defined environment, tracing the paths attackers have taken after illegally accessing a network element is difficult and often even impossible. Entire government agencies, hospitals, and companies go offline for days and weeks after such problems, until the network is patched back together with some degree of difficulty.

Security Until IT Shutdown

Companies with legacy security strategies are usually aware of their shortcomings, and the most abominable workarounds tend to sprout up: Mail servers are hidden in even deeper internal network segments, or complex firewall constructs are created that have to be extended by rules with hyper-complex processes. At the end of the madness is one big something that the admins of a company often can no longer keep track of in its entirety, partly because it is unclear which security measures take effect

where and when. The usability of a company's IT services falls victim to the quest for the greatest possible and most complete security.

Administrators then come to the bitter realization that a client being located in a certain place (or not) cannot be a valid indicator for or against protective measures. Tools that propagate this approach (e.g., VPN-based products) are a relic of the past, propagating dubious security strategies and limiting the usability of services but offering few effective benefits.

Once this incontrovertible fact has seeped in, it sharpens the eye for real opportunities. Once again, industry leader Google has shown how this can look. At Google, the idea of the secure network became obsolete years ago. Instead, the company is pursuing a strategy known as BeyondCorp [1]. The core aspect of this strategy is that the focus is no longer on the individual

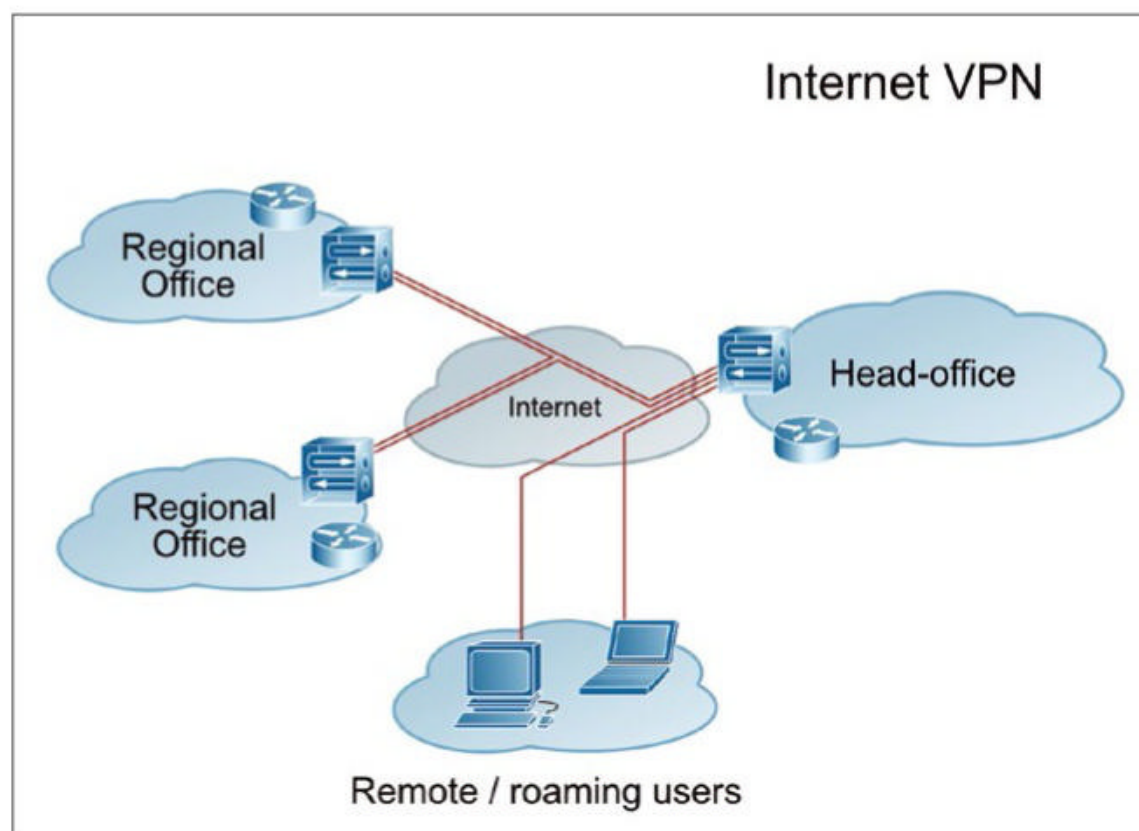


Figure 2: VPN networks also propagate the principle of the secure local network but fail to acknowledge the complex attack scenarios of today.

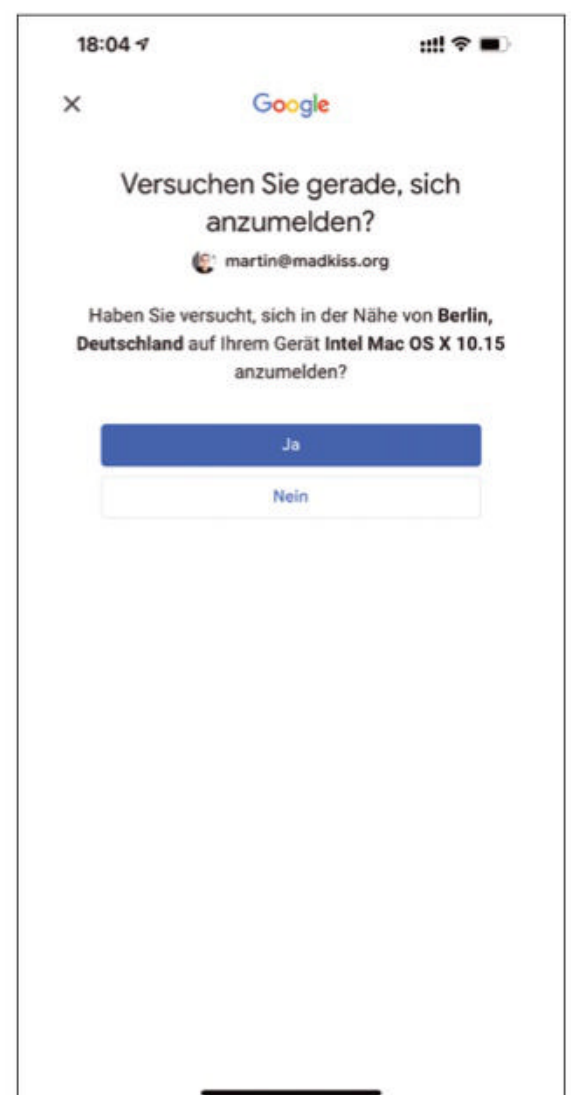


Figure 3: In BeyondCorp environments, clients are granted access only if they can identify themselves as authorized on the basis of multiple factors. Here, the app is asking if I am trying to log in and whether I have tried logging in near Berlin, Germany, on my Intel Mac OS X 10.15 device.

client, but on its applications and the way they communicate with the existing infrastructure.

BeyondCorp

I want to take the wind out of the sails of an argument often raised against the BeyondCorp principle right from the start: Google's solution does not mean that admins completely dispense with security precautions. Quite the opposite: The servers that Google operates for services like Gmail or Google Drive are subject to explicit, very strict, and tightly meshed rules. The point is far more that Google does not differentiate between "internal" and "external" connections for its own services. Instead, every client is basically considered untrustworthy.

Therefore, an individual client does not automatically receive more rights than others simply because it has a

certain network address or is located at a certain physical location (Figure 3). Rather, any employee can access their resources from any client in the world at any time, as long as the client with which this happens adheres to a few rules that make it trustworthy, and it is those rules that make up the BeyondCorp principle. Incidentally, Google's approach has inspired a number of other tech giants. Netflix also says that it now uses a zero trust architecture (i.e., a system in which the provider's services do not trust a client at all, regardless of where it is). Netflix calls the principle location-independent security approach (LISA) and has admittedly invented a far nicer name than Google. BeyondCorp, LISA, and most zero trust approaches use the same basic principles; therefore, in the further course of this article, I will no longer distinguish between the implementations.

What BeyondCorp Is All About

BeyondCorp's central approach is to approve access to any service only if it meets several requirements. It must always be authenticated; that is, the requesting client must have proven that it is authorized for the requested resource. High standards apply. Additionally, access must be authorized. A central rights management system must therefore specify that the client is allowed access to the resource it is currently trying to access.

Encryption of the connection is mandatory, not optional. At least this point is taken for granted from today's perspective; however, if you go back 10 years, you could still find some web stores that did not have SSL certificates to process orders securely. Google's BeyondCorp concept would not, of course, allow connections without encryption, because

Shop the Shop  shop.linuxnewmedia.com

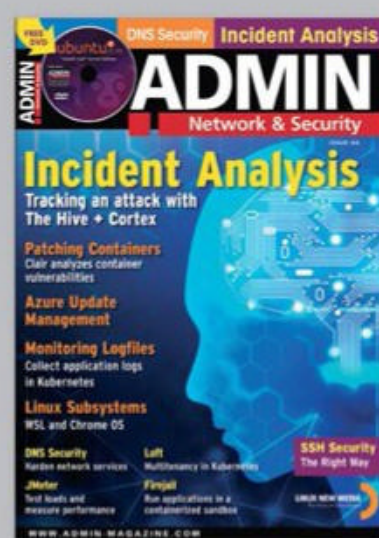
Discover the past and invest in a new year of IT solutions at Linux New Media's online store.

Want to subscribe?

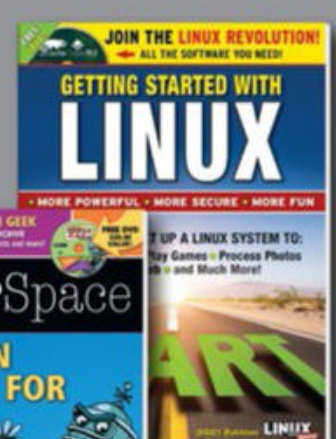
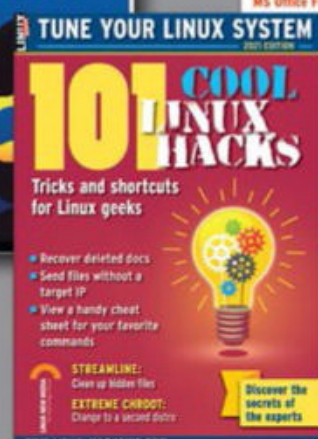
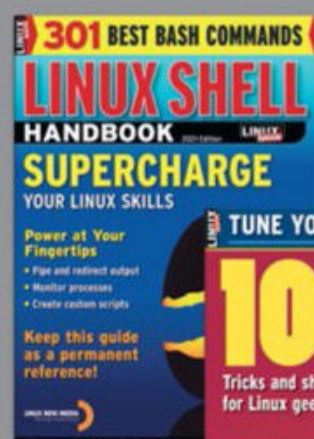
Searching for that back issue you really wish you'd picked up at the newsstand?

➤ shop.linuxnewmedia.com

DIGITAL & PRINT
SUBSCRIPTIONS



SPECIAL EDITIONS



that would mean that any bad guy with access to the line between client and server – often several thousand kilometers in length – could read the data traffic. For a client to be trusted from the service’s point of view, it has to be able to use encrypted connections.

However, the requirements for the client and its user are not yet complete. In the context of a BeyondCorp procedure, a user is only granted access to a resource if it is possible to establish a direct connection between them, their environment, and the technical client. What is stated in the BeyondCorp guidelines in somewhat cryptic terms generally means two-factor authentication (2FA) in everyday life.

In this way, Google consequently eliminates the eternal password problem: If 2FA is activated for access, it is initially irrelevant if a user’s username and password fall into the hands of attackers. For them to log in and access the client’s data, they need the second factor – usually a smartphone – with a suitable app that can be used to grant approval for the respective access. Authentication with an SMS text code has rightly fallen into disrepute today, and applications such as Google’s Authenticator offer better alternatives.

For its own services, Google now goes so far as to display a warning in the respective apps (e.g., Gmail) if the same account logs in on another device. If the user does not confirm this access on their own smartphone, Google rejects it. As a rule, smartphones are also secured against access by strangers – for example, by an unlock pattern or facial recognition, which is practically a third factor. Even if the bad guys were to get their hands on the smartphone in addition to the combination of username and password, they would still not be able to do anything with the stolen device.

Strict Regime for Clients

A large portion of endpoint security products only begin to make some

form of sense in the context of BeyondCorp. After all, only if a service can make decisions about allowing or denying the connection on the basis of various properties and parameters of the endpoint in question does the admin gain genuine control over the individual clients, which admittedly requires a bit more than just properly configured services.

For good reason, mobile device management is a fixed component of all LISA and BeyondCorp environments. If a smartphone is lost, the respective owner (e.g., the company) can remotely delete the device and render it unusable, making BYOD scenarios possible: Anyone who wants to use their own iPad can do so if they place the device under the auspices of the relevant compliance and security team. As a rule, this condition does not restrict the functions, but the user does relinquish some of their sovereignty over the device.

Zero Trust - No Alternatives

Anyone who has ever struggled as I have with the sometimes unusable infrastructure services of German corporate IT will sooner or later come to the conclusion that BeyondCorp is an absolute must-have. Many companies shy away from this realization because it requires a huge rebuild of their own infrastructure. For this reason, it is not possible to share tips or advice here with regard to individual components.

Anyone who gets around to implementing a zero trust concept for their own company usually starts on a green field and redesigns their IT application landscape, leading to unease and costly outlays. Google itself, however, proves with statistics from its own business that BeyondCorp does pay off in the long run. Higher employee effectiveness, a less complex infrastructure to maintain, and fewer sprawling processes in the company are just a few of the benefits that ultimately show up in the bottom line. Of course, Google wouldn’t be Google if the company hadn’t long ago

bundled BeyondCorp into a boxed product that is available for a price. The provider even offers migration consultancy to interested customers. Google has long since ceased to be the only player on the market. If you do not want to commit to Google’s services, and they do play a major role in BeyondCorp, you will find similar approaches and complete packages on offer from other providers. Additionally, a market of consulting companies now exist that can implement similar concepts with on-premises components in the customer’s data center.

Conclusions

Endpoint security can only work if the device that the user relies on is part of a tight network of security functions. From today’s perspective, it is grossly negligent simply to assume no danger from a client on the VPN. Anyone who has had to deal with procedures of this type from an admin point of view will be aware that it can make daily operations extremely tiresome.

Truly, most companies in Europe are still fighting against the realization that the principle of the secure network has had its day. However, this strategy is not sustainable. The principle of “better late than never” applies here. If you decide to implement a comparable strategy today, you have the option, or at least a perspective, of getting away from the IT of the past. However, if you continue to resist, you can expect to be faced with an increasingly difficult-to-maintain and convoluted infrastructure. ■

Info

[1] BeyondCorp: [\[https://cloud.google.com/beyondcorp\]](https://cloud.google.com/beyondcorp)

The Author

Freelance journalist Martin Gerhard Loschwitz focuses primarily on topics such as OpenStack, Kubernetes, and Ceph.



REAL SOLUTIONS for REAL NETWORKS

ADMIN is your source
for technical solutions
to real-world problems.

Improve your admin
skills with practical
articles on:

- Security
- Cloud computing
- DevOps
- HPC
- Storage and more!

**GET IT
FAST**

with a digital
subscription!

**SUBSCRIBE NOW AND GET
THE ADMIN ARCHIVE DVD
FREE WITH ISSUE #65**

6 issues per year!

..... **ORDER NOW**

shop.linuxnewmedia.com



Attack and defense techniques

Cybersecurity Know-How

The MITRE ATT&CK and D3FEND knowledge databases provide useful techniques for securing your IT infrastructure. By Matthias Wübbeling

IT security affects many different areas of a company. Trying to identify possible attack vectors for each area in advance and protect the IT infrastructure with effective countermeasures can be a Sisyphean task, especially for companies without a Security Operation Center (SOC). When it comes to implementing security measures, knowledge and experience are important.

MITRE, a nonprofit organization that operates various research facilities on behalf of the U.S. government, provides comprehensive information on IT security. MITRE developed the CVE system, for identifying and assigning

unique identifiers to vulnerabilities, and also developed STIX and CyBox, which are used to exchange threat information and attack indicators. The MITRE ATT&CK and D3FEND knowledge databases offer techniques that let you retrace an attacker's steps, as well as prevent attacks in the first place. Here's how to use these techniques to secure your enterprise IT.

ATT&CK

Released to the public in 2015, MITRE's ATT&CK framework provides a knowledge database of attack

techniques and methods enriched with details about hacker groups and their individual procedures. For an initial overview of the knowledge base, visit the ATT&CK website [1] and *Matrices* in the top menubar. In the sidebar on the left, the ATT&CK dataset is broken down by *Enterprise* (enterprise IT), *Mobile* (smartphones), and *ICS* (industrial control systems). Both *Enterprise* and *Mobile* are directly integrated into the interface, while *ICS* currently still links to a wiki with more information. The matrices, sorted chronologically, are based on Lockheed Martin's Cyber Kill Chain [2]. For the

Photo by Artem Bryzgalov on Unsplash

Enterprise Matrix, you'll find preparatory techniques in the Reconnaissance phase on the left. On the far right, you'll find an attacker's potential activities after successfully hijacking a system under the Exfiltration and Impact phases.

In order to take a closer look at individual techniques, I will focus on the Phishing for Information [3] technique listed under the Reconnaissance phase. If you click on *Phishing for Information*, you will be taken to a detailed page with further information. There, you will learn how attackers send phishing messages to potential victims in order to obtain more information from companies, such as login data for computer systems. Keep in mind that this technique differs from the Phishing technique listed under the Initial Access phase, where the objective is to send executable code (malware) as part of a phishing campaign.

In the Procedure Examples section, you'll find examples of groups that have used such techniques in the past, often with brief comments. The Mitigations section lists two potential countermeasures: Software Configuration and User Training. The Software Configuration countermeasure references SPF, DKIM, and DMARC (see also [4] and the "Trustworthy" article in this issue) in order to limit the success of legacy email phishing. The User Training countermeasure relies on training employees to detect and thwart phishing attempts. The Detection section primarily describes automated options for detecting the technology, which you can use for protection and also for creating situation reports. The Reference section contains sources and further information, including scientific papers, reports, and articles for further research.

D3FEND

Analogous to the attack techniques specified in the ATT&CK framework, the MITRE D3FEND [5] knowledge base provides you with information

from the defender's point of view. The D3FEND matrix has five different techniques for securing your computer systems. On the left side of the matrix, Harden lists four technique categories for securing systems and resources before using them. You'll find methods for compiling software, securing passwords, and encrypting messages, as well as how to use TPM-based boot protection or hard disk encryption.

The Detect category offers techniques that can be used to detect malicious activities or to evaluate general activities on your network. Staying with the phishing example, clicking on *Sender MTA Reputation Analysis* (located under the Message Analysis subcategory) takes you to a page with techniques for evaluating message transfer agents (MTAs). For instance, you can determine a trust rating for the sender MTA based on past behavior, such as receiving prior emails from an MTA, the domains used as sender domains, or the number of reply emails from an MTA. Each technique entry in the D3FEND database contains direct links to the relevant ATT&CK techniques, as well information about implementations or patents that cover corresponding techniques. In this way, you can jump back and forth from one MITRE database to another to quickly determine whether you thought of everything during hardening. The D3FEND knowledge base is a logical complement to the ATT&CK database.

Systematic Use

For a deeper insights, or to collect and process information in a targeted way, ATT&CK offers additional connections. For example, if you already use a tool for analyzing STIX data, you can import STIX datasets prepared by MITRE directly from the repository [6]. The ATT&CK navigator can be used to mark relevant entries during research and to display correlations, thus letting you plan and trace

your progress in securing the infrastructure. This helps you delegate individual tasks within your team and include quick overviews of the status quo in your reports.

Conclusions

MITRE ATT&CK and D3FEND knowledge bases provide you with comprehensive insights into cybersecurity techniques. You can use these databases as a foundation for securing your enterprise IT infrastructure or simply as a reference for the next capture-the-flag event. If you find something missing in the knowledge databases, you can contribute content for future versions. In this way, expert knowledge can be bundled and made available to as many companies as possible. ■

Info

- [1] ATT&CK: [\[https://attack.mitre.org\]](https://attack.mitre.org)
 - [2] Lockheed Martin Cyber Kill Chain: [\[https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html/\]](https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html/)
 - [3] Phishing for Information technique: [\[https://attack.mitre.org/techniques/T1598/\]](https://attack.mitre.org/techniques/T1598/)
 - [4] "Hardening Network Systems with DNS" by Matthias Wübbeling, *ADMIN*, issue 66, 2021, [\[https://www.admin-magazine.com/Archive/2021/66/Hardening-network-services-with-DNS\]](https://www.admin-magazine.com/Archive/2021/66/Hardening-network-services-with-DNS)
 - [5] D3FEND: [\[https://d3fend.mitre.org\]](https://d3fend.mitre.org)
 - [6] ATT&CK STIX data: [\[https://github.com/mitre-attack/attack-stix-data/\]](https://github.com/mitre-attack/attack-stix-data/)
-

The Author

Dr. Matthias Wübbeling is an IT security enthusiast, scientist, author, consultant, and speaker. As a Lecturer at the University of Bonn in Germany and Researcher at Fraunhofer FKIE, he works on projects in network security, IT security awareness, and protection against account takeover and identity theft. He is the CEO of the university spin-off Identeco, which keeps a leaked identity database to protect employee and customer accounts against identity fraud. As a practitioner, he supports the German Informatics Society (GI), administrating computer systems and service back ends. He has published more than 100 articles on IT security and administration.

Configuring X Window input and output devices

Tailor-Made

Two command-line tools, `xrandr` and `xinput`, let you optimize your X Window display from the terminal. By Thorsten Scherf

An X Window System provides several components to allow users to interact with a graphical interface. An X server lets applications, also known as X clients, use a graphical display within windows. The window manager determines the look and feel of such an interface, as well as takes care of how the windows are handled (e.g., enlarging, reducing, or closing them). A desktop manager, which is ultimately responsible for bringing order to this kind of graphical interface, displays icons, menus, panels, and other elements on the desktop. Even though various graphical tools exist to optimally adapt your existing hardware to the X Window System, you can also do this from a terminal using command-line tools. You simply need to use the right tool to make the setting you need. In this article, I will show how to adjust some typical display settings from the terminal using `xrandr` and `xinput`.

Setting up the Display

One problem that occurs time and time again relates to the correct display resolution. If you use several monitors or also use a projector, it is often difficult to determine the correct order of the devices so that you can use a mouse to easily switch between the windows of the different devices. You may also need to rotate the image shown on a display – for instance, if you have installed a projector upside down on the ceiling or want to use a monitor in portrait mode. To solve all of these problems, you can use the `xrandr` tool.

With `xrandr`, you can configure the X Window System's Resize and Rotate (RandR) extension to adjust the main window, which the X clients use for display purposes, to suit your needs. Using `xrandr --listmonitors` gives you an overview of all the monitors connected to the system and their current configurations:

```
xrandr --listmonitors
```

```
Monitors: 3
0: +*eDP-1 1920/309x1080/174+3000+0 eDP-1
1: +DP-2-2 1080/510x1920/287+0+0 DP-2-2
2: +DP-2-3 1920/598x1080/336+1080+0 DP-2-3
```

You can use `xrandr -q` to see which modes the individual devices support. You can then set the desired mode for a device as follows:

```
xrandr --output DP-2-2 --left-of eDP-1 2
--mode 1920x1080
```

This command ensures that the monitor DP-2-2, which is connected to the computer's DisplayPort, uses a resolution of 1920x1080 pixels and says that the device is located to the left of the laptop's internal display (eDP-1). You can easily move the mouse to the left to switch from the internal display to the external monitor. If you operate the monitor in portrait mode, simply extend the command to include the `--rotate left` or `--rotate right` option as follows:

```
xrandr --output DP-2-2 --rotate left 2
--left-of eDP-1 --mode 1920x1080
```

The following command is useful if you use a projector connected to an

HDMI interface and you want to display an inverted image:

```
xrandr --output HDMI-1 2
--rotate inverted --mode 1920x1080
```

Setting up the Mouse and Touchpad

Another popular X Window System setting is the configuration for natural scrolling. Often on Linux, the default setting for the connected mouse and the internal touchpad behaves like a scroll bar, which differs from the behavior on a device with a touchscreen. In other words, scrolling up causes the screen to scroll up, and scrolling down causes it to scroll down. However, on a smartphone or tablet touchscreen, it is the other way around, which means that many users will want to adjust this setting on their laptops or desktops.

In the X Window System, the *libevent* library is responsible for processing a user's input events and reacting to them accordingly. The configuration for the individual input devices relies on the `xinput` tool. An overview of the available devices can be displayed using the `xinput list` command (Figure 1).

The natural scrolling setting relates to the mouse and touchpad. All the available configuration settings for these devices can be displayed with the help of

```
xinput list-props <ID>
```

where you replace the ID with the device name (Figure 2). This is also recommended, since the ID is not static


```
> xinput list
Virtual core pointer          id=2    [master pointer  (3)]
↳ Virtual core XTEST pointer  id=4    [slave pointer  (2)]
↳ Razer Razer Huntsman Tournament Edition id=11   [slave pointer  (2)]
↳ Razer Razer Huntsman Tournament Edition id=12   [slave pointer  (2)]
↳ MOSART Semi. 2.4G Wireless Mouse id=14   [slave pointer  (2)]
↳ Elan Touchpad id=17   [slave pointer  (2)]
↳ Elan TrackPoint id=18   [slave pointer  (2)]
Virtual core keyboard        id=3    [master keyboard (2)]
↳ Virtual core XTEST keyboard id=5    [slave keyboard (3)]
↳ Power Button id=6    [slave keyboard (3)]
↳ Video Bus id=7    [slave keyboard (3)]
↳ Sleep Button id=8    [slave keyboard (3)]
↳ Lenovo ThinkPad Thunderbolt 3 Dock USB Audio id=9    [slave keyboard (3)]
↳ Razer Razer Huntsman Tournament Edition id=10   [slave keyboard (3)]
↳ Razer Razer Huntsman Tournament Edition id=13   [slave keyboard (3)]
↳ Integrated Camera: Integrated C id=15   [slave keyboard (3)]
↳ Integrated Camera: Integrated I id=16   [slave keyboard (3)]
↳ Intel HID events id=19   [slave keyboard (3)]
↳ AT Translated Set 2 keyboard id=20   [slave keyboard (3)]
↳ ThinkPad Extra Buttons id=21   [slave keyboard (3)]
↳ Razer Razer Huntsman Tournament Edition id=22   [slave keyboard (3)]
↳ Yubico YubiKey OTP+FIDO+CCID id=23   [slave keyboard (3)]
```

Figure 1: With `xinput list`, you can see all available devices on your system.

and may well change. The following two commands give you the same results for both the mouse and the touchpad:

```
xinput list-props 14
xinput list-props 2
'MOSART Semi. 2.4G Wireless Mouse'
```

```
xinput list-props 17
xinput list-props 'Elan Touchpad'
```

To change the setting, you can go back to the ID as well as to the name of the respective setting:

```
xinput set-prop 14 326 1
xinput set-prop 2
'MOSART Semi. 2.4G Wireless Mouse' 2
'libinput Natural Scrolling Enabled' 1
```

```
xinput set-prop 17 326 1
xinput set-prop 'Elan Touchpad' 2
'libinput Natural Scrolling Enabled' 1
```

The `xinput` command displays all input devices.

Persistent Settings

Keep in mind that the settings you make with `xrandr` and `xinput` are not persistent. To fix this problem, you have a variety of options. In the simplest case, you can create a startup file in which you enter the respective commands and then ensure that this file is called automatically by your desktop or window manager. If you are looking for an approach that is independent of your desktop or window manager, you can also store the

Listing 1: `/etc/X11/xorg.conf.d/99-libinput.conf`

```
Section "InputClass"
    Identifier "MOSART Wireless Mouse"
    MatchProduct "MOSART Semi. 2.4G Wireless Mouse"
    Option "Natural Scrolling" "true"
EndSection
Section "InputClass"
    Identifier "Elan Touchpad"
    MatchProduct "Elan Touchpad"
    Option "Natural Scrolling" "true"
EndSection
```

settings in the configuration file for the X server. Listing 1 shows an example for the *libevent* system. Another option is to create a script with the respective commands for each new X session. The advantage here is that you can also use the `xrandr` and `xinput` tools (and others) in this script without having to worry about the special syntax of the configuration file for the X Window System (see Listing 2).

Conclusions

The `xrandr` and `xinput` command-line tools help you make extensive settings for input and output devices on your systems. To make these settings persistent, be sure to store your settings in an X server configuration file or create a script using these tools.

Listing 2: `/etc/X11/Xsession.d/99-libinput-xrandr.conf`

```
xrandr --output eDP-1 --right-of DP-2-2 --mode 1920x1080
xrandr --output DP-2-2 --rotate left --left-of eDP-1 --mode 1920x1080
xinput set-prop 'MOSART Semi. 2.4G Wireless Mouse' 'libinput Natural Scrolling Enabled' 1
xinput set-prop 'Elan Touchpad' 'libinput Natural Scrolling Enabled' 1
```

```
> xinput list-props 14
Device 'MOSART Semi. 2.4G Wireless Mouse':
Device Enabled (188): 1
Coordinate Transformation Matrix (190): 1.000000, 0.000000, 0.000000, 0.000000, 1.000000, 0.000000, 0.000000, 0.000000, 1.000000
libinput Natural Scrolling Enabled (326): 1
libinput Natural Scrolling Enabled Default (327): 0
libinput Scroll Methods Available (330): 0, 0, 1
libinput Scroll Method Enabled (331): 0, 0, 0
libinput Scroll Method Enabled Default (332): 0, 0, 0
libinput Button Scrolling Button (333): 2
libinput Button Scrolling Button Default (334): 2
libinput Button Scrolling Button Lock Enabled (335): 0
libinput Button Scrolling Button Lock Enabled Default (336): 0
libinput Middle Emulation Enabled (337): 0
libinput Middle Emulation Enabled Default (338): 0
libinput Accel Speed (339): 0.000000
libinput Accel Speed Default (340): 0.000000
libinput Accel Profiles Available (341): 1, 1
libinput Accel Profile Enabled (342): 1, 0
libinput Accel Profile Enabled Default (343): 1, 0
libinput Left Handed Enabled (344): 0
libinput Left Handed Enabled Default (345): 0
libinput Send Events Modes Available (311): 1, 0
libinput Send Events Mode Enabled (312): 0, 0
libinput Send Events Mode Enabled Default (313): 0, 0
Device Node (314): "/dev/input/event6"
Device Product ID (315): 1578, 16642
libinput Drag Lock Buttons (328): <no items>
libinput Horizontal Scroll Enabled (329): 1
```

Figure 2: A device's configuration settings.



macOS file recovery and security

Salvage

macOS on-board tools and third-party applications can help prevent the loss of files and make security and backup your first priority. By Thomas Joos

If data loss under macOS is the result of defective hard drives, you need to adopt a structured approach for file recovery. Since macOS 10.15, the operating system has separated the system from the data, wherein macOS creates one volume for the system and one for data. Both storage locations use the APFS filesystem. Users only have read permissions for the system volume, which does not impose any restrictions on the user when working. At the same time, however, this provides better protection for the system. However, you do need to take this into account when recovering data and be careful when using recovery programs. Additionally, access to files and folders by third-party apps is restricted (see

Security and Privacy

Before I go into detail about the individual methods for dealing with recovering data on macOS, note that in *System Preferences* / *Security & Privacy* under the *General* tab, the *Allow apps download from* option usually should have only the *App Store* item selected. However, you will not be able to use some of the tools presented here if this is the case. To install these programs, you can enable *App Store and identified developers* now, and then change the settings back when you're done.

the “Security and Privacy” box). The protected directories include *Documents*, *Desktop*, the *iCloud* drive, and *Downloads*. Furthermore, apps are not allowed to access external drives without permission. Another protection is that Gatekeeper in macOS checks whether apps are digitally signed and originate from the App Store. Starting in version 10.15, the system repeats this test regularly – not just when first launched.

If you use a data recovery program and read this hard drive, you could aggravate the problem. If data has been lost because of a defective hard drive, the first step is to stop using the system. If it is clear that the hard drive is causing the trouble but is still generally working, it can be useful to make a complete copy of the disk (e.g., with Clonezilla [1]) before starting data recovery measures. After booting the Mac, the system can create an image of the disk; only then does it make sense to try to recover the data with data recovery tools.

Restore Data with On-Board Tools

Like Windows, macOS has a recycle (Trash) bin from which users can

recover data. If synchronization with iCloud is activated in the settings on the Mac, then accidentally deleted data can be partially recovered from *icloud.com* on devices that are logged in with the same Apple account. These settings can be found in *System Preferences* | *Apple ID* | *iCloud* under the *Options* button (Figure 1). Various volumes such as USB sticks have their own Trash bins. Some skill is required here because you can recover data from hidden files in the terminal. In the Finder, you can use the *Cmd + Shift + .* (period) keyboard shortcut to show hidden files. The *Library* folder often still contains copies of deleted Microsoft Office documents. To open the Finder, press *Cmd + Shift + G* and enter the path (*~/Library*). Note that you have at least two Library folders: the system Library folder at the root level of your hard drive and the user Library folder under each user account on the system. The instructions here access the user folder. After that, search for the desired files in this folder. macOS also displays the Library folder when you hold down the Option key and click *Go* in the Finder menu. This is where you will find numerous files that can possibly help with file recovery.

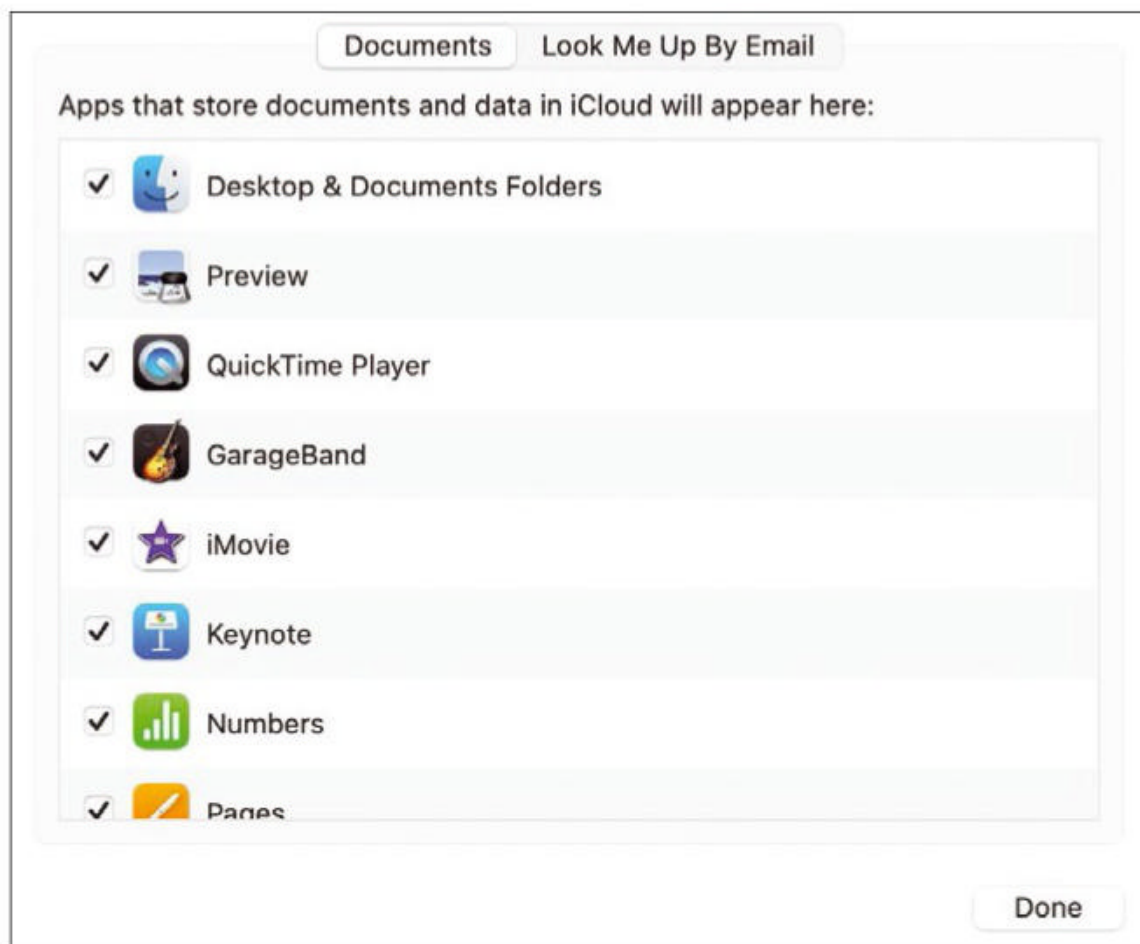


Figure 1: macOS can sync data to iCloud for recovery of lost files.

To search for hidden files in the terminal, you can configure macOS to show hidden files there:

```
defaults write com.apple.Finder \
  AppleShowAllFiles true
killall Finder
```

The `false` option lets you switch off the display again. On external data media, the Trash bin can be found as the `.Trashes` folder, where you might find further files that you can restore with the terminal.

Time Machine is available in macOS as a general data backup tool. The service can regularly and automatically back up entire volumes on Mac computers (Figure 2). Either external hard drives or network devices that support Time Machine serve as backup targets. Most network attached storage (NAS) devices can easily be linked to the backup software [2]. The TimeMachineEditor software [3] puts you in control of the various backups and schedules. The tool uses the basic Time Machine settings and does not need to run 24/7. Once you have adjusted the settings for Time Machine to your requirements, you can exit TimeMachineEditor.

Recovering Files with Tools

If the machine has neither a Time Machine backup nor an iCloud synchronization and the documents are also no longer available in the Trash, you have to rely on additional tools. One well-known example is Disk Drill [4]. However, the free version is limited and can only display files that can be recovered. For just under \$90 you can acquire the Pro version, which lets you recover data. An

alternative product is EaseUS Data Recovery Wizard for Mac [5], which can also search for recoverable files in the free version and then recover them with the commercial variant (~\$90/month or \$170 lifetime). For both tools to work properly, you need to allow access to the disks in *System Preferences* | *Security & Privacy*. Click on the *Privacy* tab and scroll down the left pane to the bottom, where you can choose *Full Disk Access*. You'll have to click the lock to make changes; then, you can click the Plus symbol under the right window and add the application. After that, the tools can scan your disks.

If these two tools do not help you, the professional Recoverit [6] is a good alternative. Here, too, you can test free of charge whether deleted files can be recovered. If this works, you can also recover Microsoft Office documents with the commercial version. The Free Mac Any Data Recovery software [7] offers a wizard to help you recover accidentally deleted files, even if they are no longer available in the Trash bin. After starting the tool, you can select which data you want to revive and where it should be placed. However, the tool requires some Mac experience, because you need to disable the System Integrity Protection feature in macOS. To do this, start the computer in the built-in recovery mode with the `Cmd + R`

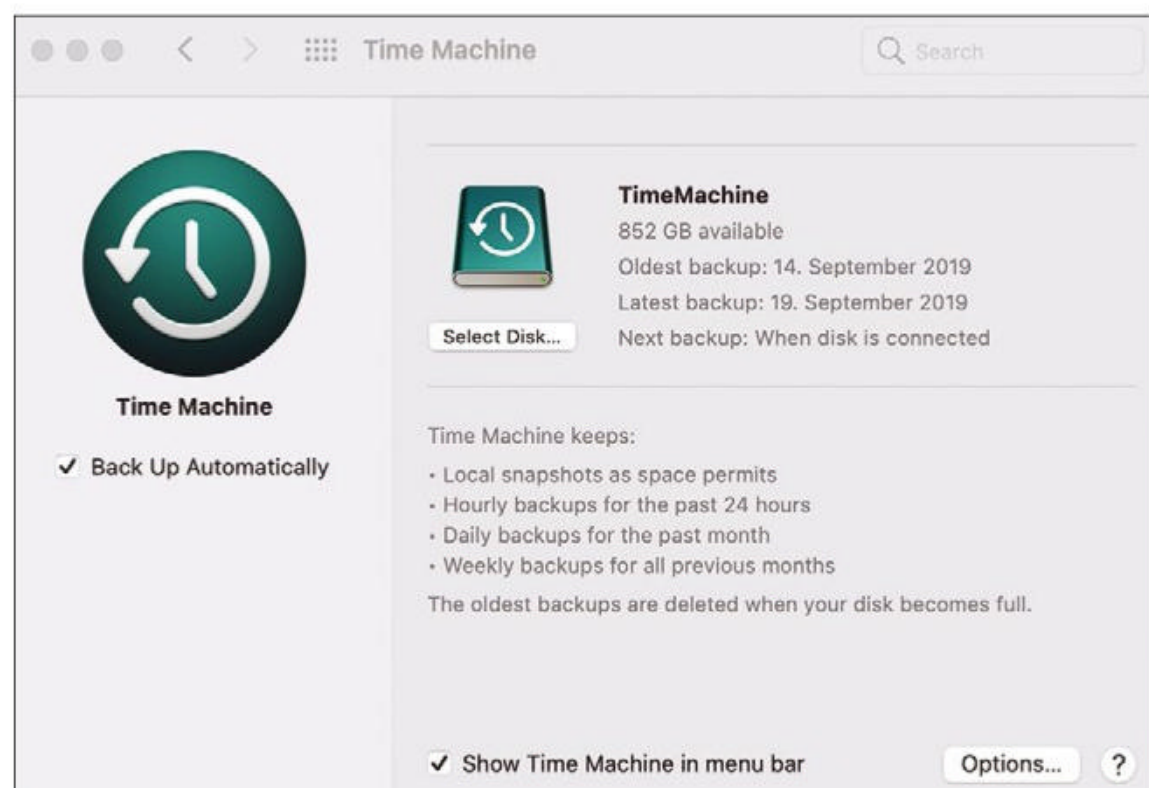


Figure 2: Time Machine is a good choice for use as a macOS data backup tool.

keyboard shortcut . In the recovery environment terminal, turn off the function and check the status with:

```
csrutil disable
csrutil status
```

After that, restart the Mac and you can use Free Mac Any Data Recovery (Figure 3). If the tool does not find data during the first scan, you can start an extended deep scan, which can take some time. The lower area shows the volume of data the system has found. Once your work is done, you will want to re-enable the System Integrity Protection feature with

```
csrutil enable
```

If the built-in recovery operating system does not start, you can also download the required program from the web by starting the Mac in Internet recovery mode with the Cmd + Alt + R keyboard shortcut. The last tool for data recovery I would like to introduce is PhotoRec [8]. It does not require any installation and starts from a USB stick. Working with the tool is a bit more complicated than with the others, but PhotoRec can often recover files where other programs fail. To begin, unpack the download archive and launch PhotoRec. macOS blocks the program run by default, because the tool does not come from a verified developer. Therefore, you need to approve execution in *System Preferences | Security & Privacy* under the *General* tab. PhotoRec needs to run with root privileges in macOS.

Double-clicking the file will open the Terminal and wait for you to enter your password; otherwise, you can run the program with sudo in the terminal. Once the tool is running, select the hard drive and the type of partition table from which you want to recover data.

Creating a Bootable USB Stick

A bootable USB stick is helpful to ensure that no data needs to be downloaded off the web when reinstalling macOS and the installation is performed locally. As a bonus, this tool is also good for data recovery, because it lets you recover data, even without a working Internet connection. The USB stick should have a capacity of at least 16GB. To start, delete the original data from the stick with Disk Utility. As the format, use *Mac OS Extended (Journaled)* and the scheme *GUID Partition Table*. Choose a name, (e.g., *USB*). You will need this name when you create the stick in Terminal:

```
sudo /Applications/
Install\ macOS\ Big\ Sur.app/
Contents/Resources/createinstallmedia
```

```
--volume /Volumes/<my Volume>/
--nointeraction
sudo /Applications/
Install\ macOS\ Big\ Sur.app/
Contents/Resources/createinstallmedia
--volume /Volumes/USB/
--nointeraction
```

If the macOS installation does not work properly during a restore, you can restart the Mac by holding down the power button for 10 seconds. At this point, it is important that you unplug all external devices from the Mac, if possible. Alternatively, reset the System Management Controller (SMC) [9], which especially helps with driver problems. Resetting the parameter RAM and the non-volatile RAM (NVRAM) can also help with problems [9].

Avoiding Data Loss with More Security

To prevent data loss on a Mac, especially if it is used in a home office or on the road, it is a good idea to look at the various firewall options. Although the operating system has a built-in firewall, attackers can work around it easily with additional tools. The first thing

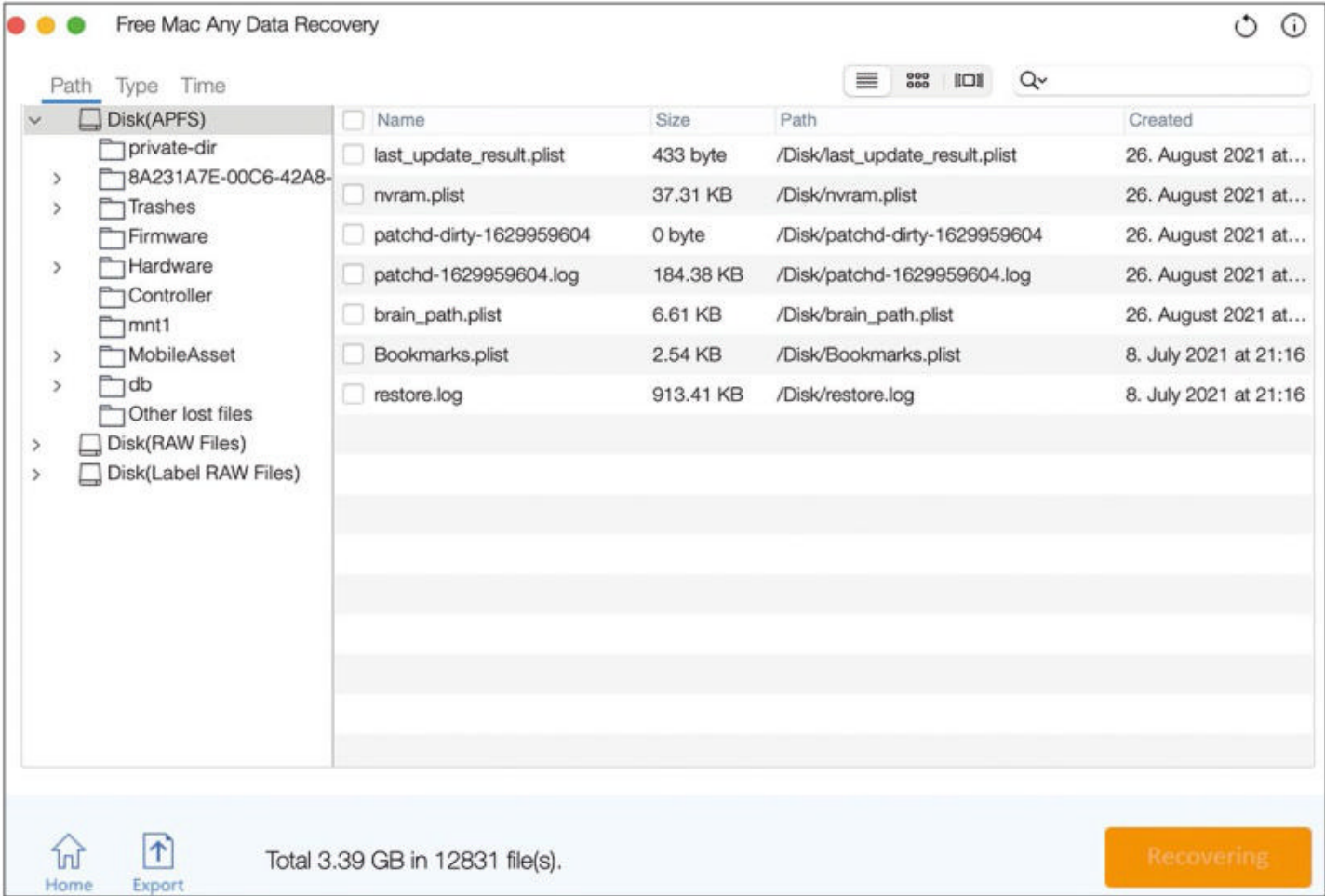


Figure 3: Data recovery with Free Mac Any Data Recovery is a tad more complicated than with comparable tools; however, it rewards the user with superior results.

you should do is check *System Preferences* | *Security & Privacy* | *Firewall* to see whether the firewall is active, which is not the case after installing macOS 11. To enable the firewall, click on the lock icon, enter your Apple ID, and start protection by clicking the *Turn On Firewall* button, which will make advanced security options available from the *Firewall Options* button. An external tool is necessary to keep a general overview of the programs that open Internet connections. Although the macOS built-in firewall blocks incoming connections, it does not give you control over outgoing traffic. The open source LuLu [10] firewall allows more control.

Clean Up and Optimize macOS

The free AppCleaner [11] software lets you remove apps and associated files that are no longer needed. The tool searches for these files stored on the system and deletes them. Documents you have created with the apps being interrogated are, of course, not affected by this procedure. The tool on a macOS 12 system reliably found all linked directories and folders in my tests. You do not have to perform the deletion process with the tool itself, but you can search for the files and directories in the Finder and delete the associated files and directories yourself. In this way, you can free up storage space and optimize the system at the same time. You can also avoid data loss with the tool, because it removes programs you no longer need in a controlled way.

After installing LuLu, you can specify which Apple applications you want the firewall to allow automatically. You can also automatically approve the internal macOS applications and block only external programs. The settings are in the *Preferences* option under the Shield in the menubar. For each program that tries to establish an Internet connection, a window appears that displays the details, where you can define whether the respective application is allowed to communicate externally. *Allow* permits the data traffic, whereas *Block* prevents it. If you want to allow only temporary access, then use the *temporarily* option. LuLu has a *rules* menu item that lets you call up the firewall rules to configure access or delete rules. If the application tries to access the Internet again, LuLu displays a window, and you can decide whether to allow or block.

Conclusions

To avoid data loss from the outset, it makes sense to start with system security and keep an eye on the firewall and outbound network traffic. If a mishap does happen, the tools and on-board resources presented here can help you recover data, which is easiest if a data backup with Time Machine has been implemented. See also the “Clean Up and Optimize macOS” box for a way to keep your filesystem orderly. ■

Info

- [1] Clonezilla: [\[https://clonezilla.org\]](https://clonezilla.org)
- [2] Restoring files with Time Machine and Spotlight: [\[https://support.apple.com/guide/mac-help/recover-files-time-machine-spotlight-mh15136/mac\]](https://support.apple.com/guide/mac-help/recover-files-time-machine-spotlight-mh15136/mac)
- [3] TimeMachineEditor: [\[https://tclementdev.com/timemachineeditor/\]](https://tclementdev.com/timemachineeditor/)
- [4] Disk Drill: [\[https://www.cleverfiles.com\]](https://www.cleverfiles.com)
- [5] EaseUS Data Recovery Wizard for Mac: [\[https://www.easeus.com/mac/mac-data-recovery/\]](https://www.easeus.com/mac/mac-data-recovery/)
- [6] Recoverit: [\[https://recoverit.wondershare.com/data-recovery-mac.html\]](https://recoverit.wondershare.com/data-recovery-mac.html)
- [7] Free Mac Any Data Recovery: [\[https://mac-data-recovery.com/free-mac-any-data-recovery.html\]](https://mac-data-recovery.com/free-mac-any-data-recovery.html)
- [8] PhotoRec: [\[https://www.cgsecurity.org\]](https://www.cgsecurity.org)
- [9] Reset SMC, PRAM, and NVRAM: [\[https://www.macworld.com/article/224955/how-to-reset-a-macs-nvram-pram-and-smc.html\]](https://www.macworld.com/article/224955/how-to-reset-a-macs-nvram-pram-and-smc.html)
- [10] LuLu: [\[https://objective-see.com/products/lulu.html\]](https://objective-see.com/products/lulu.html)
- [11] AppCleaner: [\[https://freemacsoft.net/appcleaner/\]](https://freemacsoft.net/appcleaner/)

The Author

Thomas Joos is a freelance IT consultant and has been working in IT for more than 20 years. In addition, he writes hands-on books and papers on Windows and other Microsoft topics. Online you can meet him on [\[http://thomasjoos.spaces.live.com\]](http://thomasjoos.spaces.live.com).



Stretching devices with limited resources

Fewer Memories

Compressed memory solutions for small memory problems. By Federico Lucifredi

Small embedded devices running Linux often face a trade-off between limited resources in CPU and RAM and the abysmal write performance of SD card storage with abundant capacity. The first rule, as I discussed back in 2018 [1] is to make exclusive use of high-quality SD cards for boot devices – a class 10 device or better for starters – with a reputable brand name printed on it. (I recommend SanDisk or Samsung SD cards.) Lower quality SD cards simply cannot satisfy the write pressure of system boot or other peak logging moments of operation, and while suitable for data, they cannot be used as boot devices. These are the basic table stakes, but this being Linux, you can of course optimize further!



Figure 1: ESPRESSObin v7 SBC shown in the 2GB of RAM and no embedded MultiMediaCard (eMMC) variety.

Test Bench

I am going to discuss a specific board for this column to try out the theories in practice. I will be using a Marvell ESPRESSObin [2] v7 (Figure 1), a low-cost, low-power board made by Globalscale Technologies. The ESPRESSObin was one of the earliest single-board computers (SBCs) to sport a 64-bit ARM CPU and remains remarkable today for its inclusion of a SATA port and PCI expansion. The full spec of the board is interesting in its own right (Table 1), but here I will focus on the setup for my tests. I use the external 12V power supply

provided and the USB interface for serial access.

On the client side, I employ the handy Serial [3] Mac program, which includes userspace implementations of driver stacks for the two common USB-to-serial chipsets, eliminating the hassle of installing or updating drivers on multiple computers (and the reboots that go with those Mach kernel modules). The connection runs at 115,200 baud, with no flow control and no parity.

The board bootstraps with U-Boot [4], and from there it loads the operating

Table 1: ESPRESSObin Technical Specs	
SOC	Marvell Armada 3700LP (88F3720) ARM Cortex A53 processor Dual core up to 1.2GHz
System memory	2GB DDR4 (1GB models also ship)
Storage	1x SATA interface 1x micro SD card slot Footprint for optional 4GB eMMC (not populated)
Network	1x Topaz networking switch 2x GbE LAN 1x Ethernet WAN 1x MiniPCIe slot
USB	1x USB 3.0 1x USB 2.0 1x micro-USB port
Expansion	2x 46-pin GPIO headers for I2C, GPIOs, PWM, UART, SPI, MMC
Misc	Reset button JTAG interface
Power supply	12V DC jack 5V via micro-USB port
Power consumption	Less than 1W thermal design power (TDP) at 1GHz

Lead Image © Lucy Baldwin, 123RF.com

```

USB-Serial Controller #2 — 80x24 — 115200.8.N.1
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 9 espressobin ttyMV0

espressobin login: root
Password:
Last login: Sat Feb  9 21:25:52 UTC 2019 on ttyMV0

Welcome to ARMBIAN 5.75 stable Debian GNU/Linux 9 (stretch) 4.19.20-mvebu64
System load:  1.84 0.46 0.15  Up time:    0 min
Memory usage:  4 % of 1998MB  IP:        10.0.1.23
Usage of /:    5% of 30G

root@espressobin:~#

```

Figure 2: Armbian bootup screen on ESPRESSObin v7 right after login.

system. Because the board was first released in 2017, the natural places to build current OS images are the Yocto project [5] and Armbian [6], a community-managed distribution that builds board-specific images of Ubuntu and Debian. I chose to use the stable Armbian build of Debian Stretch for the ESPRESSObin, and the results are shown in Figure 2.

The Right Trade-off

On systems with lots of storage and not enough RAM, one is naturally inclined to think about turning some of the permanent storage into swap space [7]. Unfortunately, the low write performance of SD card storage precludes this avenue to most embedded SBCs. The sensible strategy here is to turn some of the RAM into a write-friendly device and to stretch the amount of RAM allocated with the use of inline data compression for that partition.

I chose Armbian partly because it is a distribution that turns to this strategy by default. Figure 3 shows Armbian using a zram [8] device to store /var/log, offloading the primary source of boot-time writes from the physical device to a 50MB RAM drive. Zram, introduced by kernel 3.14 and considered stable, creates a block device in RAM and compresses writes to it on the fly. Default zram configurations on Debian and Ubuntu (controlled by the *zram-config* package) also create one swap device per CPU

core, as seen in Figure 3. The increase in CPU load is counterbalanced by the availability of more memory, as existing RAM is stretched with the use of compression.

Starting with kernel 4.14, zram can be configured as a write-back cache committing data to permanent storage in the background. Until then, this was commonly accomplished with zswap [9], a lightweight memory page compression driver that operates exclusively as a writeback cache for swap.

Swap Usage

Some important workloads, most notably Kubernetes, are openly averse to swap, preferring to address resource limits early rather than accept performance degradation and inconsistency. At the edge or in the Internet of Things (IOT), you have to work with the hardware you have and

```

USB-Serial Controller #2 — 80x24 — 115200.8.N.1
[ General system configuration (beta): armbian-config ]

root@espressobin:~# df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            1012240         0   1012240   0% /dev
tmpfs           204612      5540   199072   3% /run
/dev/mmcblk1p1 30423796 1230572 28854344   5% /
tmpfs           1023044         0   1023044   0% /dev/shm
tmpfs           5120         0     5120   0% /run/lock
tmpfs           1023044         0   1023044   0% /sys/fs/cgroup
tmpfs           1023044         4   1023040   1% /tmp
/dev/zram0       49584      4716   41284   11% /var/log
tmpfs           204608         0   204608   0% /run/user/0

root@espressobin:~# zramctl
NAME      ALGORITHM DISKSIZE  DATA  COMPR  TOTAL  STREAMS  MOUNTPOINT
/dev/zram0 lzo        50M     4.7M  747.1K  1.1M    2  /var/log
/dev/zram1 lzo        499.5M   4K     768    12K    2  [SWAP]
/dev/zram2 lzo        499.5M   4K     768    12K    2  [SWAP]

root@espressobin:~# swapon -s
Filename                                     Type      Size      Used      Priority
/dev/zram1                                  partition 511520    0         5
/dev/zram2                                  partition 511520    0         5

root@espressobin:~#

```

Figure 3: Three zram partitions were found in my system: two for Linux swap and one for logs.

compromises are more likely. Either way, the free [10] command provides a simple avenue to check swap use (Listing 1). Currently the system described is not using swap.

Setup

Zswap is easily configured by passing the parameter *zswap.enabled=1* to the kernel at boot in GRUB configuration

Listing 1: Checking Swap

```

root@espressobin:~# free
              total        used        free      shared  buff/cache   available
Mem:           2046088      82796      1834332         5548       128960       1891416
Swap:          1023040           0       1023040
root@espressobin:~#

```

Listing 2: Initializing Zram the Hard Way

```

# Aligning versions between kernel and modules referenced by virtual package
$ sudo apt update; sudo apt upgrade
$ sudo apt install -y linux-image-extra-virtual
# Create half a gigabyte ZRAM device at next available device file
$ zramctl --find --size=512M
/dev/zram0
$ zramctl
NAME      ALGORITHM DISKSIZE  DATA  COMPR  TOTAL  STREAMS  MOUNTPOINT
/dev/zram0 lzo-rle    512M    0B    0B    0B      2
# Make the ZRAM block device into a swap partition
$ sudo mkswap /dev/zram0
Setting up swapspace version 1, size = 512 MiB (536866816 bytes)
no label, UUID=0818f196-4e38-43be-88ad-de6b45f50ce5
# Turn the swap partition on
$ Swapon /dev/zram0
$ zramctl
NAME      ALGORITHM DISKSIZE  DATA  COMPR  TOTAL  STREAMS  MOUNTPOINT
/dev/zram0 lzo-rle    512M    4K    73B   12K      2  [SWAP]
# Cleanup procedure (destroy the ZRAM setup)
$ sudo swapoff /dev/zram0
$ sudo zramctl --reset /dev/zram0

```


options. Zram is more complicated, and not nearly as well documented as one would expect in most distributions. **Listing 2** shows how to set up two swap devices (and how to initialize and tear down the first one) with exclusively manual steps on Ubuntu Server 20.04 LTS. **Listing 3** makes use of the *zram-config* package found in Ubuntu (again, Server version 20.04 LTS) to automate part of the process

to mount a second device as a RAM drive for `/var/logs`.

Info

[1] “Assess USB performance while exploring storage caching” by Federico Lucifredi, *ADMIN*, issue 48, 2018, pg. 94, [\[https://www.admin-magazine.com/Archive/2018/48/Assess-USB-performance-while-exploring-storage-caching\]](https://www.admin-magazine.com/Archive/2018/48/Assess-USB-performance-while-exploring-storage-caching)

- [2] Marvell ESPRESSObin v7: [\[http://wiki.espressobin.net/tiki-index.php\]](http://wiki.espressobin.net/tiki-index.php)
- [3] Serial 2.12 for Macintosh: [\[https://www.decisivetactics.com/products/serial/\]](https://www.decisivetactics.com/products/serial/)
- [4] Das U-Boot – the Universal Boot Loader: [\[https://www.denx.de/wiki/U-Boot\]](https://www.denx.de/wiki/U-Boot)
- [5] Yocto Project: [\[https://www.yoctoproject.org\]](https://www.yoctoproject.org)
- [6] Armbian – Linux for ARM development boards: [\[https://www.armbian.com/espressobin/\]](https://www.armbian.com/espressobin/)
- [7] “Swap tricks” by Federico Lucifredi, *ADMIN*, issue 9, 2012, pg. 83
- [8] Kernel docs for zram: [\[https://www.kernel.org/doc/html/latest/admin-guide/blockdev/zram.html\]](https://www.kernel.org/doc/html/latest/admin-guide/blockdev/zram.html)
- [9] Kernel docs for zswap: [\[https://www.kernel.org/doc/html/latest/admin-guide/mm/zswap.html\]](https://www.kernel.org/doc/html/latest/admin-guide/mm/zswap.html)
- [10] free(1) man page: [\[https://linux.die.net/man/1/free\]](https://linux.die.net/man/1/free)

Listing 3: Initializing Zram the Easy Way

```
# install Ubuntu's ZRAM management package
$ sudo apt install zram-config
# start the service
$ sudo systemctl start zram-config
$ zramctl
NAME          ALGORITHM DISKSIZE DATA COMPR TOTAL STREAMS MOUNTPOINT
/dev/zram1 lzo-rle      495.8M   4K   73B   12K        2 [SWAP]
# Now let's make a ramdrive for /var/logs
$ zramctl --find --size=512M
/dev/zram0
# lay a filesystem on the ramdrive
$ sudo mkfs.ext2 -q /dev/zram0
# mount the ramdrive
$ sudo mount /dev/zram0 /var/log
$ zramctl
NAME          ALGORITHM DISKSIZE  DATA COMPR TOTAL STREAMS MOUNTPOINT
/dev/zram0 lzo-rle          512M  208K  3.1K   48K        2 /var/log
/dev/zram1 lzo-rle      495.8M    4K   73B   12K        2 [SWAP]
```

The Author

Federico Lucifredi (@0xf2) is the Product Management Director for Ceph Storage at Red Hat, formerly the Ubuntu Server Product Manager at Canonical, and the Linux “Systems Management Czar” at SUSE. He enjoys arcane hardware issues and shell-scripting mysteries and takes his McFlurry shaken, not stirred. You can read more from him in the O'Reilly title *AWS System Administration*.

ADMIN

Network & Security

NEWSSTAND

Order online:
bit.ly/ADMIN-Newsstand

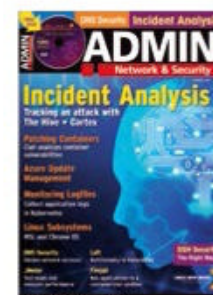
ADMIN is your source for technical solutions to real-world problems. Every issue is packed with practical articles on the topics you need, such as: security, cloud computing, DevOps, HPC, storage, and more! Explore our full catalog of back issues for specific topics or to complete your collection.

#66/November/December 2021

Incident Analysis

We look at updating, patching, and log monitoring container apps and explore The Hive + Cortex optimization.

On the DVD: Ubuntu 21.10 "Impish Indri" Server Edition



#65/September/October 2021

7 Email Clients

The features in this issue tackle digital certificates, email clients, and HP backup strategies.

On the DVD: Complete ADMIN Archive DVD



#64/July/August 2021

Bare Metal Deployment

Setting up, automating, and managing bare metal deployments gets easier with the tools presented in this issue.

On the DVD: Rocky Linux 8.4 (Minimal Install)



#63/May/June 2021

Automation

This issue we are all about automation and configuration with some tools to lighten your load.

On the DVD: Ubuntu 21.04 Server



#62/March/April 2021

Lean Web Servers

In this issue, we present a variety of solutions that resolve common web server needs.

On the DVD: Fedora 33



#61/January/February 2021

Secure Containers

Security is the watchword this issue, and we begin with eliminating container security concerns.

On the DVD: Clonezilla Live 2.7.0



WRITE FOR US

Admin: Network and Security is looking for good, practical articles on system administration topics. We love to hear from IT professionals who have discovered innovative tools or techniques for solving real-world problems.

Tell us about your favorite:

- interoperability solutions
- practical tools for cloud environments
- security problems and how you solved them
- ingenious custom scripts

- unheralded open source utilities
- Windows networking techniques that aren't explained (or aren't explained well) in the standard documentation.

We need concrete, fully developed solutions: installation steps, configuration files, examples – we are looking for a complete discussion, not just a “hot tip” that leaves the details to the reader.

If you have an idea for an article, send a 1-2 paragraph proposal describing your topic to: edit@admin-magazine.com.



Authors

Thomas Bär	30
Günter Baumgar	52
Jens-Christoph Brendel	16, 26
Chris Dock	34
Marco Föllmer	78
Rainer W. Gerling	43
Marc Grote	74
Ken Hess	3
Thomas Joos	70, 90
Jeff Layton	46
Rubén Llorente	64
Martin Loschwitz	10, 20, 80
Federico Lucifredi	94
Dr. Holger Reibold	40
Thorsten Scherf	88
Frank-Michael Schlede	30
Jens-Henrik Söldner	58
Jack Wallen	8
Matthias Wübbeling	62, 86

Contact Info

Editor in Chief

Joe Casad, jcasad@linuxnewmedia.com

Managing Editors

Rita L Sooby, rsooby@linuxnewmedia.com
Lori White, lwhite@linuxnewmedia.com

Senior Editor

Ken Hess

Localization & Translation

Ian Travis

News Editor

Jack Wallen

Copy Editors

Amy Pettie, Aubrey Vaughn

Layout

Dena Friesen, Lori White

Cover Design

Dena Friesen, Illustration based on graphics by vska, 123RF.com

Advertising

Brian Osborn, bosborn@linuxnewmedia.com
phone +49 8093 7679420

Publisher

Brian Osborn

Marketing Communications

Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
4840 Bob Billings Parkway, Ste 104
Lawrence, KS 66049 USA

Customer Service / Subscription

For USA and Canada:
Email: cs@linuxnewmedia.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)

For all other countries:
Email: subs@linuxnewmedia.com
www.admin-magazine.com

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the DVD provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks © 2022 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media unless otherwise stated in writing.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in Nuremberg, Germany by hofmann infocom GmbH.

Distributed by Seymour Distribution Ltd, United Kingdom

ADMIN (ISSN 2045-0702) is published bimonthly by Linux New Media USA, LLC, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA. January/February 2022. Periodicals Postage paid at Lawrence, KS. Ride-Along Enclosed. POSTMASTER: Please send address changes to ADMIN, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA.

Published in Europe by: Sparkhaus Media GmbH, Bialasstr. 1a, 85625 Glonn, Germany.

DrupalCon

PORTLAND2022

25-28 APRIL

Oregon Convention Center

events.drupal.org/portland2022

Join us for DrupalCon Portland,
where the people who make
amazing digital experiences
possible come together to
make them even better.





Business Desktop Replacement

TUXEDO InfinityBook S 17



Intel Core i7-11370H
Intel Iris Xe Graphics



17" display. 15" format.
93 % Screen-to-body ratio



Lift-up hinge
Enhanced ergonomics & cooling



73 Wh battery
up to 16 h runtime



100%
Linux

5

Year
Warranty



Lifetime
Support



Built in
Germany



German
Privacy



Local
Support

TUXEDO
COMPUTERS

 tuxedocomputers.com