TNE30019 Communications Research Report

Jonathan Frances
School of Computer Science
Swinburne University of Technology
Melbourne, Australia
103426340@student.swin.edu.au

I. Introduction

Cloud solutions have achieved a level of maturity that has seen them develop into the solution of choice for businesses looking to maximise resource utilisation for the lowest cost. Inclusive of any decision to deploy to the Cloud is the requirement to consider the advantages and disadvantages of a variety of interconnected system, in particular the operating system that will be chosen to manage the business's operations. While seemingly dispersed along the sidelines relative to the well-known and celebrated names of Microsoft's Windows and Mac OS, Unix has proven itself as a stable, secure, and affordable option. The suite of features and long track record of performance has seen Unix become a ubiquitous component of a variety of core services including network devices including routers and switches, large-scale email and file servers, and even critical infrastructure. This report will document several of the key advantages and disadvantages underlying the dynamics of Unix's success. With the conclusions provided in this report, business and users alike will be better equipped to discern whether Unix-based operating systems are the correct choice for their particular deployment in the expanding domain of Cloud technologies.

II. Advantages of Unix for the Cloud

Unix-based operating systems have become ubiquitous around the world for the advantages they have been able to provide. Compared with more recent systems such as Windows, Unix have had a long history that has given that a head-start on the competition. Unix pre-dates the Internet, Ethernet, local-area networks, and was spawned in the earliest days of computer development when it was purely an academic pursuit. During this time Unix developers collaborated to innovate on numerous early designs and ideas into fully formed instantiations that are now considered fundamental to the principles of security, availability, and scalability that is increasingly sought in modern Cloud-computing scenarios.

A. Stability

Unix is widely regarded as an extremely stable and reliable operating system. Fundamental to this reputation is the design of the Unix operating system overall. For example, separating the kernel from the user space and ensuring the kernel is kept focused on core capabilities including resource management and allocation were key early decisions that allowed Unix to avoid hazards such as memory leakage that would inevitably lead to a server requiring a system reboot [3]. In addition, the ability for developers and users to review code of the operating system down to the kernel has remedied a vast number of potential bugs and issues that existed within these systems as they continued operating throughout the world. Handling heavy workloads then has been a hallmark of Unix-based operating systems for a long time, marking them as being ideal for servers, critical infrastructure, and mission-critical applications. This confluence of factors has also allowed Unix to remain tightly wedded to its core principles without needing too change to keep up with changing demands or resolve serious underlying issues. Unix-based operating systems provide certainty with "a bedrock of unchanging basics — languages, system calls, and tool invocations — that one can actually keep using for years, even decades" [16].

B. Security

Unix-based operating systems gain their reputation for improved security over their counterparts (especially Windows) by virtue of the incorporation of modularity as a fundamental design feature of the operating system. Unix provides modularity insofar as every aspect of the operating system can be added, reconfigured, fixed, or removed at will. In systems that do not require certain functionality, the modules that control that functionality can be removed. This allows for any administrator of a Unix system the capacity to focus on securing only what is required to run the system, rather than needing to be concerned with developments across an entire host of potential areas of risk [21]. For reference, in Windows Server 2019 there are 103 services that require the RPC service to function, none of which can be disabled without seriously altering how the Windows OS operates [19]. As with many other operating systems Unix also incorporates features such as user permissions, access controls, file encryption, and virtualisation, with the added advantage that Unix typically was at the forefront or outright invented several of these features that we recognise as fundamental today. In the case of access controls, the division of the operating system into user and kernel space has significant security advantages, meaning that "processes running throughout the system aren't all necessarily available to users (depending on role privileges) and, likewise, the user processes aren't available to the system as a whole" [11].

C. Flexibility

Given the somewhat counter-intuitive advantage of modularity enabling security, Unix-based operating system also benefit from increased flexibility in what they can offer to system administrators seeking to utilise their capabilities. System configuration tools such as rc allow system administrators to efficiently allocate system resources, allowing for an administrator to completely disable functionality such as the GUI or graphics drivers for the display a visual feed when they are not required in a system. Such a system might instead only need to be accessed via remote access, allowing for faster boot times and the ability reallocate system resources away from graphics hardware and into greater CPU time for more important operations. Unix-based systems are also flexible by virtue of their ability to scale across multiple types of hardware. While the required drivers must be available on the machine, the incorporation of a ubiquitous programming language like C allows for system calls to remain accessible across hardware specifications, allowing for upgrades such as increased storage or computational resources with hardware produced by virtually all manufacturers. Unlike Windows and Mac OS which require specific, sometimes manufacturer specific hardware such as the M1 chip for the latest versions of Mac OS, "it would be easier to count the processor architectures that Linux doesn't run on than provide a list of all the architectures that it does work on" [4].

D. Compatibility

In addition to being compatible with a wide variety of hardware, UNIX-based operating systems provide compatibility across all layers of the OSI stack as a prerequisite for any fully featured configuration. From the earliest days Unix systems bucked the trend of siloed development whereby software was developed for use exclusively with the manufacturers hardware. Unix charted a more sensible trajectory by pursuing development of core functionality according to open standards, from networking protocols and security frameworks such as encryption. This alternate course ensured the greatest possible compatibility with programming languages, hardware, and other software including essential tools and libraries across devices from workstations to servers, handheld devices and network infrastructure. Paired with the stability of Unix-based systems, the development of networking protocols and server configurations at an international level (through organisations like IEEE) ensure Unix-based operating systems provide functionality that is thoroughly reviewed and designed for portability, longevity, efficiency, and security.

E. Community

As discussed in several of the previous sections the Unix community contributes significantly to ongoing development of Unix systems. Despite being mostly volunteers the community has contributed a trove of online resources, maintained active forums for Q and A, and provided up to date documentation for almost every aspect of the Unix environment. Because of the open nature of Unix, the Unix community also includes members of the international conferences that gather to develop new frameworks and standards for secure computing as well as developers, end users, and administrators alike.

F. Affordability

Unix-based operating systems and many of the most important and useful tools that run on them are typically available free of charge. Compared with complex propriety operating systems that require ongoing licenses like Microsoft Windows and the multitude of management systems build on top of them, Unix systems drastically reduce operating costs. In conjunction with the cost benefits from system stability and hardware compatibility the overall cost to run a server using Unix is significantly below that of other operating systems. While there are proprietary versions of Unix, or systems that have been developed from a Unix base such as Android or Mac OS, a significant section of the computing space including file and mail servers, databases, and network infrastructure can use an open source version of Unix for free.

III. Disadvantages of Unix for the Cloud

Though the advantages of Unix are many there are several significant disadvantages that have prevented Unix from advances further in achieving more widespread adoption in areas such as person computing and enterprise deployments for Fortune 500 corporations among many others.

A. Steep learning curve

The main drawback of Unix-based operating systems is the often intimidating and unapproachable user interface meaning "users need to learn a new set of commands and tools to interact with the system" [20] to confidently complete a task. Compared with the relative intuitive interface provided in Windows, Mac OS and other personal computer solutions where users "are more accustomed to controlling systems with mice and touch screens"[13], the Unix command line interface (CLI) provides very little guidance for new users on how to even begin to learn the commands necessary for even simple tasks such as editing or moving a file. As discussed earlier the customisability of Unix provides great flexibility and security however it can also pose an intimidation factor and increase mistakes due to misconfiguration. With the capacity for complete customisability comes the implication of needing to be aware of and manage every aspect of a networked deployment. In the case of sudo, administrators benefit from the principle of least privilege while also suffering from increased management requirements as "sudo requires you to manage every host separately, and quickly gets complex, time-consuming, and ultimately, untenable" [12].

B. Poor Enterprise offering

In modern Enterprise environments the requirements demanded by regulations and standard business practices include requirements of computing systems that are unfortunately not a standard feature of most Unix systems. Requirements

for reporting on compliance, data backup and retention, and options for recovery for instance, are either not typically provided as standard or are relatively tedious to create except in only the newest distributions [8]. For businesses operating in enterprise contexts that are looking to deploy a Unix-based system while still meeting these requirements, investment in proprietary solutions is the only available option. Such solutions are naturally not inexpensive and therefore present a costly hurdle to all except large-scale firms or businesses that cannot afford not to incorporate the functionality to manage compliance, audits, data protection, and recovery features in their respective organisations.

C. Poor support

While there is a substantial community online of knowledgeable developers and passionate end-users, the ability for those individuals to provide support to large businesses that are running a Unix-based system within their deployment is next to zero. While in a Windows deployment the license that a firm would be required to subscribe to provides some measure of official support that includes contact with Microsoft support staff and in some instances the ability to schedule an official technician, the same is not true of Unix. Without any top level organisation to organise technicians to provide support or indeed any formal licensing model to collect subscriptions to provide salaries to support staff there is very little ability for Unix deployments to include additional support. Instead, organisations that deploy Unix-based operating systems must rely on in-house staff or contracted third-party organisations to deploy, manage, and maintain their installations.

IV. Unix-based network tools

A brief analysis of several of the most useful network-related tools available within the Unix environment, including their benefits and the risks posed by their use that network administrators will need to be wary of is provided below.

A. hostname

- **Pros:** Provides a convenient way to view the name associated with the device's IP as configured by the DNS that is used to uniquely identify the device over a network [9].

- **Cons:** Services that rely on hostnames as the basis of access control lists can be exploited via malicious spoofing of hostnames to gain elevated privileges and unauthorised access to change configurations [7].

B. ping

- **Pros:** Provides a simple way of testing network reachability by sending an "ICMP Echo Request (ECHO_REQUEST) packet to a remote Unix host once per second. Each packet that is echoed back via an ICMP Echo Response packet is written to the standard output, including round-trip time" [10].

- **Cons:** If used improperly ping can be abused to launch extremely damaging Denial of Service attacks using the UDP protocols capacity for high bandwidth, high frequency traffic transmission [5].

C. finger

- **Pros:** Easily obtain information about users currently logged in on a remote machine or entire domain.

- **Cons:** Passes information such as user-IDs, login name, email address, last logged in time, directory, and active shell without any authentication required, extremely insecure as all of this information can form the basis for a brute-force or social engineering attack against users in networked environment [6].

D. netstat

- **Pros:** Provides a simple command-line utility for auditing incoming and outgoing UDP and TCP network connections, interface statistics, routing tables, and kernel route information [14]

- **Cons:** Netstat is limited by the overwhelming amount of information it can often provide as well as its ineffectiveness as a security tool due to limitations in detecting kernel-mode rootkits [17].

E. dig

- **Pros:** Displays the entire DNS response message in response to a formatted DNS name query including the "various sections (header, question, answer, authority, and additional) with resource records in those sections printed in master file format" [1].

- **Cons:** Not as commonly utilised as nslookup on Unix-based operating systems and may therefore be required to be built before usage. Dig also does not include an interactive mode and therefore requires exact specification of flags and arguments to function as desired.

F. nslookup

- **Pros:** Simple, uncomplicated response information for forward and reverse lookups of domain names from the configured DNS server. Also provides functionality to "make a DNS server operational and to diagnose DNS-related issues" [2].

- **Cons:** Can be manipulated to provide call-and-response functionality for a malicious exploit including to exfiltrate sensitive device information to an attack server [15]

G. traceroute

- **Pros:** Assists in providing more granular network reachability information by performing a ping command for each router between a target and its a destination and returning round-trip latency details.

- **Cons:** Due to the prevalance and ease of ping-based denial of service attacks many public and shared networks now block ping traffic or configure routers within their network to not respond to ping requests as a security precaution. Such configuration changes render traceroute effectively useless [18].

V. Conclusion

In sum, the reasons why Unix-based operating systems have come to be used to operate such a large portion of the technologies that are deployed around the world is due to the stability, security, flexibility, compatibility, supporting community, and affordability that is on offer in such distributions as FreeBSD and Linux. In Cloud deployments, Unix-based operating systems represent a good option where stability, modularity, and cost are priorities. Equally however, businesses or individuals thinking about deploying a Unix-based operating system must be aware of the technical difficulties posed by the Unix environment and be prepared to shoulder the burden of resolving technical difficulties or deploying data protection and other enterprise solutions without enterprise support.

References

[1] Paul Albitz and Cricket Liu. *Using dig*. URL: docstore.mik.ua/orelly/networking_2ndEd/dns/ch12_09.htm. (accessed: 07:09:2023).

[2] Ashley. *How to Use nslookup Command in Linux and Windows?* URL: https://operavps.com/docs/nslookup-command/. (accessed: 07:09:2023).

[3] Hossein Ashtari. *UNIX vs. Linux vs. Windows: 4 Key Comparisons*. URL: https://www.spiceworks.com/tech/tech-101/articles/unix-linux-windows-comparison/. (accessed: 07:09:2023).

[4] Oliver Bailey. *Linux and IoT Scalability*. URL: https://www.comptia.org/blog/linux-and-iot-scalability. (accessed: 07:09:2023).

[5] CloudFlare. *Ping (ICMP) flood DDoS attack*. URL: https://www.cloudflare.com/en-au/learning/ddos/ping-icmp-flood-ddos-attack/. (accessed: 07:09:2023).

[6] CQR. *Finger*. URL: https://cqr.company/wiki/protocols/finger/. (accessed: 07:09:2023).

[7] Simson Garfinkel and Gene Spafford. *17.3 Primary UNIX Network Services*. URL: https://docstore.mik.ua/orelly/networking/puis/ch17_03.htm#:~:text=Because%20many%20UNIX%20applications%20use,to%20break%20into%20your%20systems.. (accessed: 07:09:2023).

[8] Geeks For Geeks. *Difference between UNIX and Windows Operating System*. URL: https://www.geeksforgeeks.org/difference-between-unix-and-windows-operating-system/. (accessed: 07:09:2023).

[9] Geeks For Geeks. *hostname command in Linux with examples*. URL: https://www.geeksforgeeks.org/hostname-command-in-linux-with-examples/. (accessed: 07:09:2023).

[10] Vivek Gite. *UNIX and Linux ping Command Examples*. URL: https://www.cyberciti.biz/faq/unix-ping-command-examples/. (accessed: 07:09:2023).

[11] Red Hat. *Understanding Linux*. URL: https://www.redhat.com/en/topics/linux. (accessed: 07:09:2023).

[12] Karl Lankford. *Unix Linux Server Security: 10 Best Practices*. URL: https://www.beyondtrust.com/blog/entry/server-security-best-practices-for-unix-linux-systems. (accessed: 07:09:2023).

[13] Steven Melendez. *Advantages Disadvantages of the Unix Operating System*. URL: https://www.techwalla.com/articles/advantages-disadvantages-of-the-unix-operating-system. (accessed: 07:09:2023).

[14] Priya Pedamkar. *Linux netstat*. URL: https://www.educba.com/linux-netstat/. (accessed: 07:09:2023).

[15] PortSwigger. *What is OS command injection?* URL: https://portswigger.net/web-security/os-command-injection. (accessed: 07:09:2023).

[16] Eric Steven Raymond. *The Durability of Unix*. URL: http://www.catb.org/~esr/writings/taoup/html/ch01s02.html. (accessed: 07:09:2023).

[17] Chris Sanders. *Determining If You are Actively Being Compromised*. URL: https://techgenix.com/determining-you-actively-being-compromised/. (accessed: 07:09:2023).

[18] Steve Taylor and Jim Metzler. *Ping and Tracert: We lose, the hackers win*. URL: https://www.networkworld.com/article/2299718/ping-and-tracert--we-lose--the-hackers-win.html. (accessed: 07:09:2023).

[19] Core Technologies. *Essential Windows Services: Remote Procedure Call (RPC) / RpcSs*. URL: https://www.coretechnologies.com/blog/windows-services/rpcss/#:~:text=Indeed%2C%20if%20you%20examine%20the,RpcSs%20on%20Windows%20Server%202019!. (accessed: 07:09:2023).

[20] Tutorial and Example. *Advantage and Disadvantage of UNIX*. URL: https://www.tutorialandexample.com/advantage-and-disadvantage-of-unix. (accessed: 07:09:2023).

[21] Robert Yeckley. *UNIX Has Always Been More Secure Than Windows*. URL: https://www.ipswitch.com/blog/unix-has-always-been-more-secure-than-windows. (accessed: 07:09:2023).