Published on *Tux Machines* (http://www.tuxmachines.org)

Server: Managing GNU/Linux Servers and Cost of Micro-services Complexity

By *Roy Schestowitz*
Created *16/08/2019 - 10:50am*
Submitted by Roy Schestowitz on Friday 16th of August 2019 10:50:17 AM Filed under Server [1]

- **Keeping track of Linux users: When do they log in and for how long? [2]**

  > The Linux command line provides some excellent tools for determining how frequently users log in and how much time they spend on a system. Pulling information from the /var/log/wtmp file that maintains details on user logins can be time-consuming, but with a couple easy commands, you can extract a lot of useful information on user logins.

- **Daily user management tasks made easy for every Linux administrator [3]**

  > In this article, we will be going over some tasks that a Linux administrator may need to perform daily related to user management.

- **The cost of micro-services complexity [4]**

  > It has long been recognized by the security industry that complex systems are impossible to secure, and that pushing for simplicity helps increase trust by reducing assumptions and increasing our ability to audit. This is often captured under the acronym KISS, for "keep it stupid simple", a design principle popularized by the US Navy back in the 60s. For a long time, we thought the enemy were application monoliths that burden our infrastructure with years of unpatched vulnerabilities.
  >
  > So we split them up. We took them apart. We created micro-services where each function, each logical component, is its own individual service, designed, developed, operated and

monitored in complete isolation from the rest of the infrastructure. And we composed them ad vitam æternam. Want to send an email? Call the rest API of micro-service X. Want to run a batch job? Invoke lambda function Y. Want to update a database entry? Post it to A which sends an event to B consumed by C stored in D transformed by E and inserted by F. We all love micro-services architecture. It?s like watching dominoes fall down. When it works, it?s visceral. It?s when it doesn?t that things get interesting. After nearly a decade of operating them, let me share some downsides and caveats encountered in large-scale production environments.

[...]

And finally, there?s security. We sure love auditing micro-services, with their tiny codebases that are always neat and clean. We love reviewing their infrastructure too, with those dynamic security groups and clean dataflows and dedicated databases and IAM controlled permissions. There?s a lot of security benefits to micro-services, so we?ve been heavily advocating for them for several years now.

And then, one day, someone gets fed up with having to manage API keys for three dozen services in flat YAML files and suggests to use oauth for service-to-service authentication. Or perhaps Jean-Kevin drank the mTLS Kool-Aid at the FoolNix conference and made a PKI prototype on the flight back (side note: do you know how hard it is to securely run a PKI over 5 or 10 years? It?s hard). Or perhaps compliance mandates that every server, no matter how small, must run a security agent on them.

Server

Source URL: http://www.tuxmachines.org/node/127014

Links:
[1] http://www.tuxmachines.org/taxonomy/term/147
[2] https://www.networkworld.com/article/3431864/keeping-track-of-linux-users-when-do-they-log-in-and-for-how-long.html
[3] http://techgenix.com/linux-administrator/
[4] https://j.vehent.org/blog/index.php?post/2019/08/15/The-cost-of-micro-services-complexity