

Top 10 tenets of a System Administrator

1. If you're doing it more than once, automate.

Any task that is worth doing more than once is worth automating. That means you should keep your scripting skills up to date on any platform you have to work on. This will also reduce the chance for mistakes the next time you accomplish this task.

2. Documentation is a process.

Document everything. Don't wait until *after* the project is done to start documenting, do it during. It will be easier for you and more accurate if you don't have to remember things you've done a long time ago. Making documentation a daily part of your routine will lessen the chance you might forget.

3. Generalize as much as possible.

Follow the Unix KISS philosophy. Your scripts should be kept simple and do one task well. They should be made generic enough to be reusable as often as possible. Similarly, your documentation should assume a minimum of previous knowledge. Think of someone who is new to the job and needs to be shown how to do something from the ground up.

4. Stay organized.

You don't need to read a book about thought management to become more organized. Decide now on how where your scripts will be stored and ensure you always follow the same procedure. Documentation can take many forms, but often the simplest and oldest is best, such as a web portal running a wiki. You don't want to chase down your documentation across sticky notes, emails, text files and so on. Whether you use OneNote, Evernote, or any other solution, you should never have to Google for a solution twice.

5. Patch and monitor.

Patching is something that should be a part of your automation. Whether it's desktop systems, servers or software applications, every part of the infrastructure should be automated, and you should have a way to verify that this is happening. If you can't tell at a glance how well the environment you're responsible for is doing, improve your process.

6. Handle security in layers.

Security doesn't end at the firewall. Don't leave privileged account passwords in text files. Implementing a password vault is quick and will make a big impact, both in making sure credentials are kept secure, but also serve as part of your documentation. Segment your networks so privileged systems don't co-exist with regular ones. Find the weak points, sandbox your web apps so they don't put the host server at risk. Monitor your firewall rules and IDS/IPS to make sure no unwanted traffic goes through. Make sure your anti-virus software is up to date and educate your users on how to behave in a secure way.

7. Be prepared for the worst.

Stay optimistic, but plan for the worst. This means doing proper backups using the 3-2-1 system, having three copies of any important data in two formats, making sure you always keep one copy off-site. Test your restore process, document that process and have a recovery plan that makes sense for your environment. Think up scenarios from software bugs to online attacks, physical breaches, power failures, flooding and fire, and find the best solution for them. People make mistakes, your procedures should keep those mistakes isolated.

8. Keep learning.

Don't get set in your ways. Always strive to learn more, and keep a percentage of each year to learn new software, products, or get new certifications. Be ready to handle the next shiny thing or switch role at a moment's notice as your business evolves. Take advantage of the incredible amount of free resources from YouTube videos to the Microsoft Virtual Academy, recorded talks at USENIX, DefCon and more.

9. Don't change for change's sake.

Don't fall into the trap of wanting to change something just for change's sake. Hype is not a business case. That Perl app may be old, but if it fulfills its task, leave it be. Account for the inevitable delays, cost overruns and scope changes before undertaking any new project. Avoid feature creep and ask yourself if there's a simpler way to accomplish a goal before implementing an overly complex system.

10. Have fun.

Don't get burned out. Be respectful to your users and colleagues, but learn to say no. Think about what is most important to you, and how you will think back on these days in 10 years.