WeGet letters

by Michael W Lucas

Dear Most Honest Advice Columnist in Technology,

I've been told that I must write my employer's disaster recovery plan. Seriously, this whole exercise is bogus. We don't have the resources to do actual disaster recovery, nor will the cheapskates I work for pay for any preparedness. I gotta come up with something though. What's the quickest, easiest way to ditch this problem so I can get back to my real job?

—Falling Behind on Doom WADs

Dear Experienced Sysadmin,

The main problem with cynicism in IT professionals is that it is inadequate. We think that by becoming cynical we brace ourselves to cope with the worst this industry can do to us, but cynicism is not a Boolean. One is neither "cynical" or "naïve." Cynicism is an evil overlord's dungeon. The more you explore it, the deeper you learn you can go, but you can never quite delve the worst depths. Cynicism can always be deepened and sharpened.

Your objection is not rooted in needing a disaster recovery plan. Anyone who works with computers knows the innate treachery of all hardware and software. Your real problem is that you have been assigned work that will never be used. Your organization has not devoted any resources to disaster recovery. It's a checkbox on the Checkbox Compliance Chief's list. The company believes that they need to fill the checkbox, not actually plan for the disaster. Your question does nothing but illuminate your lamentable shortage of cynicism--when the solution is obvious:

Be the disaster you want to see in the world.

Bad things happen. Everybody knows this intellectually. People don't internalize how vulnerable they are until they experience a short sharp shock of ruin. By providing seamless and robust computing experiences, you are depriving your organization of necessary inoculations of panic and despair. People do not believe in the need for disaster recovery until they have experienced disasters.

In the days before Wi-Fi, I was responsible for internal technology at a consulting firm. One of my duties was "security." If you are pretending to run a secure environment, every piece of equipment should require some sort of authentication for configuration. We had many printers without passwords and a large contingent of people who did not want to bother with passwords on them. I could have spent months or years arguing about the importance of passwords, or I could accept consensus and wait. One day, the company owners had a whole bunch of visitors in the office for a critical meeting. Several were connected to the office network so they

could get Internet. Ten minutes into the presentation, some heinous prankster connected to an over–welcoming printer and swapped its IP address and default gateway. Moments later, I was notified that the entire company network was broken. Thanks to my vast expertise with a packet sniffer, I was able to identify and fix the problem within minutes. Sadly, I was unable to determine who performed that malicious prank at such an inconvenient moment, but the CTO mandated my preferred password policy thirty minutes later. My recommendation for setting up an isolated visitor network in the big conference room was also immediately accepted, which made implementing future disasters more difficult but not impossible. A sysadmin should always rise to a challenge, after all.

A disaster recovery policy need not be onerous. Look at your organization's functions. Which are critical, and which should they have stopped doing years ago? The most vital function, of course, is payroll. Being assigned the duty of disaster recovery planner gives you the power to be nosy. Dig into your organization to verify that no matter what, you will be paid. Of all the disaster recovery tasks, this is always the easiest to get cooperation on. Everybody is only there for the money, after all. Even folks who claim to be entirely "mission driven" become surly when the money doesn't show. Your payroll person has done their best, but I have never seen an accountant who truly understands technology's eagerness for perfidy. You will find problems. Write up your recommendations for this most vital of tasks. This will get you credibility, because you've demonstrated that you understand the organization's most important role in people's lives.

From there, expand. Present each part of your plan as you create it. If you get pushback, tell people "that's fine" and change your plan to read *in a disaster, this function will not be restored*. Hey, it's a plan. It's written down. It's even honest. What more could anyone ask? Sometime later--not too soon, you don't want people noticing any sort of pattern or trend--a small disaster might make them change their minds and you will already have a plan prepared.

No plan can be considered complete until it is tested.

No plan can be considered complete until it is tested. Ideally, test each part of your plan as you write it. There is no need for anyone to know all the flaws in the first draught of your plan. It would only worry them. Verify that you can restore backups on decommissioned servers over junk switches. Make sure that you have enough spinning rust in the scrap pile to hold the important databases as well as the stupid ones that the CEO insists remain active because of this one personally traumatizing incident fifteen years before. Once you've successfully tested everything, schedule a disaster recovery test that you tell people about. It will still have problems, of course, but nobody would believe it was a real test if it ran perfectly.

The nice thing about disaster recovery plans is that they are always used. Even if your organization's headquarters is not immolated by fire-breathing hippopotami before your next performance review, one day your fancy all–SSD storage array will implode into a naked singularity, and you'll be forced to retrieve the derelict spinning rust array from the scrapheap and try to make it stagger along under load. The DBAs won't be happy, but DBAs are never happy so ignore their whining. Network gear will fail unexpectedly, because even though you've configured everything to spew its errors to syslog, nobody reads the logs. Fortunately, HP still honors their famous lifetime warranty on their 10/100 switches, so if you make sure they still work beforehand you will have a ready-steady replacement.

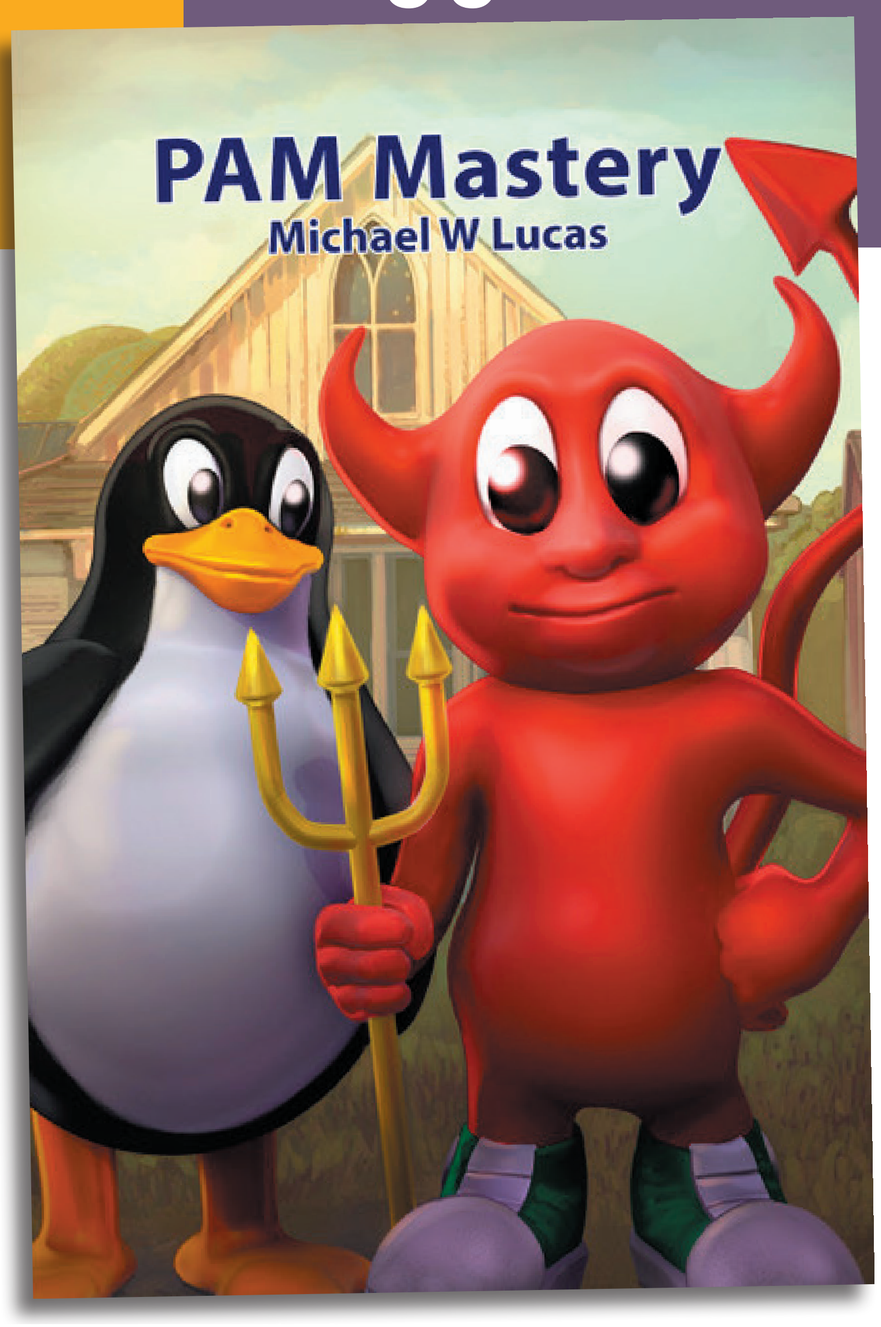
You can't do much about the greatest disaster of all, people. While you can list solutions in your plan, implementing any of them would violate the law. You might choose to carry out such plans anyway but do remember that documentation is considered evidence of premeditation.

Yes, DOOM is fun, but real disasters can be more fun. Disaster recovery plans do not have to be pointless wastes of your time. Hone your cynicism, seize control of the inevitable, and use disasters to improve your life.

Have a question for Michael?
Send it to letters@freebsdjournal.org

Many people consider **MICHAEL W LUCAS** an expert in disasters, but only because he is so often found at the center of them. He is the author of *Absolute FreeBSD*, the *FreeBSD Mastery* books, as well as books documenting the debacles of PAM, SNMP, TLS, DNSSEC, and more. Get a full list at https://mwl.io. His most recent book is *Letters to Ed(1)*, a collection of the first three years of this column. Spending your hard-earned money on any of them would be another disaster.

Pluggable Authentication Modules: Threat or Menace?



PAM is one of the most misunderstood parts of systems administration. Many sysadmins live with authentication problems rather than risk making them worse. PAM's very nature makes it unlike any other Unix access control system.

If you have PAM misery or PAM mysteries, you need PAM Mastery!

"Once again Michael W Lucas nailed it." — nixCraft

PAM Mastery **by Michael W Lucas**

https://mwl.io