foxtechnologies.

Puppet: The Good, Bad and Ugly for Configuration Management & Security

AUTHOR | MICHAEL MURRAY, DIRECTOR OF SALES ENGINEERING, FOXT

THE GOOD, THE BAD AND THE UGLY

I was at Red Hat Summit in Boston at the end of June. We had lots of activity at our exhibitor stand, and lots of discussions being passed on to me by our Sales Team

"Hey Mike, can you have a chat with Lucy about her infrastructure?"

This was my first time as a supplier SE/architect at the Summit, rather than a customer infrastructure manager, and I have to say it was weird having the same conversation again and again over three days. This seems to be the year people have finished bedding down Puppet in their server/VM infrastructure, and are looking for ways to fill gaps where Puppet isn't so useful. Like problems with OS Security.

Don't get me wrong, we think config management tools like Puppet are great. 80% of our worldwide installed base already use Puppet, Ansible, Chef, or CFEngine. And the remaining 20% are using "old school" tools like HSPA, Tivoli, or CA (though we think most of them will "rev-up" during the next tool selection cycle).

(Config management tools have also helped kill a nasty religious argument in the Infosecurity space about Agent or Agentless security software. If you need something to actually change on a server, you need something to accept the change… but I digress.)

So, back to Groundhog Day at the Summit. The conversation I was having with folks like "Lucy" would start something like this…

Lucy: "So we've been using Puppet for a while now, will probably upgrade to PE 3.8 by the Fall. My management has told me we need to fix a couple of security holes and production problems, and I'm kind-of stuck, wondered if you can help?"

As you might expect, I asked about her environment and how she was using Puppet to maintain security…

- How many server VMs? (thousands)
- Deploying sudoers files? (yes)
- Block access to root account (yes)
- Creating application functional accounts (yes)
- Block direct login to functional accounts? (most of the time, sometime we need break-glass access, and that is a pain)
- Creating/updating Linux groups (yes)
- Creating user accounts on the VM (usually no, users live in LDAP)
- SSH keys (it depends)
- Who wrote your OS security recipes? (a consultant, when we started two years ago)

THERE IS HOPE! ...AND A SOLUTION

1. Puppet is great for spinning up new VMs, blades and containers with static configurations that have short lifespans. However, if the server/VM is going to be around for any length of time using Puppet for Linux (and UNIX) OS security control does not scale very well. We seem to start having these discussions with organizations with 3000 plus servers or VMs. There are a lot of files spread around the OS to keep consistent. That's a lot of recipes to deploy. And God forbid you have multiple architectures and OSes, as you will need variant recipes for each platform and OS. With FoxT you can make IDM and security changes, and this operates cross-platform, and at scale, with a single command.

   Puppet is poor at UID/GID consistency. A strange statement for a configuration management product you might think, so I'll explain. Puppet was never designed to manage an identity management namespace and only interact with one, and as a result cannot enforce consistent UID/GID. If a deployed recipe says "make this change to this user on this server", the Puppet Agent will do so, even if it breaks UID/GID consistency across your domain. We see this pops up as an operation problem that gets worse over time, and more so as your server/VM estate expands.

2. SUDO configuration varies within your infrastructure and creeps over time. Sub-groups of servers may need their own configuration. New application versions often require changes, and all of a sudden you lose that "one policy" view of privileged command execution. Puppet is good at sudoers file snapshots with wide deployments and very low numbers of applications/databases. With growing app complexity and count in real life, we see administrators also making quick fixes on individual servers/VMs to fix immediate operational problems. As soon as that happens, you've lost both root control and auditability of your operations… not a good place to be. FoxT maintains a centralized SUDO like policy, maintained in one place, and deployed automatically to all or only one affected servers/VMs immediately.

3. Accessing root and functional accounts in the correct way with controlled SU or direct login in breakglass situations is not controlled by your config management tool at all. You need to invent (or buy) another tool to handle this. With FoxT's solution, you can be precisely control how, where and when this can be allowed, and with what two factor authenticator you may want to use.

4. Securely accessing user accounts on the OS is an operational headache. Typically you want to say "users live in LDAP." If you're using SSH for server/application support you have a sliiiight problem. Authorized_keys files are server-OS-local, not in directories. A user must have a home directory on the server. Your SSH keys have to be managed, distributed and stored for each user on the OS on each node where you might work. If you have multi-platforms (say Linux and Solaris), then keys are stored in different formats. With Puppet recipes for SSH key management are complex, multi-step, lead to failure, and do not scale. We handle SSH key distribution on an as-needed basis, serving keys to the live SSH session from our repository in real time, and you can concurrently auto-update Authorized_keys files in all formats as required.

5. OS and user security is a life cycle and a process. Config management is a series of snapshots, and doesn't take into account life cycle at all. They can live together, but not in the same tool IF you want to web-scale. With FoxT you can "play nice" with Puppet, which feeds inputs into OS, user security, and live access management decision processes without the headache of trying to create lifecycles into complex recipes that can be impossible to maintain.

Most FoxT customers are using Configuration management tools like Puppet for exactly that – software configuration management. Puppet's product documentation suggests it can also be used for manipulating user accounts and groups. But as your infrastructure becomes web-scale, this crumbles and starts to breaks down… especially as identity lifecycle, privileged access management, and security consistency becomes REALLY important. At that point, you need to consider a true identity and access management solution that works in partnership with config management. Admins have the ability to radically simplify their recipes and software management configuration and can leverage tools that are designed to provide the control required to maintain their organization's security policies and standards.

If you'd like to know more how Puppet and FoxT works together in partnership, we would be happy to show you how our other customers make savings in time, effort, and worry.

ABOUT FOXT

Fox Technologies, Inc. helps companies protect corporate information assets with network security and access management software as well as striving to simplify compliance and streamline administration with an award-winning access management and privileged account control solution. Our access management software centrally enforces granular access entitlements in real time across diverse server environments.

To learn more about Fox Technologies, please visit us at: **www.foxt.com**

CONTACT INFO

North America
3300 Eagle Run Drive NE, Suite 202
Grand Rapids, MI 49525
+1 877 818 3698 (Toll Free)

Sweden
FoxT Sweden AB
Kungsängsgatan 18A
SE-753 22, Uppsala
+46 18 16 00 00 (Main)

United Kingdom
400 Thames Valley Park
Reading , Berkshire RG6 1PT
+44 1189 637 681 (Main)

www.foxt.com | info@foxt.com

foxtechnologies.

You may also be interested in:
Best Practices for Unix/Linux Privileged Identity and Access Management