Managing accounts in a PCI environment

PCI Security Standards Council ™

In the PCI world we find that personalised Privileged Accounts are well understood. They fit in with the policy of knowing just who did what, where and when on your payment-related systems.

The generic, default or built-in system accounts pose more of a problem, and these problems are often ignored. One of the key issues is the number of accounts and who gets to use them. In any system there is a separation of concerns, so the system will have a 'root' Privileged Account, but the database will have its own credentials and there's often a developer account to access the code base. It's worth remembering that backup systems need to use highly Privileged Accounts simply because they need that access level to do the job.

Then there are those accounts created for the help desk. Rather than being personalised, they tend to be set up for specific operations and end up with names like 'helpdesk' or 'first_line_support'.

Software vendors have considered that they wish to make it difficult for a rogue administrator in the customer domain to be able to interfere with the database. They may also consider that the database is on a different system and hence cross-network authentication is required. Now the issues start. Administrator passwords are easy to change and will get changed regularly over the life of a PCI system's operation. Databases on the other hand tend to have persistent credentials, because

cont. overleaf

Osirium is a UK software development team that has pioneered the concept of a virtual air gap for privileged account access. The team have delivered a virtual appliance that can recognise an incoming identity, create a connection to a system, device or application, perform single sign-on and enterprise class password lifecycle management, and then hand the pre-prepared session back to the incoming request ready for system management. The session can be recorded, subject to time windows and device group separation. We have delivered millions of privileged tasks and sessions for many of our blue chip clients. Osirium currently has four patents pending.

they are more difficult to change.

To sum up the differences, Personalised Privileged Accounts are for day-to-day operation and Generic Accounts get used for maintenance. We need to consider how much work is subsequently outsourced to vendors or managed service providers.

Let's examine how to bring all these accounts under a control regime that would match PCI policies.

Firstly we need to separate the people from the credentials. People are notoriously lax with credentials, and particularly those that don't directly affect them. Contrast this with the way that people will protect their identity or the credentials to their bank accounts.

Then we need a scheme to map identities to roles. So, for example, if we know the identity of a developer in the vendors' organisation we can map this through to the generic developer account. If that mapping is made in Osirium's PxM Platform, each time it's used, we have complete traceability rather than just 'the developer account was accessed'.

We could add time windows and even incident tickets to this workflow. A vendor, developer or outsourcer would not be able to get a connection to the system without quoting a valid incident or change ticket. Now we have an audit trail with the 'why' filled in.

We could take this one level higher with session recording. Now we can report against particular tickets and see 'what' was done to the system in response. If we let the developers and SysAdmins see that they are being recorded, we get two benefits. Firstly, it's a great deterrent to the inside attacker and second, since the actions are on show, it pushes the quality up.

Most help-desk access revolves around specific tasks, for example resetting passwords or restarting queues. Since these are well-known command sequences these can be easily wrapped up into PxM Platform tasks. These tasks can be parameterised with the parameters value limited and sanitised. If you remove the need for any interactive login to a system, you've reduced the risk associated with that Privileged Account – no wandering, no prodding, no trying arbitrary commands.

By following these steps the possibility for external unauthorised access is virtually zero. Internal access is either through identity and ticket, or through the fine control of delegated tasks. With everything audited and cross-referenced you'll have met compliance and have all the metrics at hand to start improving your PCI workflow.