Colm MacCárthaigh **@colmmacc** *Wed Jun 01 17:57:56 +0000 2022*

A quick rage-thread about credentials. When security auditors just say things like "Critical credentials need to be rotated every 90 days" you need to fire them into the sun with urgency. Here's what you actually need ...

First rule of credential management: Rotation does nothing. It's revocation that matters. You always need a well-tested mechanism to make sure that you can remove or invalidate a credential that has been compromised.

Second rule of credential management: Have closed loops. Deactivated credentials are a common source of outages. When introducing a new credential you see it everywhere it needs to be before using it. When you remove one, you need to see it gone from use before deactivating.

Though you can't make that last part impossible to over-ride, because you do need to be able to lock out an attacker. Which brings up the next rule ...

Third rule of credential management: logging and detective controls are key. You need to be able to see when and where a credential is being used. This is important for operational safety and security. How would you even detect a stolen credential without this?

Fourth rule of credential management: be INCREDIBLY wary of time-based expiry. Use only when there is no other option (e.g. public SSL certificates). There's really no way to win with time-based expiry.

If your expiry time is something like a year, you don't get much security. Are you ok with an attacker using that cred for a year? So you still need revocation. If your expiry time is very short, like hours, are you *always* going to beat that renewal deadline? got good clocks?

Super short ephemeral credentials can be done, we do it at AWS, but it takes a *lot* of resources and diligence that most organizations don't have. Even we prefer to use real closed loops where we can.

Fifth rule about credentials: Store credentials only where they are needed. This seems obvious but is rarely done. In particular it's common to see "treasure trove" secret-distribution control-planes that know all of the credentials.

Distribution planes for secrets could use one or more of end-to-end, multi-party, or threshold encryption, so that those systems themselves don't know the secrets. We do this in places, but it's not a common pattern in industry that I've seen.

Sixth rule of credentials: if there is no reason to suspect credential disclosure or mis-use, leave it alone. Replacing credentials usually exposes them to more systems, at least temporarily. How do you know that's not more risky?

Seventh rule of credentials: asymmetric cryptography when you can, if not then choose between either memory-hard compute-hard hashing or derived-key symmetric auth depending on what fits your use-case. Avoid storing valuable secrets server side.

Eight rule of credentials: keep credentials inside one-way enclaves like TPMs, TEEs, HSMs, when you can. Best line of defense is to keep credentials inaccessible.

Ninth rule of credentials: If you can't write down a common password comparison side-channel from memory, do not implement your own authentication. Yes this is gatekeeping. Sorry, but no.

Tenth rule of credentials: Check for all-zeroes creds, and repeated values. You can do this with hashing, you don't need to record the secrets. Coding errors, failures of entropy systems, and erasure mistakes are common enough to make this check worth doing.

I'll stop there for now, maybe add more later. These are just some of the points I go through in reviews. Would love to hear from others about their own lessons and learnings. CYA-culture shallow audits drive my crazy, I hate to see customers trapped by them.