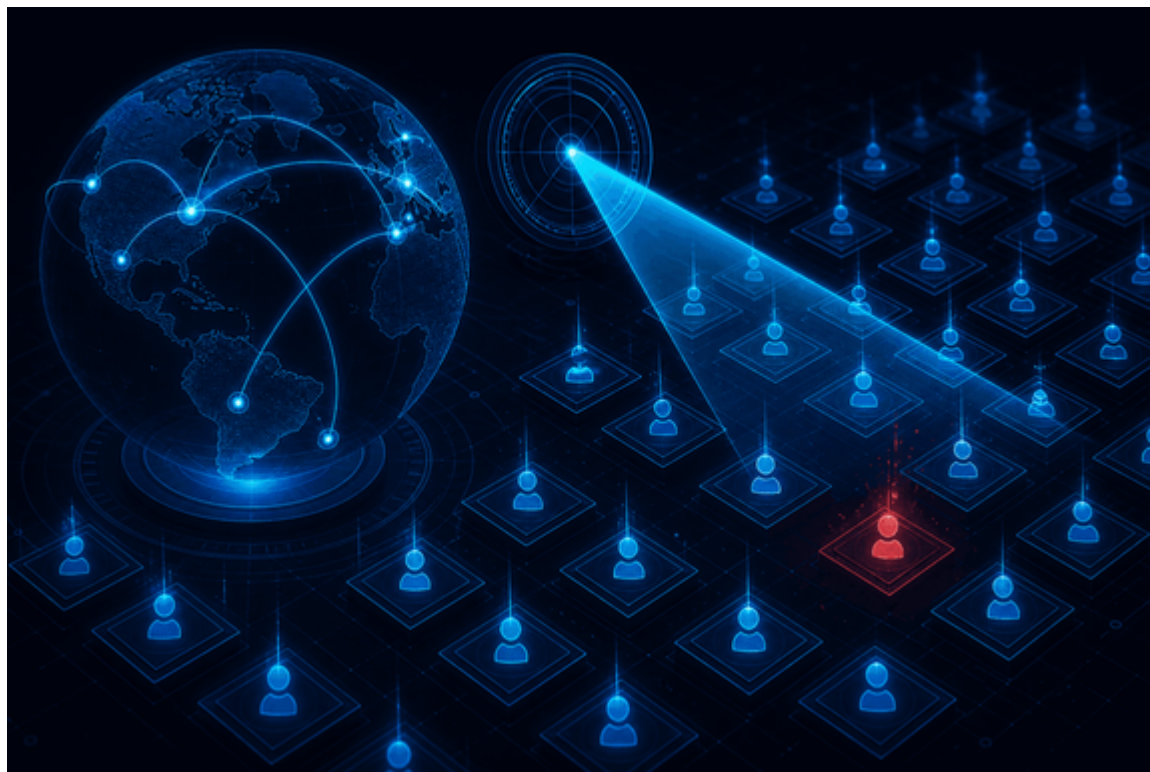


Catching Auth Attacks: Brute Force, Credential Stuffing, MFA Fatigue

Thu, Aug 6, 4:00 PM EDT · <https://scottslab.io/posts/detecting-auth-attacks-brute-force-credential-stuffing>

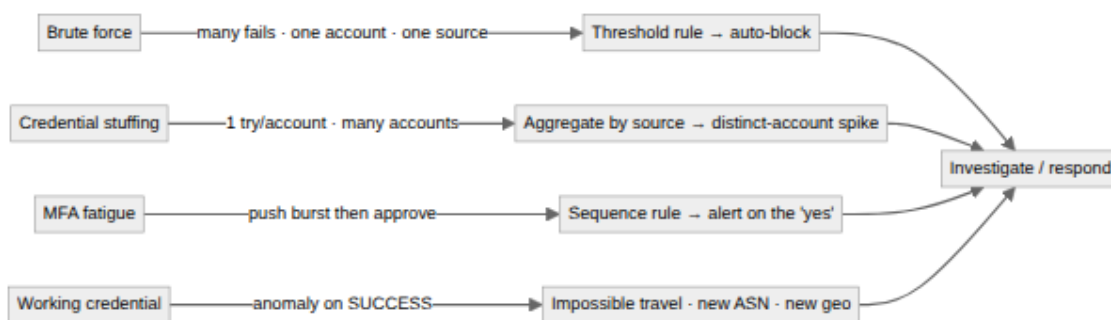


TL;DR

Brute force is loud, easy to catch, and mostly not what gets you; credential stuffing and MFA fatigue hide in low-and-slow attempts and, crucially, inside successful logins. The highest-signal detections watch successful auth: impossible travel, a new ASN, a burst of push prompts followed by a yes, not just failure counts. The attacker's whole goal is to look like a normal login, so that's where your detections have to be looking.

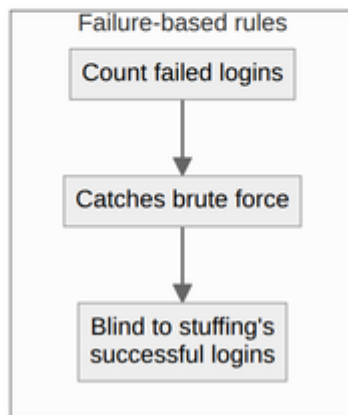
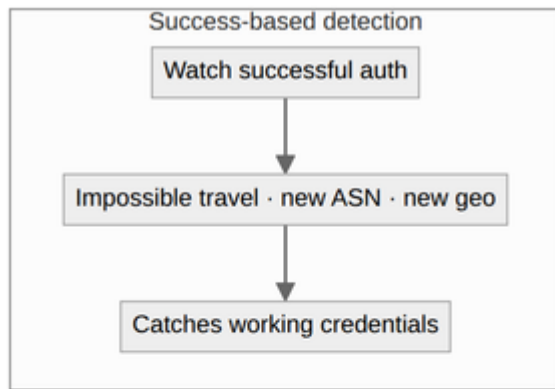
Good authentication decides who gets in. Detection decides whether you notice the people trying who shouldn't. The two failure modes look nothing alike in the logs, which is the part that trips up most home-grown alerting. If your only auth alert is "too many failed logins," you're catching the loudest, dumbest attack and missing the ones that actually work.

Brute force is the easy one, and the one everyone builds first. Many failures against one account, from one source, in a short window: it has a clean frequency signature, and a threshold rule catches it. I run exactly that in my SIEM and it auto-blocks the source. The problem is that serious attackers stopped doing this years ago precisely because it's trivial to detect. It's still worth alerting on, but if it's your whole program you're guarding the one door nobody serious uses.



Credential stuffing is the same tool turned sideways, and it's the one that quietly works. Attackers take username/password pairs from someone else's breach and try each pair *once* against you. From a single account's point of view there's one failed login. Nothing. The signal isn't per-account failures, it's the shape across accounts: one source, or a botnet spread across many, touching hundreds of different usernames with a low per-account failure rate and the occasional success. You detect it by aggregating the other way: distinct accounts per source, failures spread across the tenant, not by watching any single login. And the successes are what matter, because a stuffed password that works produces a *valid* login that no failure-based rule will ever flag.

That's why the highest-signal detections watch successful auth, not failed auth. Impossible travel, the same account authenticating from two places too far apart to be real, catches a working credential no matter how it was obtained. A first-seen login from a new country, a new ASN, a headless user-agent: none of these are failures, they're anomalies on success, and they're where takeovers actually surface. MFA fatigue lives here too. The tell is a burst of push challenges to one user followed by an approval, so the thing to alert on isn't a failed factor. It's many prompts then a yes. A rhythm, not an error.



The through-line is the same as the rest of my detection work: the loud attacks are easy and mostly harmless, and the quiet ones hide inside successful logins. So I spend the alerting budget accordingly: a cheap threshold rule for brute force, and the real attention on cross-account patterns and anomalies on the logins that *worked*. The attacker's entire goal is to become a normal successful login. Your detections have to be looking there when he arrives.