1. The attack started with a **social engineering** campaign on Uber employees, which yielded access to a VPN, in turn granting access to Uber's internal network \*.corp.uber.com.
2. Once on the network, the attacker found some PowerShell scripts, one of which contained [hardcoded credentials](https://www.gitguardian.com/glossary/remediate-sensitive-data-leaks-api-keys-hardcoded-source-code?ref=blog.gitguardian.com) for a domain admin account for Thycotic, Uber’s Privileged Access Management (PAM) solution.
3. Using admin access, the attacker was able to log in and take over multiple services and internal tools used at Uber: AWS, GCP, Google Drive, Slack workspace, SentinelOne, HackerOne admin console, Uber’s internal employee dashboards, and a few code repositories.

The critical vulnerability that granted the attacker such high levels of access was **hardcoded credentials in a PowerShell script**. These credentials gave admin access to a Privileged Access Management (PAM) system: [Thycotic](https://thycotic.com/?ref=blog.gitguardian.com). This tool carries huge amounts of privilege, making it a single point of failure; it stores both end-user credentials for employee access to internal services and third-party apps as well as DevOps secrets used in the context of software development. **This is a worst-case scenario**. The PAM system controls access to multiple systems, and having admin access means you can give yourself or extract secrets to all connected systems. This has appeared to give the attacker complete access to all of Uber's internal systems.