**Kerberoast**

How does it work?

When a domain account is configured to run a service in the environment, such as MS SQL, a Service Principal Name (SPN) is used in the domain to associate the service with a login account. When a user wishes to use that specific resource they receive a Kerberos ticket signed with the NTLM hash of the account that is running that service. The end result is that any valid domain user can request an SPN, and the ticket can bet taken offline and cracked.

How do I use it?

1. Have a valid domain account/password; it doesn’t matter if they’re the CEO or the janitor
2. Find GetUserSPNs.py; it’s installed on Kali by default
3. Run the following: python GetUserSPNs.py –request –dc-ip (IP of domain controller) domainname.local/validusername
4. Enter the password for said username
5. Take the Kerberos ticket(s) to hashcat (use mode 13100)
6. If you can crack it, you’re probably domain admin; use wmiexec/psexec/secretsdump.py or any other preferred method to create a new domain admin account(name it nGuard to avoid terrifying the customer)

Mitigation

Ensure all service accounts (user accounts with Service Principal Names) have long, complex passwords. Service Accounts with elevated AD permissions should be the focus on ensuring they have long, complex passwords. Ensure all Service Account passwords are changed regularly (at least once a year). If possible use [group managed service](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh831782(v=ws.11)) accounts which have random, complex passwords (>100 characters) and are managed automatically by Active Directory.

References

<https://www.blackhillsinfosec.com/a-toast-to-kerberoast/>

<https://adsecurity.org/?p=2293>

<https://github.com/nidem/kerberoast>