7-Password Spraying

Password spraying can result in gaining access to systems and potentially gaining a foothold on a target network. The attack involves attempting to log into an exposed service using one common password and a longer list of usernames or email addresses. The usernames and emails may have been gathered during the OSINT phase of the penetration test or our initial enumeration attempts.

ان انتا بتجرب باسورد معروفة مثلا علي كل username, email ,services

Scenario 1

In this first example, I performed all my standard checks and could not find anything useful like an SMB NULL session or LDAP anonymous bind that could allow me to retrieve a list of valid users. So I decided to use the `Kerbrute` tool to build a target username list by enumerating valid domain users (a technique we will cover later in this section). To create this list, I took the `jsmith.txt` username list from the statistically-likely-usernames GitHub repo and combined this with results that I got from scraping LinkedIn. With this combined list in hand, I enumerated valid users with `Kerbrute` and then used the same tool to password spray with the common password `Welcome1`. I got two hits with this password for very low privileged users, but this gave me enough access within the domain to run BloodHound and eventually identify attack paths that led to domain compromise.

Scenario 2

In the second assessment, I was faced with a similar setup, but enumerating valid domain users with common username lists, and results from LinkedIn did not yield any results. I turned to Google and searched for PDFs published by the organization. My search generated many results, and I confirmed in the document properties of 4 of them that the internal username structure was in the format of `F9L8`, randomly generated GUIDs using just capital letters and numbers (`A-Z and 0-9`). This information was published with the document in the `Author` field and shows the importance of scrubbing document metadata before posting anything online. From here, a short Bash script could be used to generate 16,679,616 possible username combinations.

```
#!/bin/bash

for x in {{A..Z},{0..9}}{{A..Z},{0..9}}{{A..Z},{0..9}}{{A..Z},{0..9}}
```

```
    do echo $x;
done
```

I then used the generated username list with `Kerbrute` to enumerate every single user account in the domain. This attempt to make it more difficult to enumerate usernames ended up with me being able to enumerate every single account in the domain because of the predictable GUID in use combined with the PDF metadata I could locate and greatly facilitated the attack. Typically, I can only identify 40-60% of valid accounts using a list such as `jsmith.txt`. In this example, I significantly increased my chances of a successful password spraying attack by starting the attack with ALL domain accounts in my target list. From here, I obtained valid passwords for a few accounts. Eventually, I was able to follow a complicated attack chain involving Resource-Based Constrained Delegation (RBCD) and the Shadow Credentials attack to ultimately gain control over the domain.

Password Spraying Considerations

هنا مثلا بيشرح ازاي password spray بيحصل مثلا فهو بدل ما يحرب اكثر من كلمة علي شخص واحد
فده ممكن يسبب rate limit وممكن يحصل انه هو يوقفه ويجرب بعد وقت معين فهو بدل ما يجرب كل
الكلمات هيجرب كلمة واحدة علي الاشخاص وبعد كده يجرب الكلمة التانية و هكذا

While password spraying is useful for a penetration tester or red teamer, careless use may cause considerable harm, such as locking out hundreds of production accounts. One example is brute-forcing attempts to identify the password for an account using a long list of passwords. In contrast, password spraying is a more measured attack, utilizing very common passwords across multiple industries. The below table visualizes a password spray.

Password Spray Visualization

Attack	Username	Password
1	bob.smith@inlanefreight.local	Welcome1
1	john.doe@inlanefreight.local	Welcome1
1	jane.doe@inlanefreight.local	Welcome1
DELAY		
2	bob.smith@inlanefreight.local	Passw0rd
2	john.doe@inlanefreight.local	Passw0rd
2	jane.doe@inlanefreight.local	Passw0rd
DELAY		
3	bob.smith@inlanefreight.local	Winter2022
3	john.doe@inlanefreight.local	Winter2022

Attack	Username	Password
3	jane.doe@inlanefreight.local	Winter2022

It involves sending fewer login requests per username and is less likely to lock out accounts than a brute force attack. However, password spraying still presents a risk of lockouts, so it is essential to introduce a delay between login attempts. Internal password spraying can be used to move laterally within a network, and the same considerations regarding account lockouts apply. However, it may be possible to obtain the domain password policy with internal access, significantly lowering this risk.

It’s common to find a password policy that allows five bad attempts before locking out the account, with a 30-minute auto-unlock threshold. Some organizations configure more extended account lockout thresholds, even requiring an administrator to unlock the accounts manually. If you don’t know the password policy, a good rule of thumb is to wait a few hours between attempts, which should be long enough for the account lockout threshold to reset. It is best to obtain the password policy before attempting the attack during an internal assessment, but this is not always possible. We can err on the side of caution and either choose to do just one targeted password spraying attempt using a weak/common password as a "hail mary" if all other options for a foothold or furthering access have been exhausted. Depending on the type of assessment, we can always ask the client to clarify the password policy. If we already have a foothold or were provided a user account as part of testing, we can enumerate the password policy in various ways. Let's practice this in the next section.