Trimarc Research: Detecting Password Spraying with Security Event Auditing

By Sean Metcalf

Published: 2017-02-10 · Archived: 2026-04-05 21:34:18 UTC

A common method attackers leverage as well as many penetration testers and Red Teamers is called "password spraying". Password spraying is interesting because it’s automated password guessing. This automated password guessing against all users typically avoids account lockout since the logon attempts with a specific password are performed against against every user and not one specific one which is what account lockout was designed to defeat. The attacker starts with a list list of passwords they're going to try which starts with the most likely passwords ("Fall2017", "Winter2018", etc).

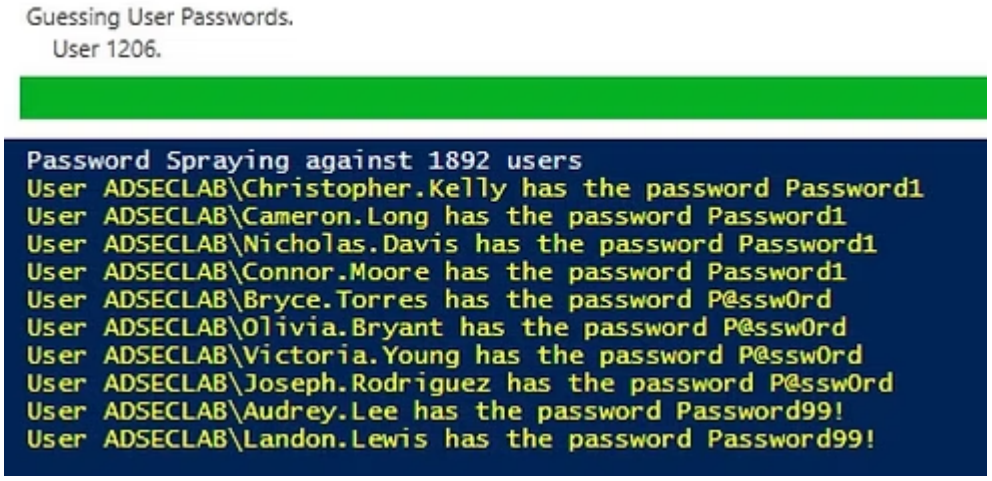
When password spraying begins, we start with the first password in the list. That first password is used in an attempt to authenticate as every user in Active Directory. This one password is attempted against each AD user and once all users have been tested with that password, we move on to the next one.

Since the Active Directory user lockout threshold is 5, we can try 4 different passwords for every user. Then we wait for >30 minutes (lockoutobservationwindow where the DCs keep the lockout count, after this it resets to 0), and try again. It's trivial to gather the information about the AD environment's password policy and have the password spraying tool automatically adjust to them.

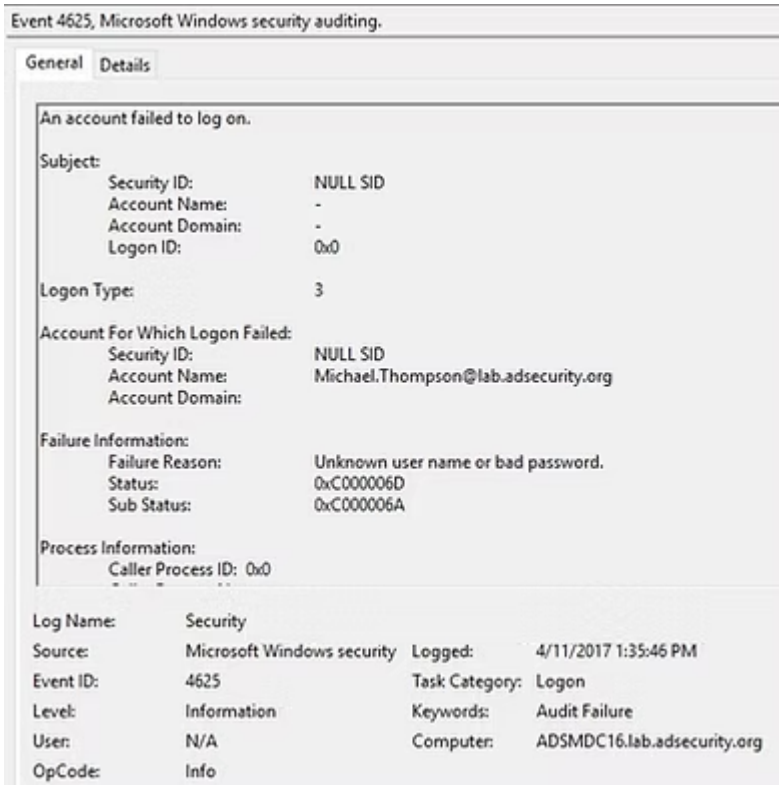
Graphic shows the Domain Password Policy for the lab domain environment using the AD PowerShell cmdlet Get-ADDefaultDomainPasswordPolicy cmdlet.

This works most of the time because users have bad passwords (especially if the password policy includes a password minimum of <10 characters). Often password spraying connects to an SMB share or a network service, so let’s start with connections to the PDC's netlogon share (\\PDC\NETLOGON) which is common on many networks. After password spraying has run for a while, we have discovered many user passwords, which may also include privileged accounts.

Graphic shows Password Spraying with a quick PowerShell script I wrote.



Password spraying against SMB on a Domain Controller results in event ID 4625 "logon failure" being logged on the DC and most organizations are logging that so when this happens, it should be detected.



Graphic shows event ID 4625 logged on the Domain Controller while password spraying.

However, many organizations haven't created correlation rules that state if *x* number of 4625 events occur within *y* time frame that password spraying is happening.

Graphic shows numerous 4625 event IDs logged in the lab domain environment while password spraying.

There is another way to discover password spraying in Active Directory. Every user account has an associated attribute named ""Bad-Password-Time" which is shown as "lastbadpasswordattempt " when using the Active Directory PowerShell cmdlet *Get-ADUser*. This attribute displays the date and time of the last bad password attempted for the account. Running the following PowerShell cmdlet shows the users in the AD domain with the attributes relating to bad password attempts.

```
get-aduser -filter * -prop lastbadpasswordattempt,badpwdcount | select name,lastbadpasswordattempt,b
```

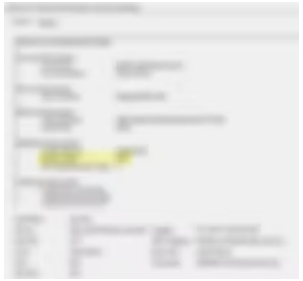
Graphic shows AD user accounts with the lastbadpasswordattempt & badpwdcount attributes in the lab domain environment after password spraying.

Looking at the results of the PowerShell command shown above, all of the bad password attempts are within the same minute and most are within seconds of each other. That’s unusual.

The attacker can avoid event ID 4625 from being logged by changing the service they connect to, so instead of connecting to SMB, we connect to the LDAP service on a Domain Controller. What happens? No more 4625 events are logged.

Graphic shows the lack of event ID 4625 when password spraying against LDAP.

A lot of organizations are monitoring for 4625 events, but if we connect to the LDAP service for password spraying, no 4625 events are logged. Kerberos logging needs to be enbled to log event ID 4771 and monitor for "Kerberos preauthentication failed". In the event id 4771 there's a failure code set to "0x18" which means bad password.



Graphic shows event ID 4771 which is logged when Kerberos logging is enabled on the Domain Controllers when password spraying against LDAP.

When password spraying on a domain-joined computer, event ID 4648 is logged ("a logon was attempted using explicit credentials") when the attacker is running password spraying on this system. There are numerous 4648 events showing that Joe User logged on and attempted to use the credentials for "Alexis Phillips" or "Christopher Kelley" or whomever and these are logged within seconds of each other. This type of activity is unusual.

The following four graphics shows event ID 4648 logged on the workstation where password spraying was performed. Audit logging must be enabled for this event ID to be logged.

Configuring Password Spraying Detection:

Password spraying happens in many AD environments and can be detected with the appropriate logging enabled and effective correlation. The primary methods for detection include:

Enable appropriate logging:

```
Domain Controllers: "Audit Logon" (Success & Failure) for event ID 4625.
```

```
Domain Controllers: "Audit Kerberos Authentication Service" (Success & Failure) for event ID 4771.
```

```
All systems: "Audit Logon" (Success & Failure) for event ID 4648.
```

```
Configure alerts for >50 4625 events within 1 minute.
```

```
Configure alerts for >50 4771 events with failure code=0x18 within 1 minute.
```

```
Configure alerts for >100 4648 events on workstations within 1 minute.
```

Write a PowerShell script that runs every day and reports on potential password spraying. The following command provides the required information.

```
get-aduser -filter * -prop lastbadpasswordattempt,badpwdcount | select name,lastbadpasswordattempt,b
```

Each of these alerting rules need to be tuned for your environment by increasing the number of alerts threshold and/or reducing the timeline.

Trimarc provides leading expertise in security solutions including security reviews, strategy, architecture, and implementation. Our methodology leverages our internal research and custom tooling which better discovers multiple security issues attackers could exploit to compromise the environment. Trimarc security services fit between traditional compliance/audit reviews and standard penetration testing/red teaming engagements, providing deep understanding of Microsoft technologies, typical security issues and misconfigurations, and provide recommendations based on our own best practices custom-tailored to balance operational and security challenges.
