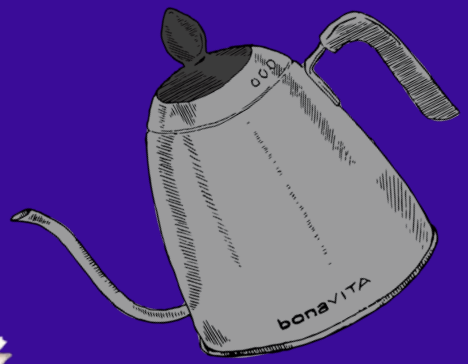

MaLDAPtive — pwsh
DEFCON
ENGAGE
(
|
(MaLDAPtive:¯_(LDAP)_/¯=ObFUsc8t10n)
(De-Obfuscation &:=De*te)(!c=tion)
)
Sabajete Elezaj (Sabi) & Daniel Bohannon (DBO)

PERMISO







aka
"DBO"



DANIEL BOHANNON

PRINCIPAL THREAT RESEARCHER

USA

MANDIANT (5 yrs)

Microsoft (2 yrs)

PERMISO (2 yrs)

@danielhbohannon

danielhbohannon

danielbohannon/Invoke-Obfuscation
/Invoke-CradleCrafter
/Invoke-DOSfuscation
/Revoke-Obfuscation
Permiso-io-tools/CloudConsoleCartographer

Solaris







aka
"SABI"

SABAJETE ELEZAJ

SENIOR CYBER SECURITY ENGINEER

Albania

Government (1 yr)

Consulting (3 yrs)

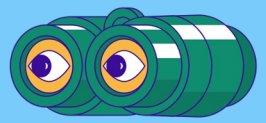
Engineering (3 yrs)

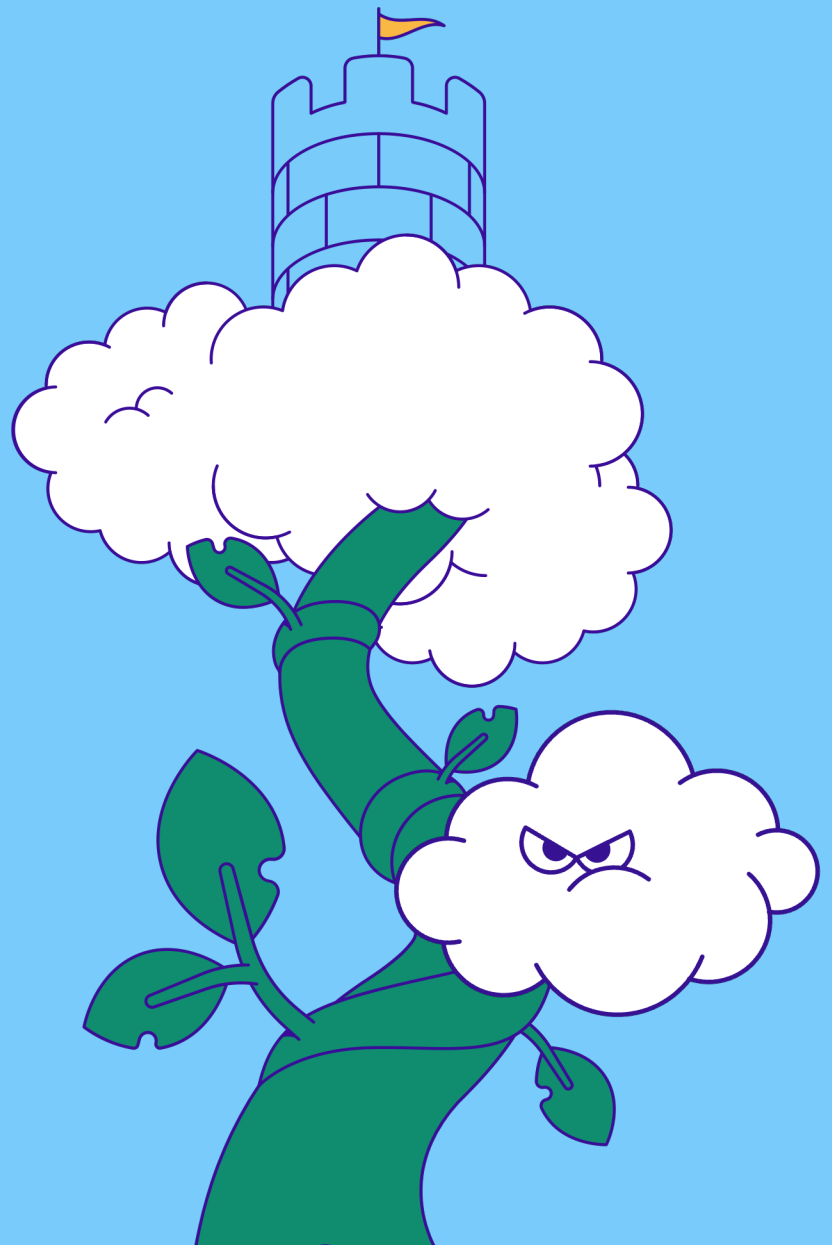
@sabi_elezi

sabajete-elezaj



AGENDA

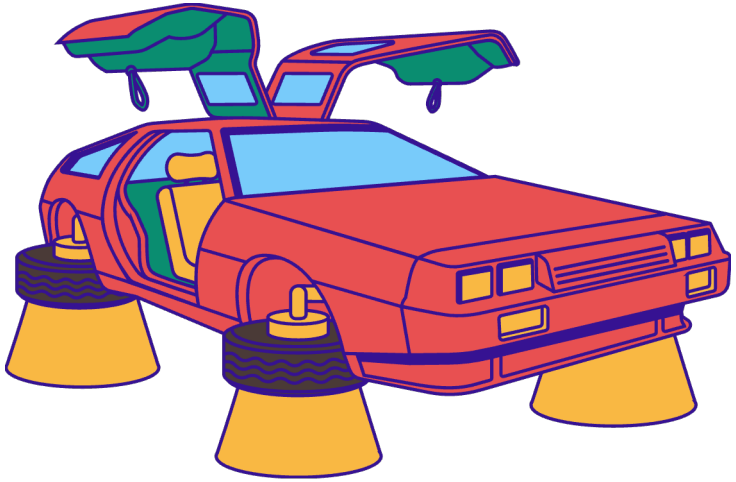
- Introduction 
- LDAP Overview
- PROBLEM: Obfuscating LDAP
- SOLUTION: Parse, Enrich, Detect
- MaLDAPtive Tool Demo + Release



Back to the History



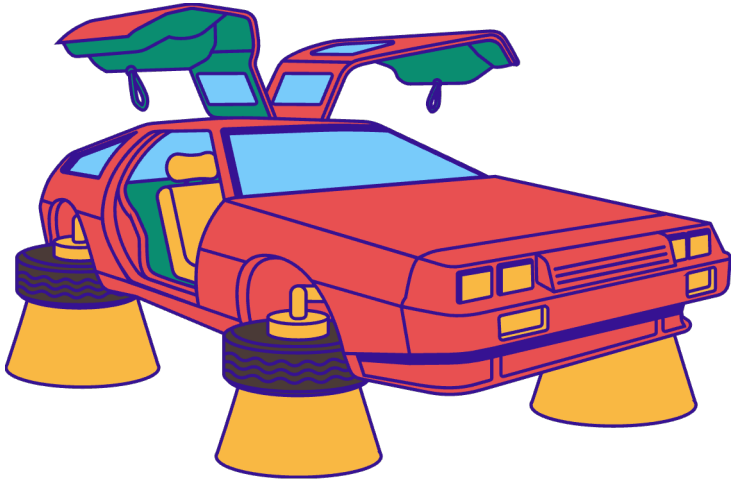
- 1980s
  - X.500 directory services
  - X.500 Directory Access Protocol (DAP)
- 1993–1997
  - “Lightweight” Directory Access Protocol (LDAP) v1–3
  - Used the simpler TCP/IP protocol stack
- 1998
  - OpenLDAP developed by OpenLDAP Project
- 2000
  - Microsoft released Active Directory (AD)
  - **Ensured compliance of AD with LDAP!**

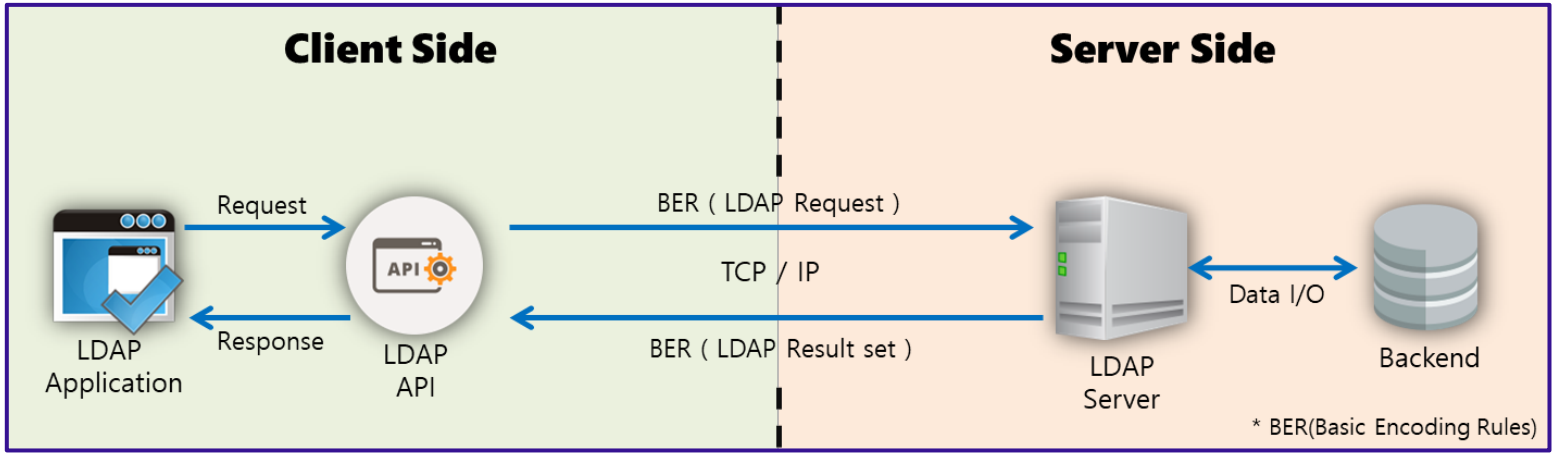


Back to the History



- 1980s
  - X.500 directory services
  - X.500 Directory Access Protocol (DAP)
- 1993–1997
  - “Lightweight” Directory Access Protocol (LDAP) v1–3
  - Used the simpler TCP/IP protocol stack





https://ldap.or.kr/wp-content/uploads/2017/07/%EA%B7%B8%EB%A6%BC3.png

Back to the Future

- Open-source tools for LDAP visibility (defensive & offensive usages)
  - 2015 – PowerView (@harmj0y)
  - 2016 – Bloodhound (SpecterOps)
  - 2017 – PingCastle (Vincent Le Toux)

BLOODHOUND

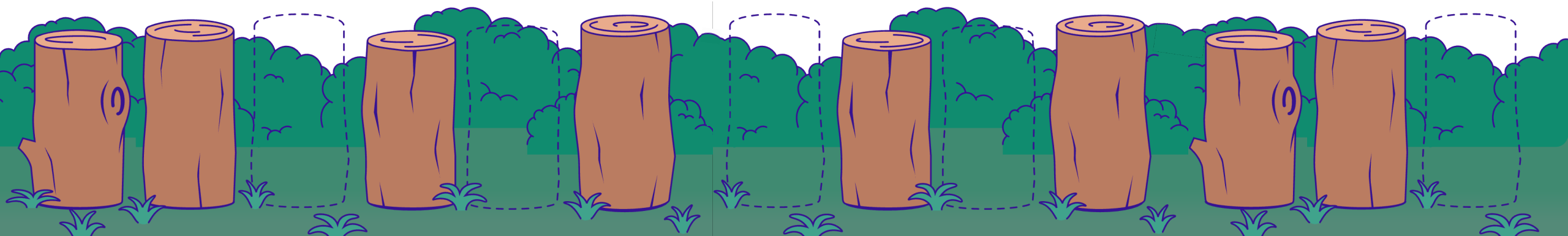
PING CASTLE



Back to the Logs?

- How to get LDAP logs **in a lab**?
  - SilkETW (Ruben Boonen, 2019)
  - LDAPMon (Johnny Johnson, 2023)
- How to get LDAP logs **in production**?
  - Defender for Endpoint (EDR agent)
  - Defender for Identity (sensor on DC)





Client-side vs Server-side LDAP Logs

- Client-side logs
  - WYSIWYG
  - #YOLO
  - Obfuscation ripe

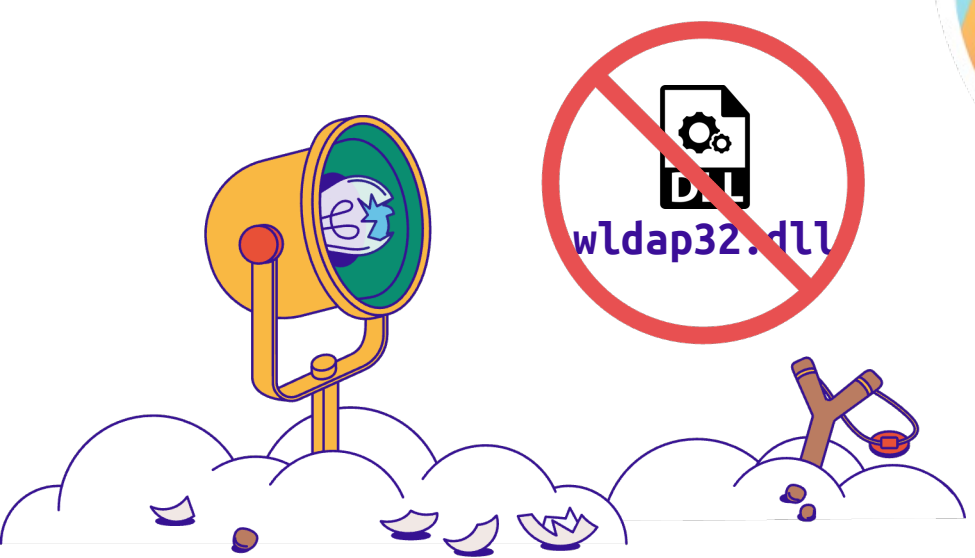


- Server-side logs
  - Significant normalization (but not 100%)
  - Undocumented expansions and substitutions of values



Client-side vs Server-side LDAP Logs

- Client-side logs
  - WYSIWYG
  - #YOLO
  - Obfuscation ripe
  - 100% log evasion opportunities





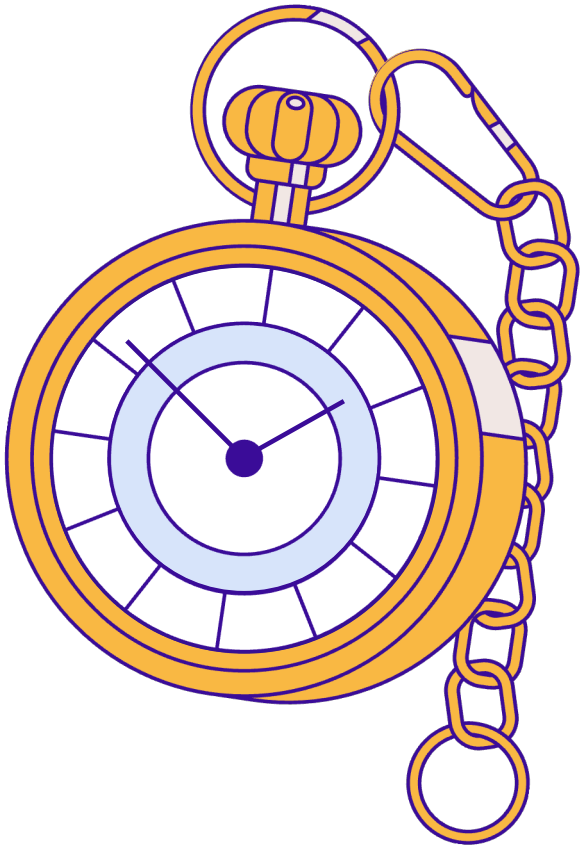
ADWS
(Active Directory Web Services)

- Server-side logs
  - Significant normalization (but not 100%)
  - Undocumented expansions and substitutions of values



What's Happening Over the Next 40 Minutes

1. Speed run of LDAP SearchRequest parsing, obfuscation, deobfuscation & detection
2. Completion of our **2,000-hour** R&D journey
3. Demo & release our brand new open-source framework: **MaLDAPtive**



Merriam-Webster

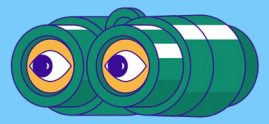
maladaptive adjective

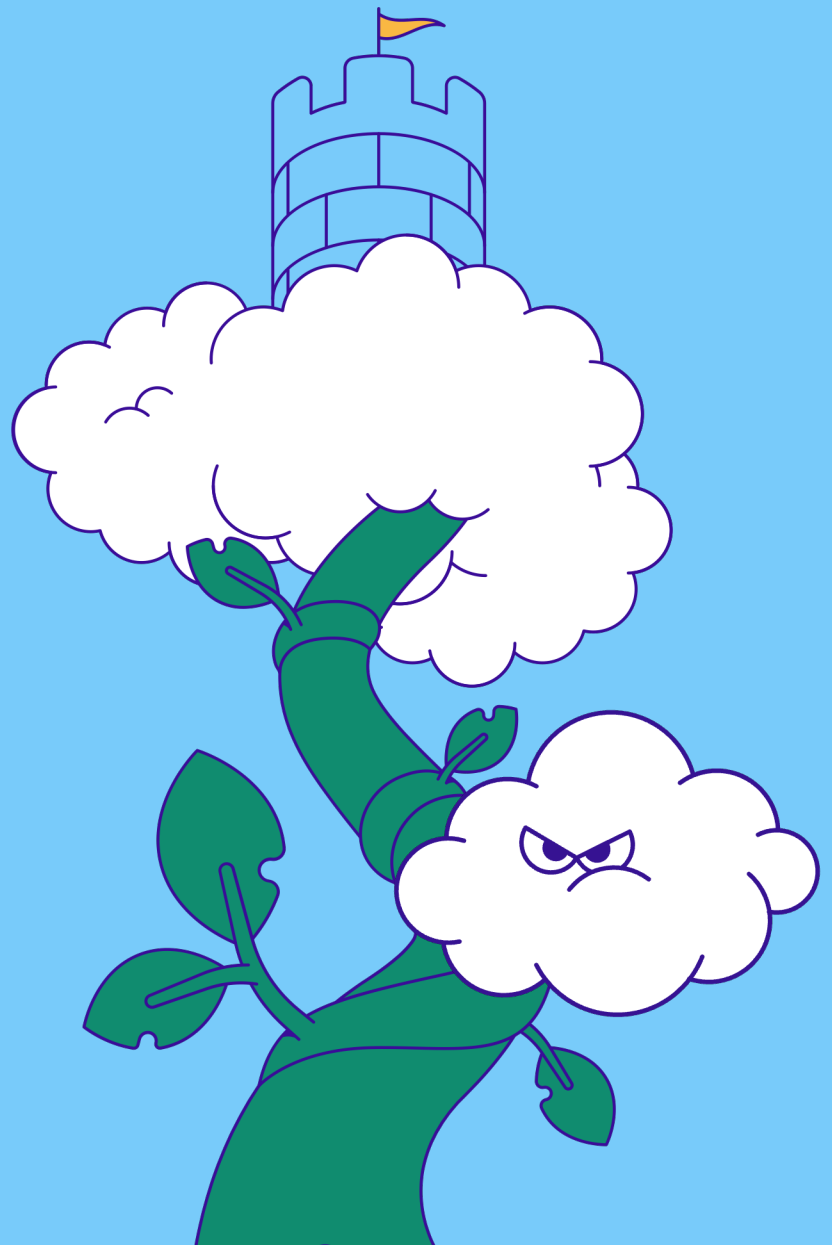
mal·adap·tive ˌma-lə-ˈdap-tiv

1 : marked by poor or inadequate adaptation

2 : not conducive to adaptation

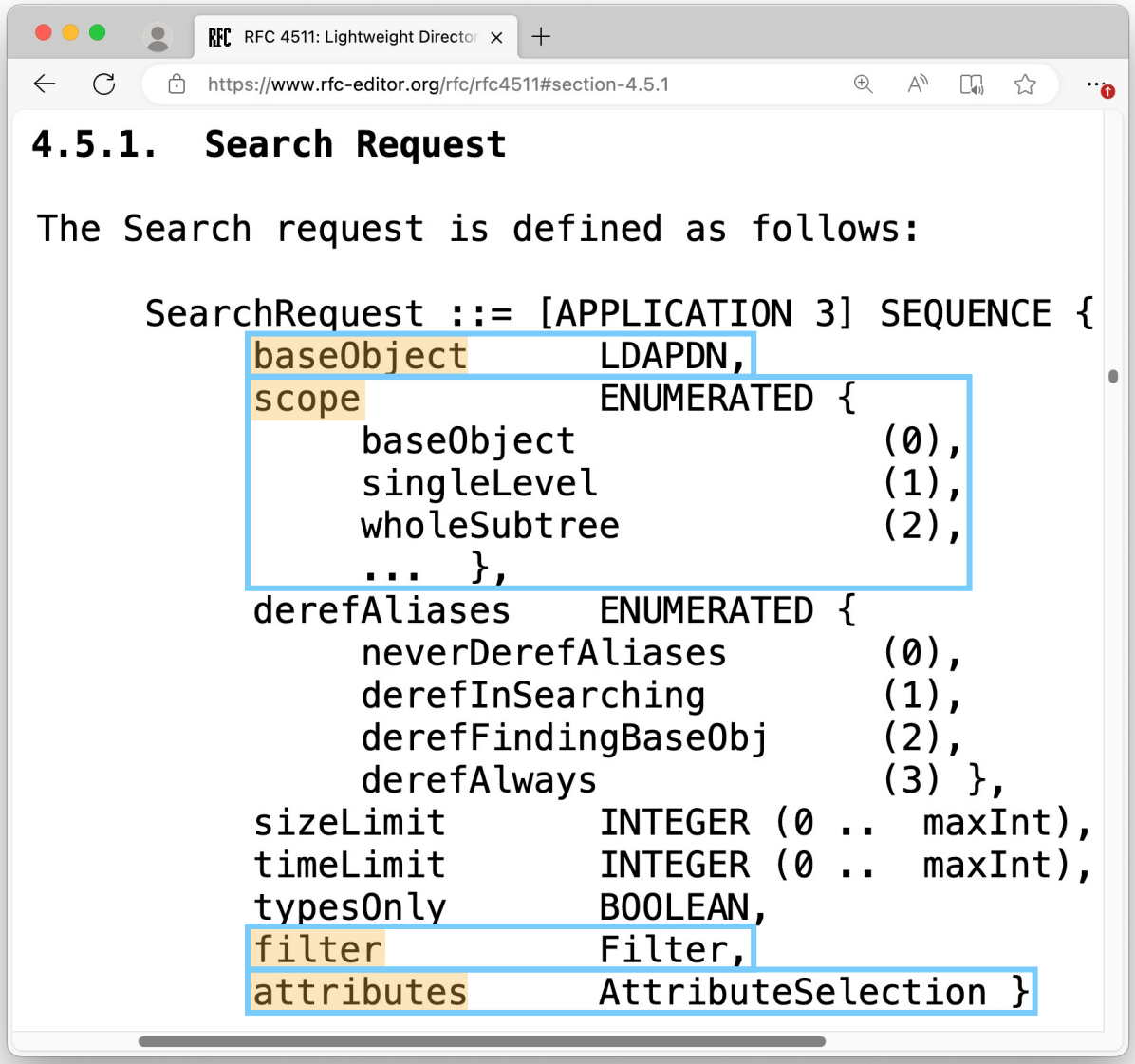
AGENDA

- Introduction
- LDAP Overview 
- PROBLEM: Obfuscating LDAP
- SOLUTION: Parse, Enrich, Detect
- MaLDAPtive Tool Demo + Release



Anatomy of an LDAP SearchRequest (RFC 4511)

- RFC 4511 Section 4.5.1
  - BaseObject
    - LDAP://DC=contoso,DC=com
  - Scope
    - Subtree
  - Filter
    - (name=Sabi)
  - AttributeSelection
    - name,lastlogon,memberof



Anatomy of an LDAP SearchRequest (RFC 4511)

- RFC 4511 Section 4.5.1
  - BaseObject
    - `LDAP://DC=contoso,DC=com`
  - Scope
    - `Subtree`
  - Filter
    - `(name=Sabi)`
  - AttributeSelection
    - `name,lastlogon,memberof`

Secondary obfuscation focus

Primary detection focus

Primary obfuscation focus

Tertiary obfuscation focus

Anatomy of an LDAP SearchFilter (RFC 1558)

- Required tokens
  - GroupStart, Attribute, ComparisonOperator, Value, GroupEnd



Anatomy of an LDAP SearchFilter (RFC 1558)

- Required tokens
  - GroupStart, Attribute, ComparisonOperator, Value, GroupEnd
- Optional tokens
  - BooleanOperator, ExtensibleMatchFilter

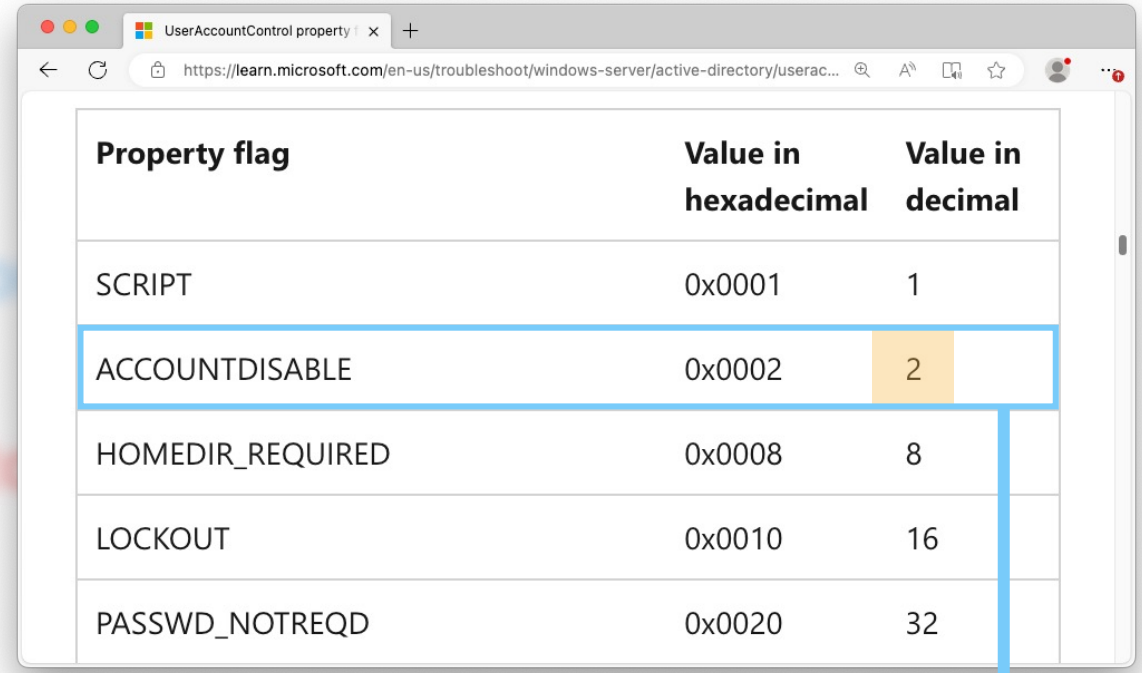
```
(name=Sabi)

(!(userAccountControl:1.2.840.113556.1.4.803:=2))
```

BooleanOperator

ExtensibleMatchFilter

Anatomy of an LDAP SearchFilter (RFC 1558)

- Required tokens
  - GroupStart, Attribute, Comparison
- Optional tokens
  - BooleanOperator, ExtensibleMatch




```
(name=Sabi)

(!(userAccountControl:1.2.840.113556.1.4.803:=2))
```

BooleanOperator

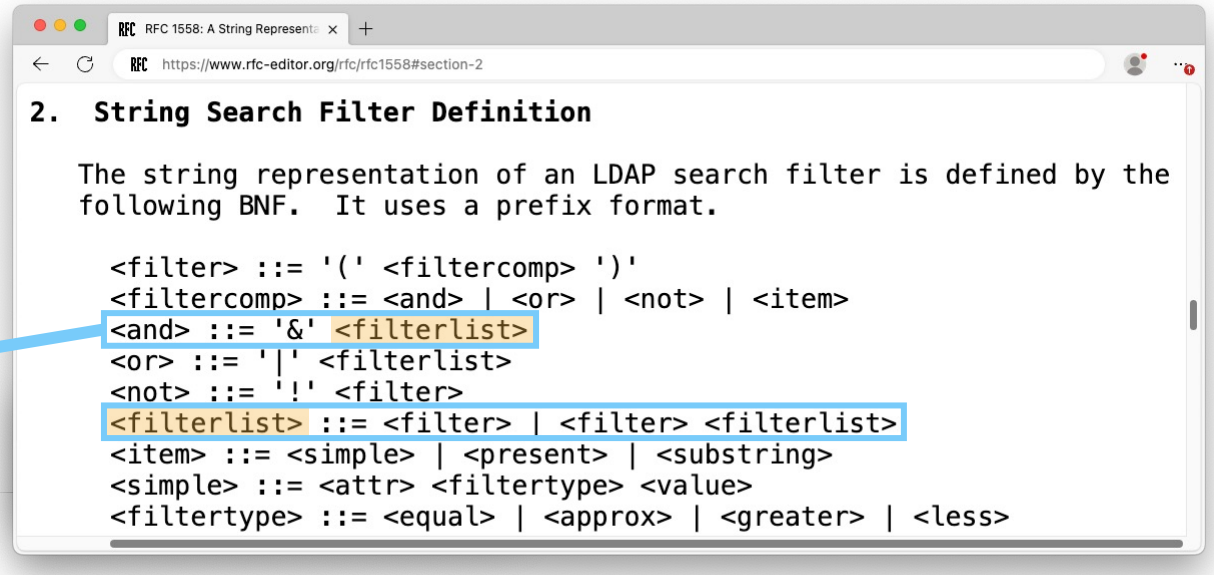
ExtensibleMatchFilter

"Object is NOT disabled."

Anatomy of an LDAP SearchFilter (RFC 1558)

```
(name=Sabi)

(!(userAccountControl:1.2.840.113556.1.4.803:=2))
```

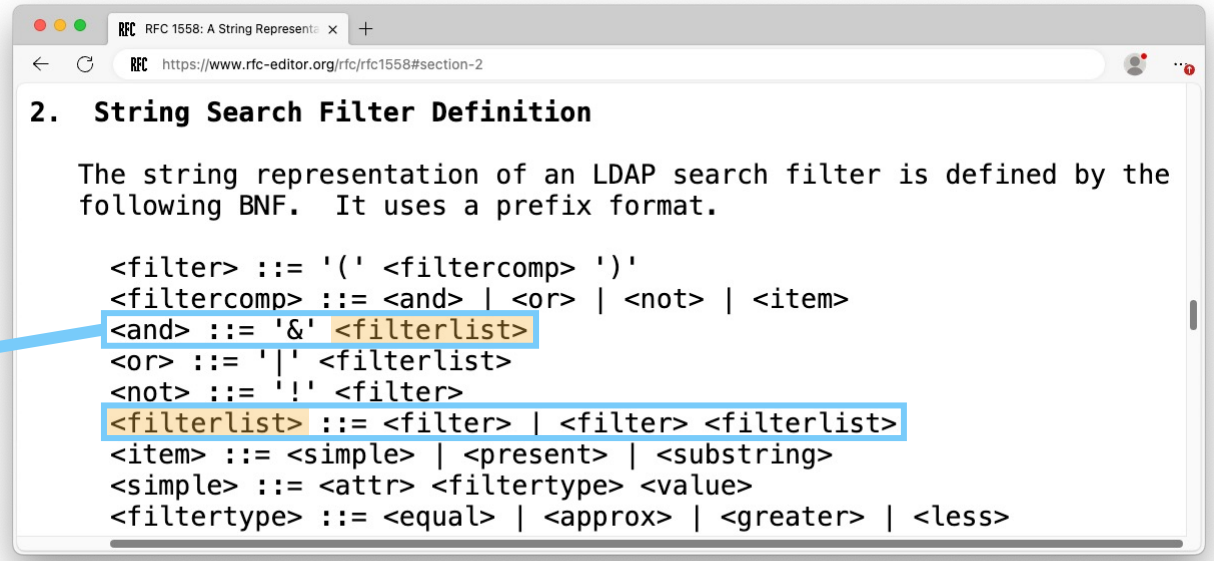
Anatomy of an LDAP SearchFilter (RFC 1558)

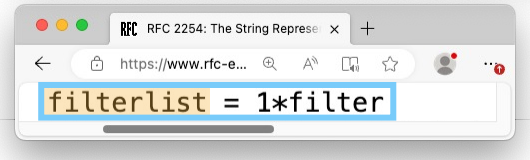



```
(
  &
  (name=Sabi)
  (!(userAccountControl:1.2.840.113556.1.4.803:=2))
)
```

Anatomy of an LDAP SearchFilter (RFC 1558)

```
(
  &
  (name=Sabi)
  (!(userAccountControl:1.2.840.113556.1.4.803:=2))
)
```

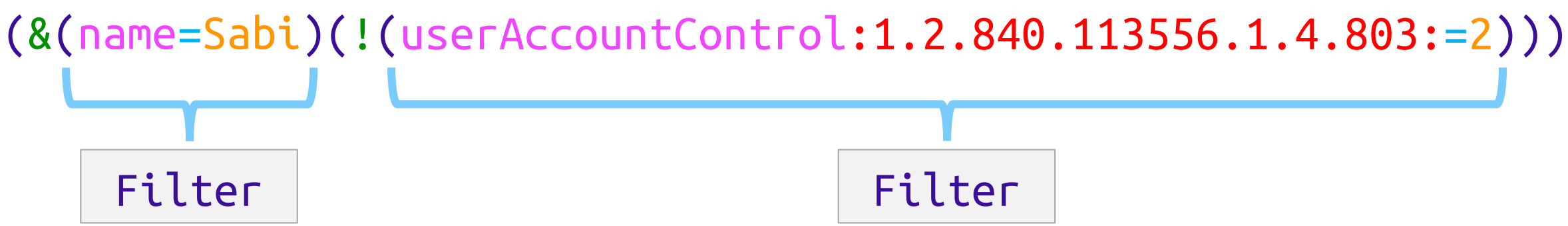




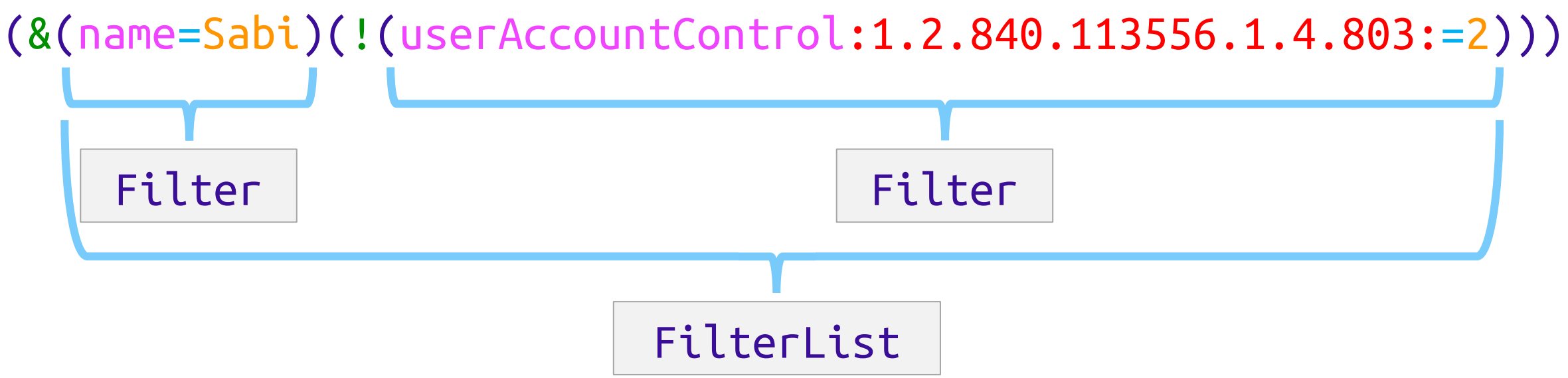
Anatomy of an LDAP SearchFilter (RFC 1558)

```
(&(name=Sabi)(!(userAccountControl:1.2.840.113556.1.4.803:=2)))
```

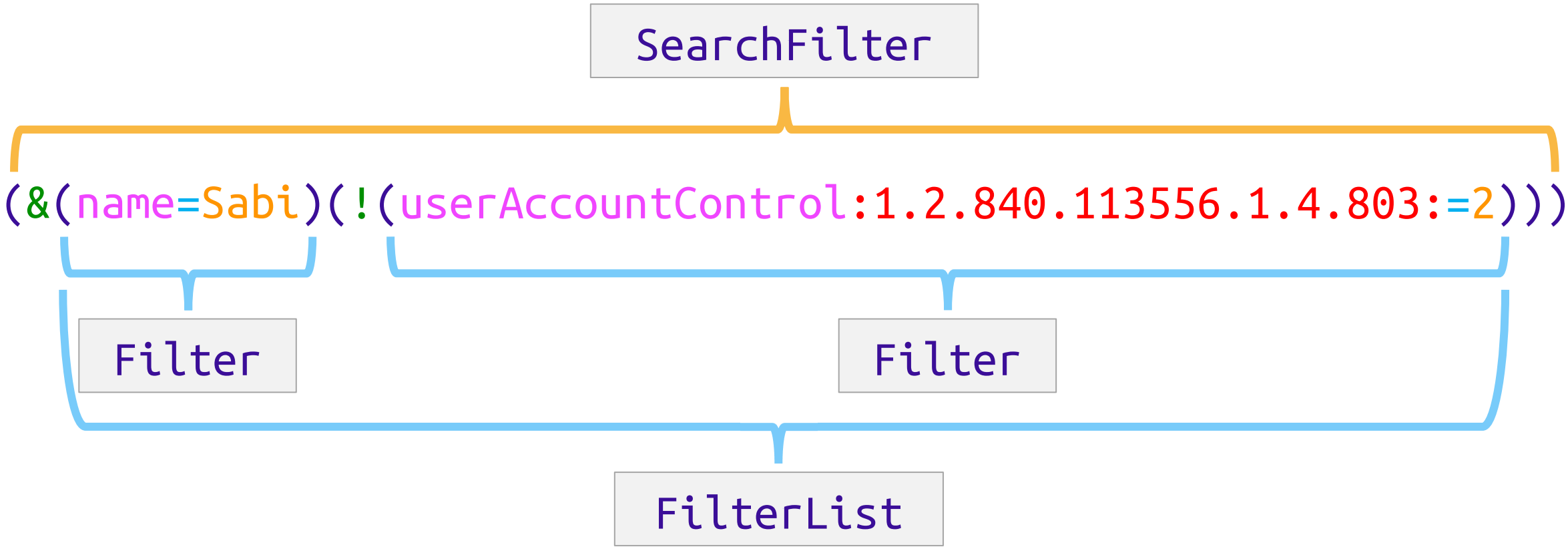
Anatomy of an LDAP SearchFilter (RFC 1558)



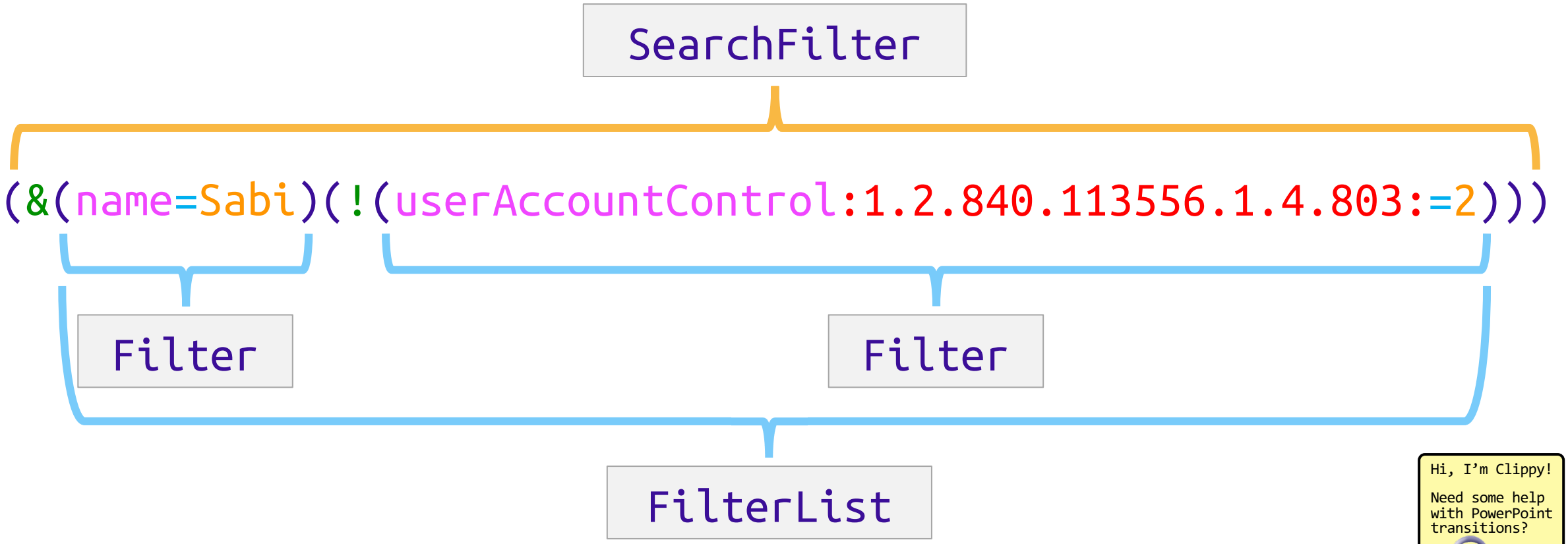
Anatomy of an LDAP SearchFilter (RFC 1558)

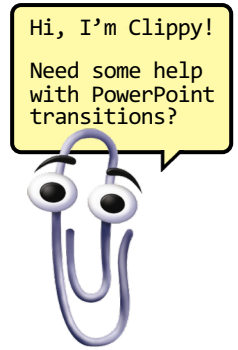


Anatomy of an LDAP SearchFilter (RFC 1558)

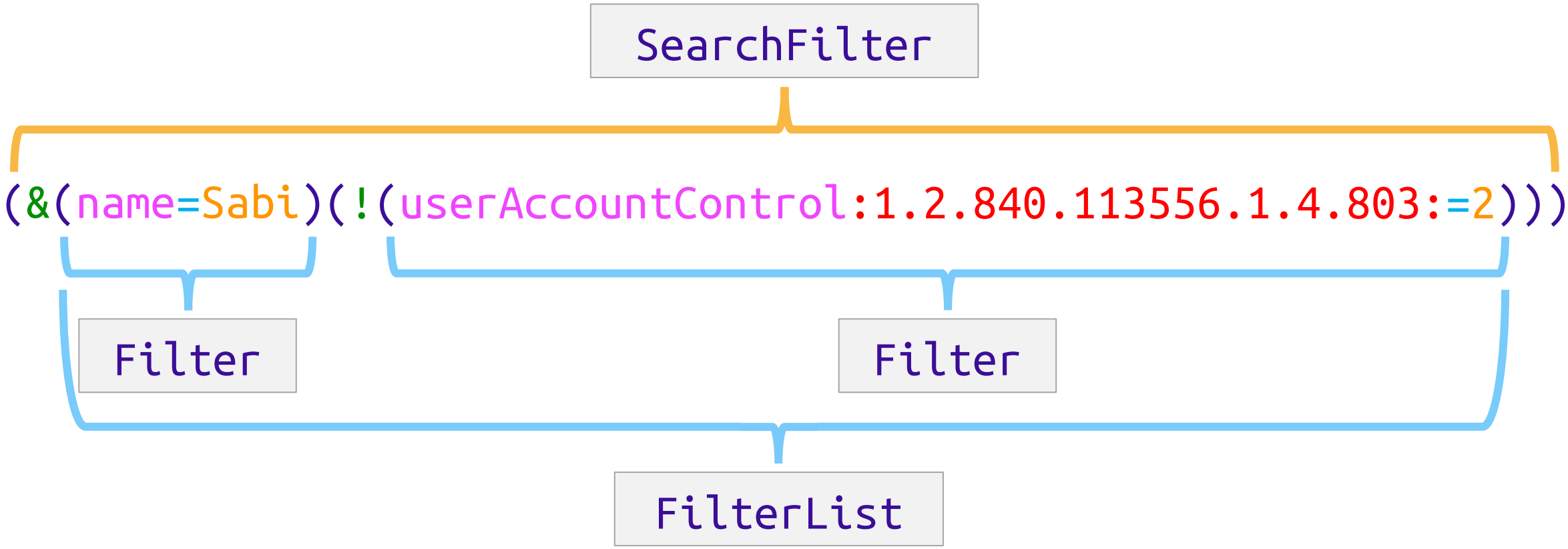


Anatomy of an LDAP SearchFilter (RFC 1558)

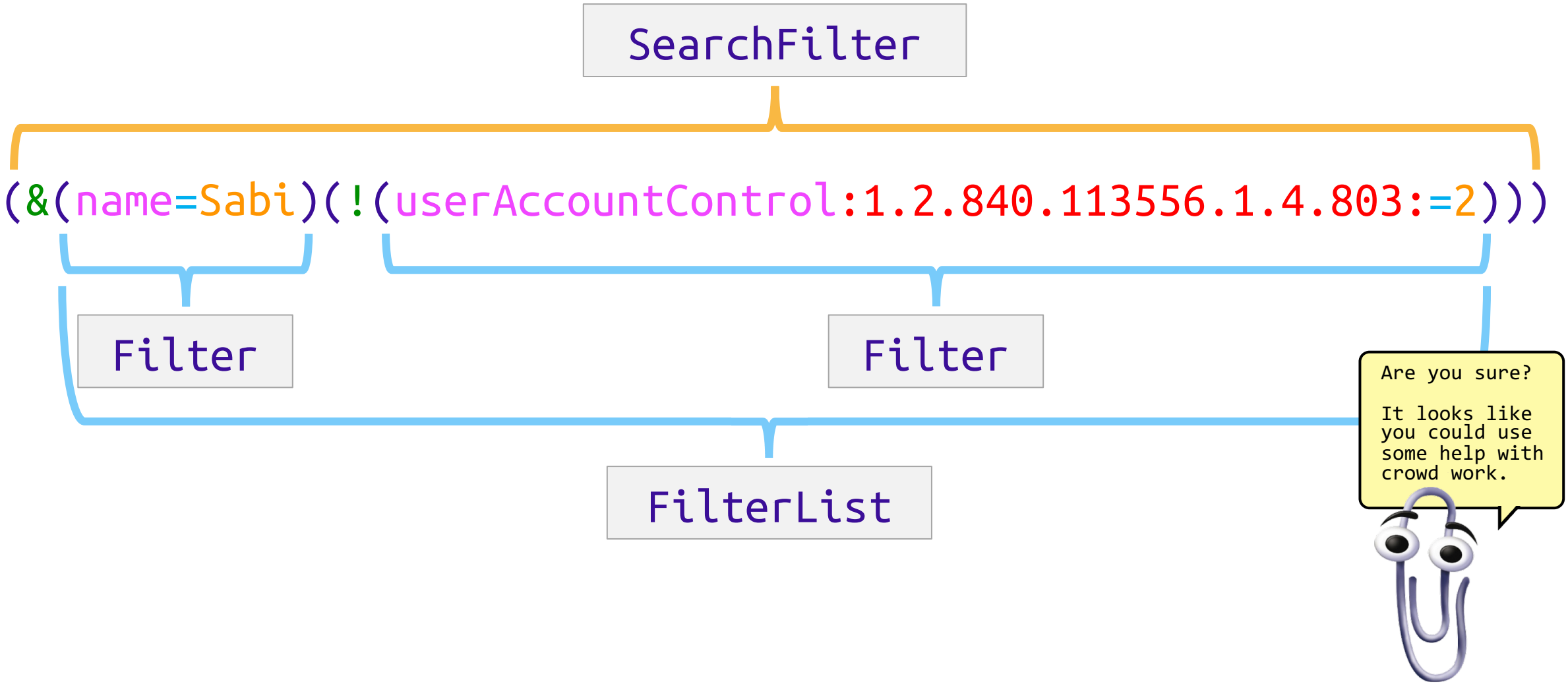




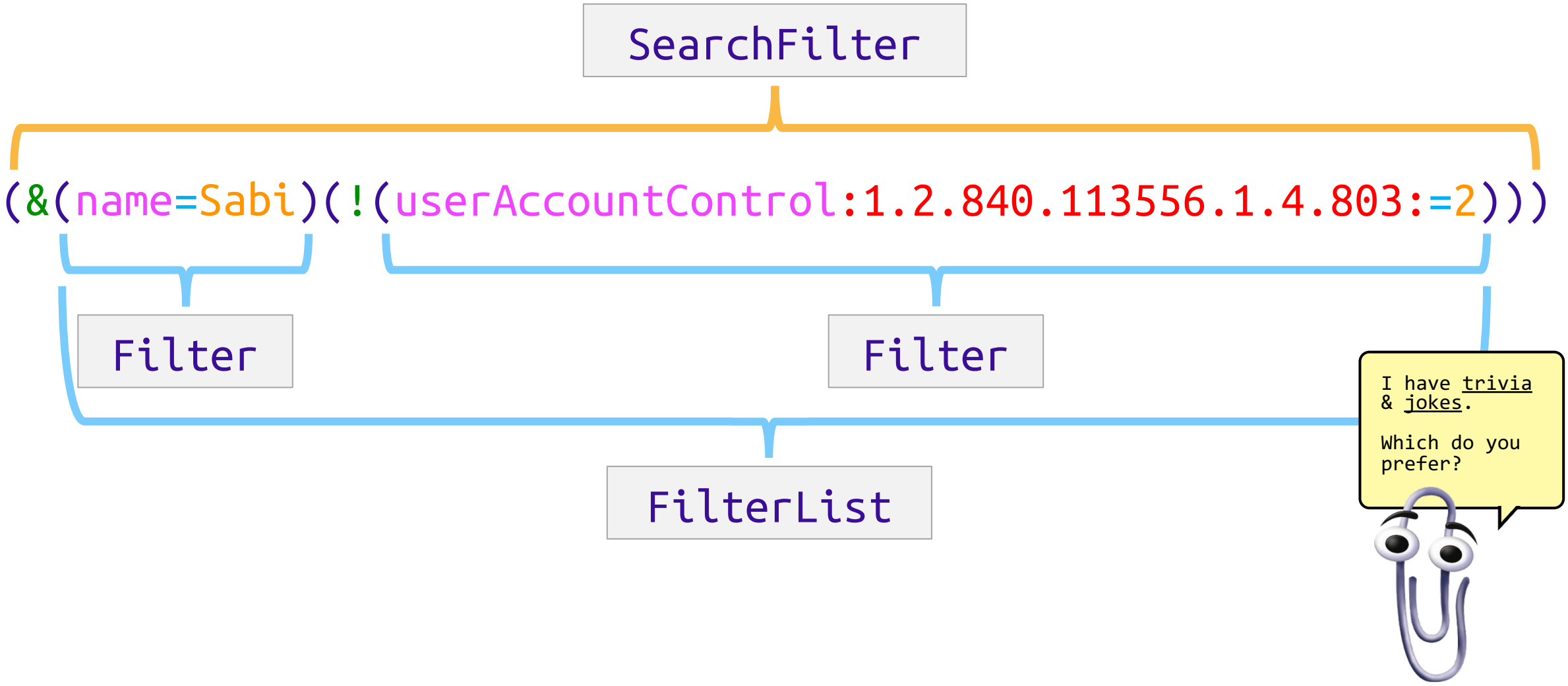
Anatomy of an LDAP SearchFilter (RFC 1558)



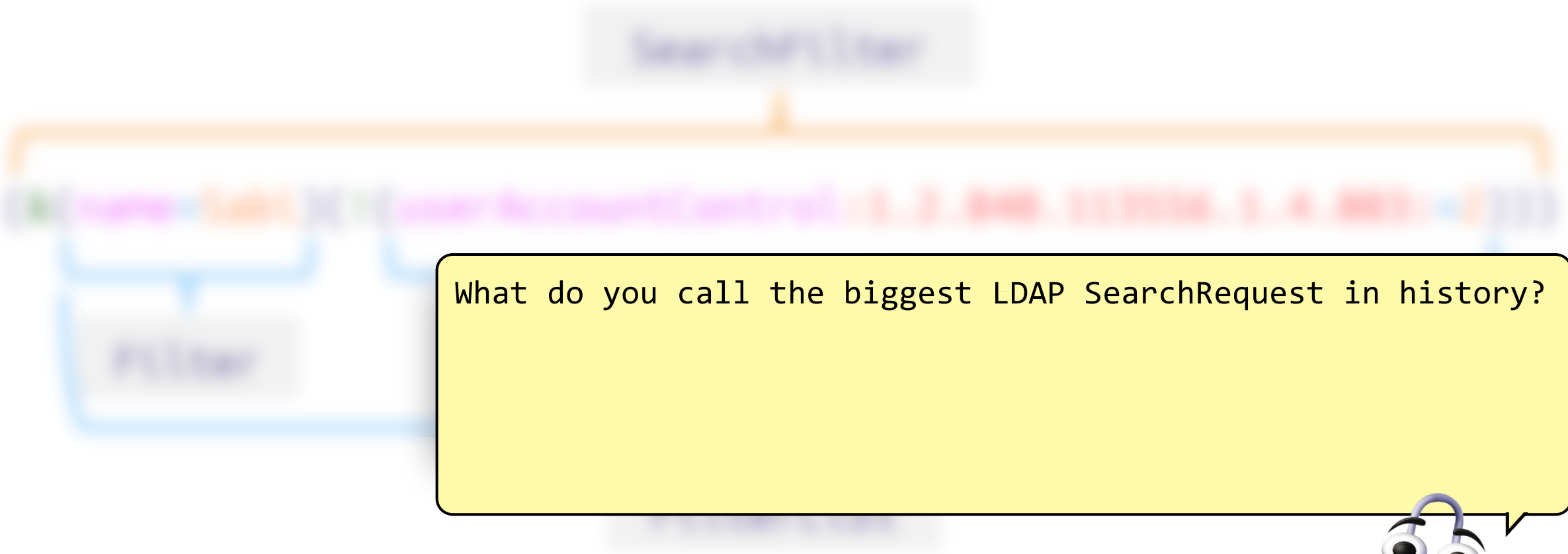
Anatomy of an LDAP SearchFilter (RFC 1558)



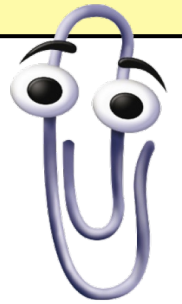
Anatomy of an LDAP SearchFilter (RFC 1558)



Anatomy of an LDAP SearchFilter (RFC 1558)



What do you call the biggest LDAP SearchRequest in history?

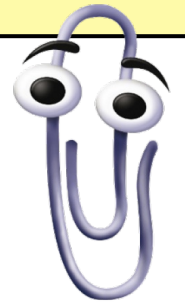


Anatomy of an LDAP SearchFilter (RFC 1558)

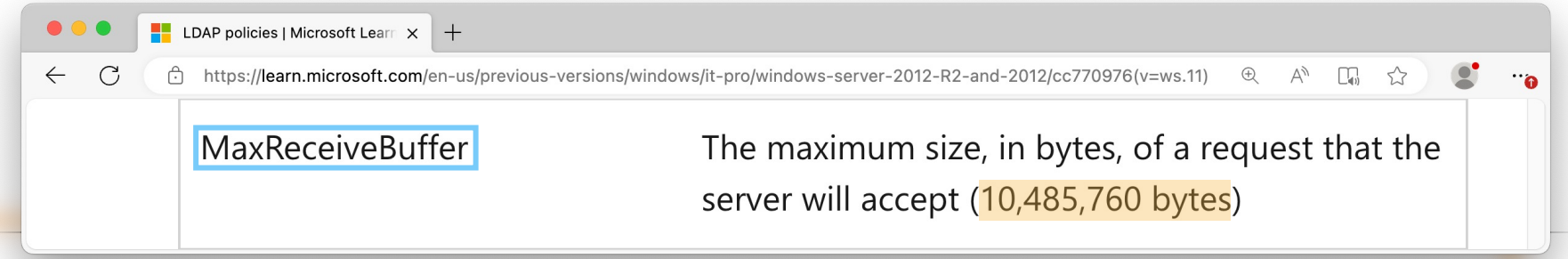
What do you call the biggest LDAP SearchRequest in history?

10,485,760 bytes (~10.5 MB)

That’s so funny it qualifies as trivia AND a joke!



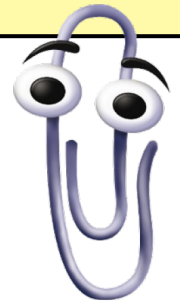
Anatomy of an LDAP SearchFilter (RFC 1558)



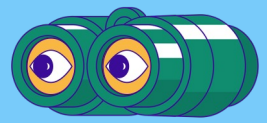
What do you call the biggest LDAP SearchRequest in history?

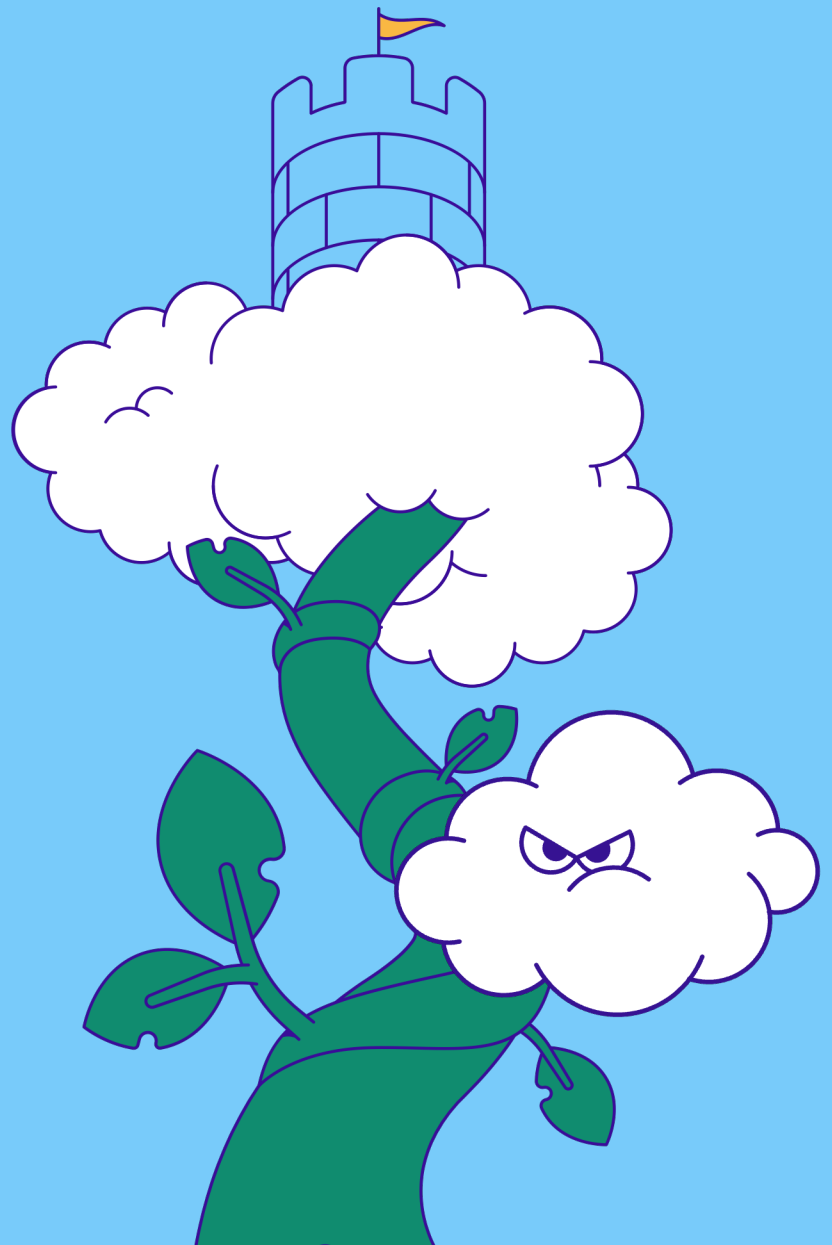
10,485,760 bytes (~10.5 MB)

That's so funny it qualifies as trivia AND a joke!



AGENDA

- Introduction
- LDAP Overview
- PROBLEM: Obfuscating LDAP 
- SOLUTION: Parse, Enrich, Detect
- MaLDAPtive Tool Demo + Release



PROBLEM: Obfuscating LDAP 👀



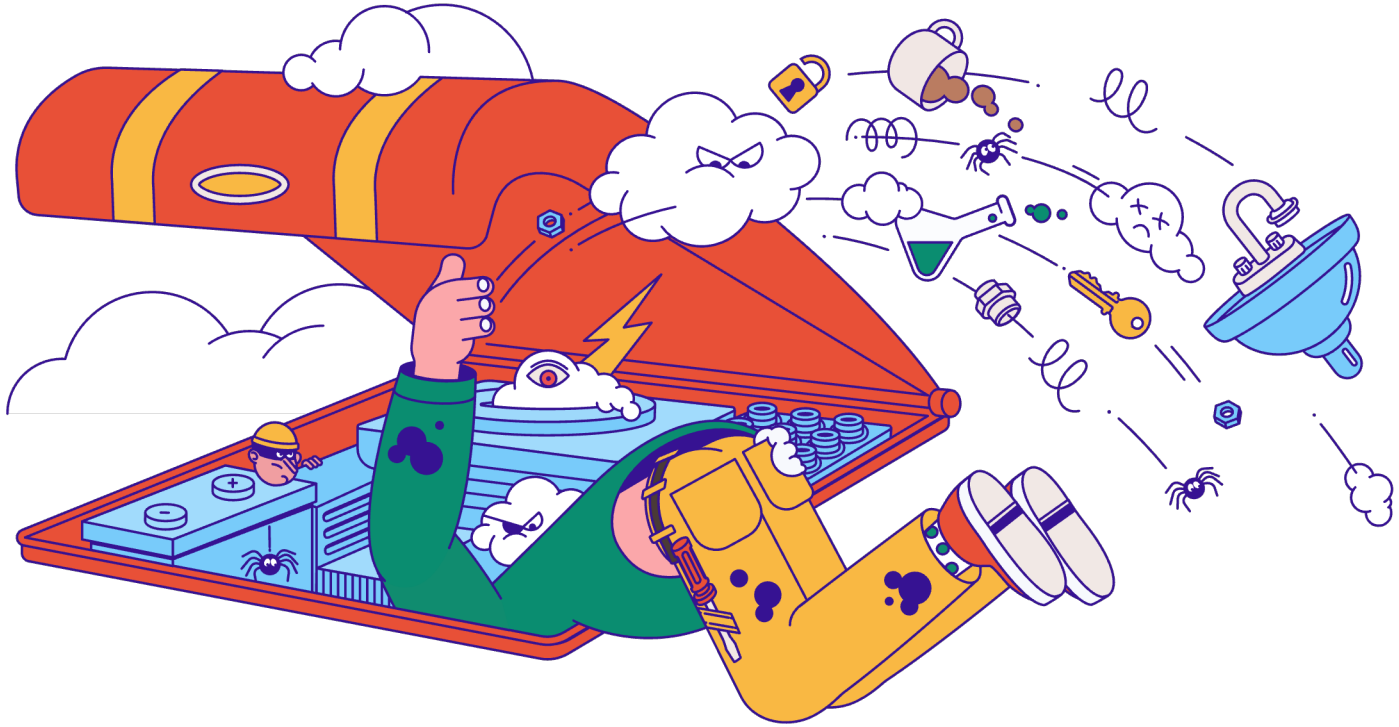
Obfuscating LDAP: 3 Deep Dives * 3 Genres

- Obfuscation Deep Dives:
  1. Filter → `(name=Sabi)`
  2. BaseObject → `LDAP://DC=contoso,DC=com`
  3. AttributeSelection → `name,lastlogon,memberof`
- Obfuscation Genres:
  - Substitution
  - Insertion
  - Transformation



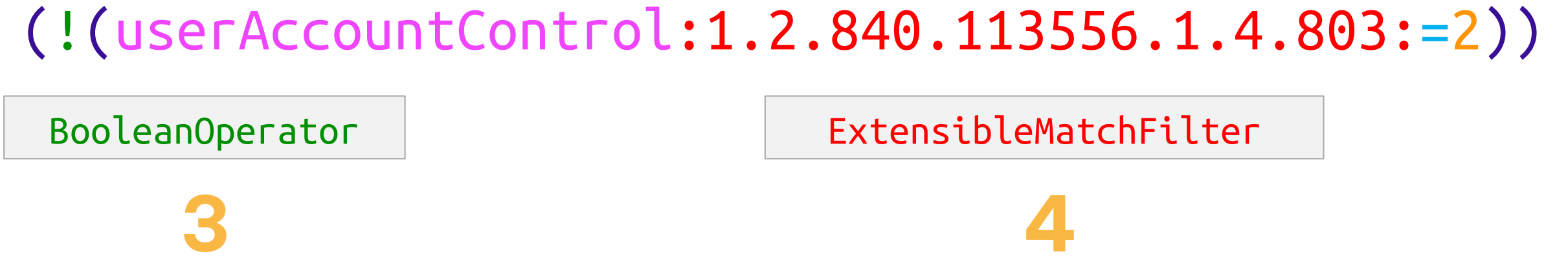
WARNING! Undocumented Findings Ahead





LDAP SearchFilter Token Types





Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing

```
(name=Domain Admins)
```

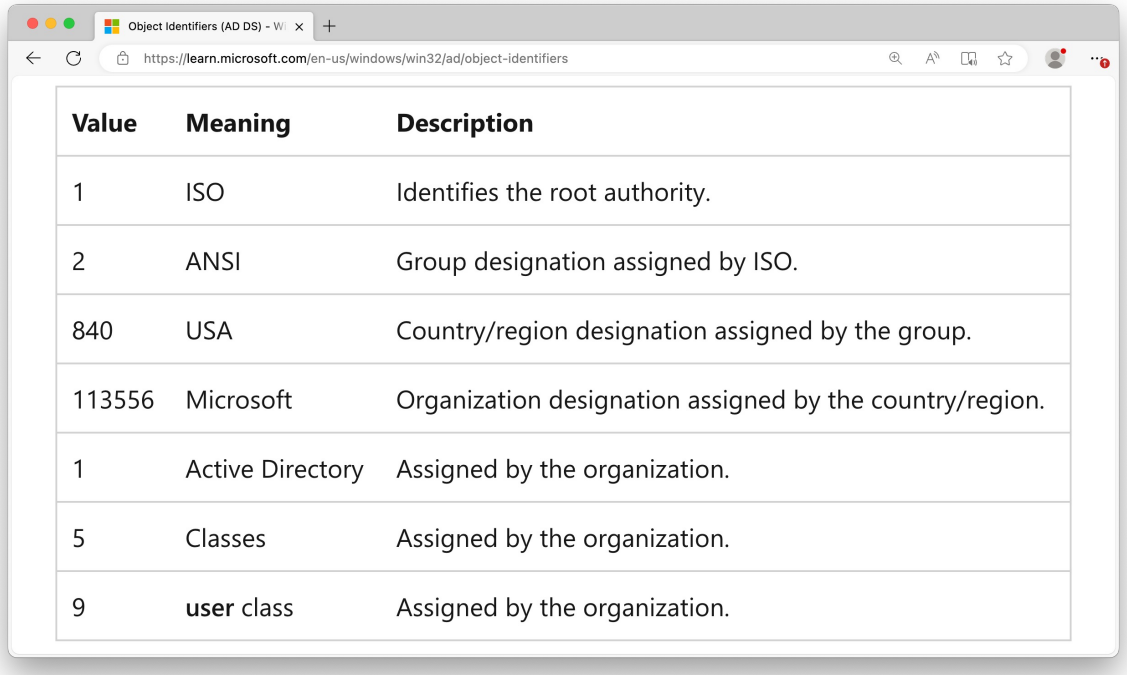
Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing

```
(nAmE=Domain Admins)
```

Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing
  - OID (Object Identifiers) notation
    - Unique numeric values to identify object classes, **attributes** & syntaxes
    - Defined by directory service
    - *OID notation is a **dotted string of numbers** which is described in the following table*

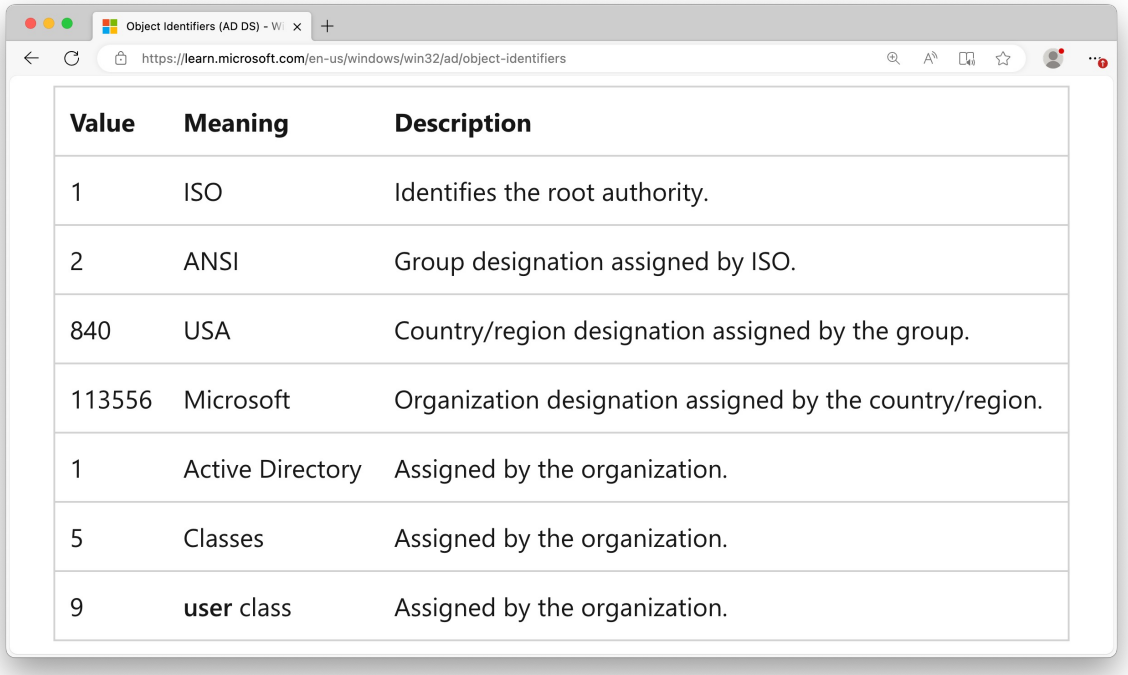


Value	Meaning	Description
1	ISO	Identifies the root authority.
2	ANSI	Group designation assigned by ISO.
840	USA	Country/region designation assigned by the group.
113556	Microsoft	Organization designation assigned by the country/region.
1	Active Directory	Assigned by the organization.
5	Classes	Assigned by the organization.
9	**user** class	Assigned by the organization.

```
(1.2.840.113556.1.4.1=Domain Admins)
```

Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing
  - OID (Object Identifiers) notation
    - Unique numeric values to identify object classes, **attributes** & syntaxes
    - Defined by directory service
    - *OID notation is a **dotted string of numbers** which is described in the following table*

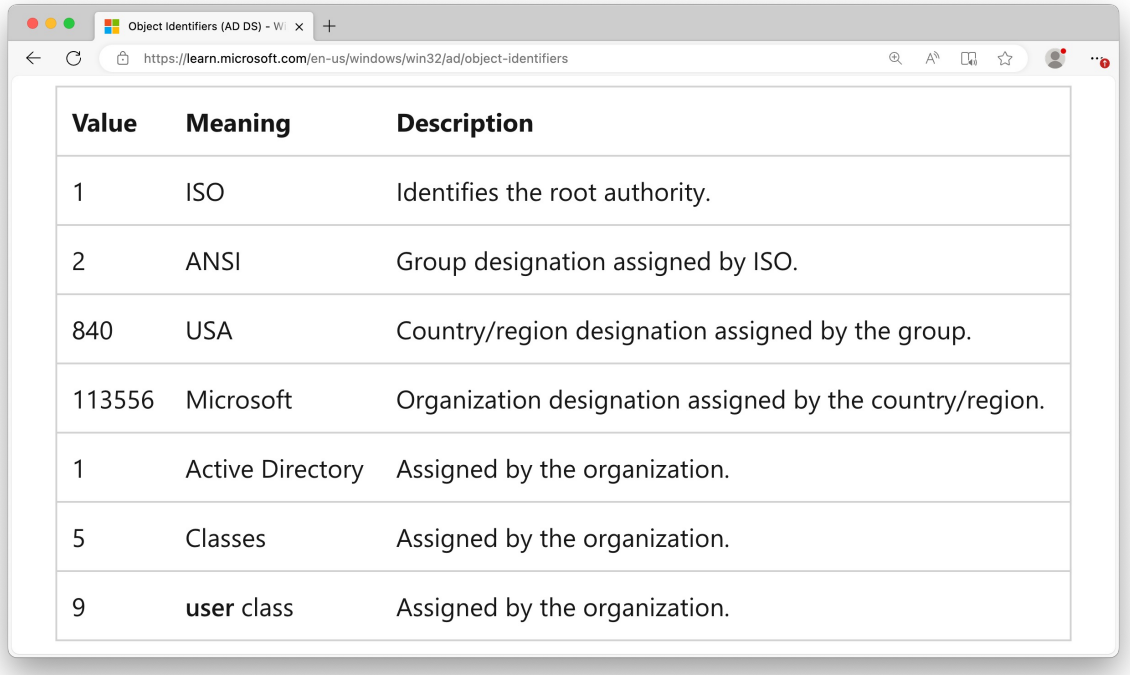


Value	Meaning	Description
1	ISO	Identifies the root authority.
2	ANSI	Group designation assigned by ISO.
840	USA	Country/region designation assigned by the group.
113556	Microsoft	Organization designation assigned by the country/region.
1	Active Directory	Assigned by the organization.
5	Classes	Assigned by the organization.
9	**user** class	Assigned by the organization.

```
(OID.1.2.840.113556.1.4.1=Domain Admins)
```

Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing
  - OID (Object Identifiers) notation
    - Unique numeric values to identify object classes, **attributes** & syntaxes
    - Defined by directory service
    - *OID notation is a **dotted string of numbers** which is described in the following table*



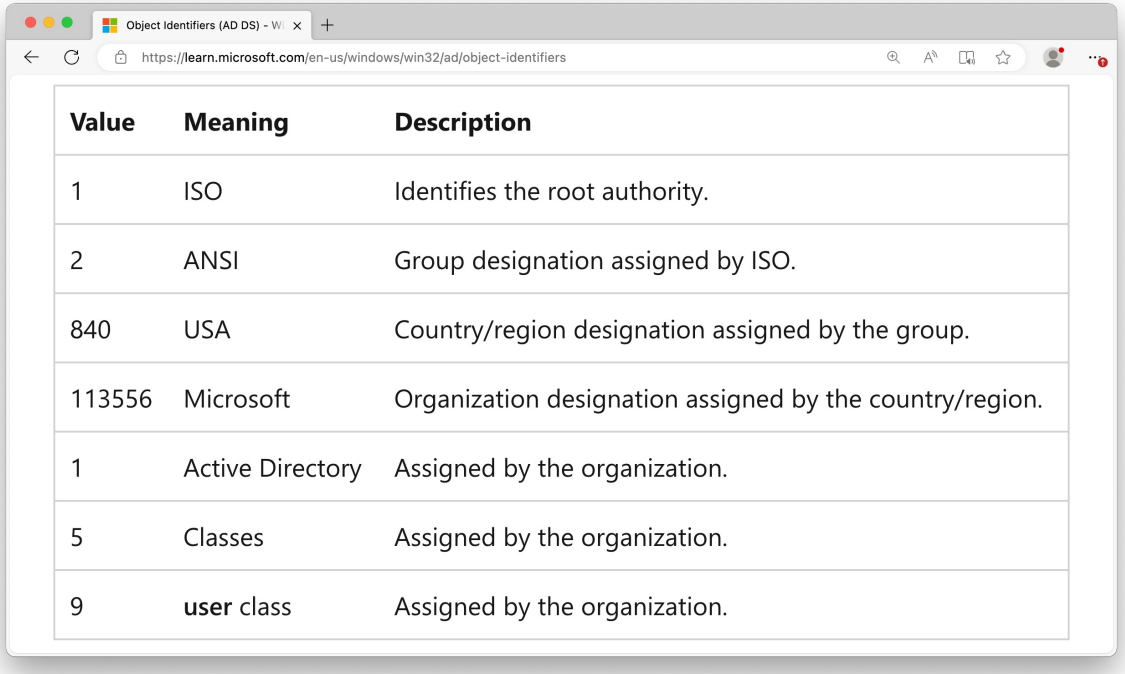
Value	Meaning	Description
1	ISO	Identifies the root authority.
2	ANSI	Group designation assigned by ISO.
840	USA	Country/region designation assigned by the group.
113556	Microsoft	Organization designation assigned by the country/region.
1	Active Directory	Assigned by the organization.
5	Classes	Assigned by the organization.
9	**user** class	Assigned by the organization.

```
(oId.1.2.840.113556.1.4.1=Domain Admins)
```

Obfuscation::Filter::Attribute



- Attribute obfuscation
  - Casing
  - OID (Object Identifiers) notation
    - Unique numeric values to identify object classes, **attributes** & syntaxes
    - Defined by directory service
    - *OID notation is a **dotted string of numbers** which is described in the following table*



Value	Meaning	Description
1	ISO	Identifies the root authority.
2	ANSI	Group designation assigned by ISO.
840	USA	Country/region designation assigned by the group.
113556	Microsoft	Organization designation assigned by the country/region.
1	Active Directory	Assigned by the organization.
5	Classes	Assigned by the organization.
9	**user** class	Assigned by the organization.

(oId.001.02.000840.00113556.000001.04.001=Domain Admins)

Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing
  - OID (Object Identifiers) notation
  - aNR (Ambiguous Name Resolution)

```
(name=Domain Admins)
```

Obfuscation::Filter::Attribute

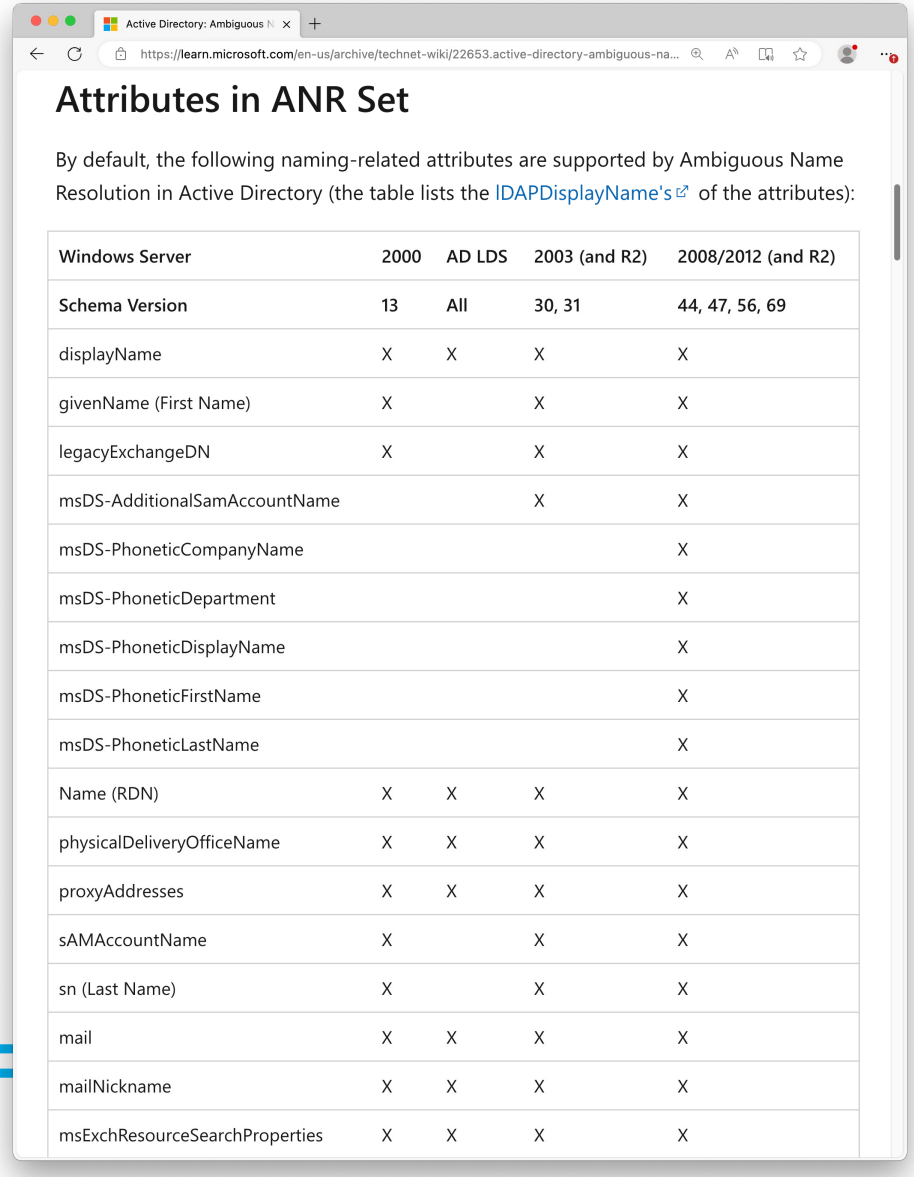
- Attribute obfuscation
  - Casing
  - OID (Object Identifiers) notation
  - aNR (Ambiguous Name Resolution)

```
(name=Domain Admins)
        ↓
(aNR=Domain Admins)
        ↓
(1.2.840.113556.1.4.1208=Domain Admins)
```

Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing
  - OID (Object Identifiers) notation
  - aNR (Ambiguous Name Resolution)

```
(name=Domain Admins)
        ↓
(aNR=Domain Admins)
        ↓
(1.2.840.113556.1.4.1208=
```



Attributes in ANR Set

By default, the following naming-related attributes are supported by Ambiguous Name Resolution in Active Directory (the table lists the lDAPDisplayName's of the attributes):

Windows Server	2000	AD LDS	2003 (and R2)	2008/2012 (and R2)
Schema Version	**13**	**All**	**30, 31**	**44, 47, 56, 69**
displayName	X	X	X	X
givenName (First Name)	X		X	X
legacyExchangeDN	X		X	X
msDS-AdditionalSamAccountName			X	X
msDS-PhoneticCompanyName				X
msDS-PhoneticDepartment				X
msDS-PhoneticDisplayName				X
msDS-PhoneticFirstName				X
msDS-PhoneticLastName				X
Name (RDN)	X	X	X	X
physicalDeliveryOfficeName	X	X	X	X
proxyAddresses	X	X	X	X
sAMAccountName	X		X	X
sn (Last Name)	X		X	X
mail	X	X	X	X
mailNickname	X	X	X	X
msExchResourceSearchProperties	X	X	X	X

Obfuscation::Filter::Attribute

```
(
   |
   (displayName=domain admins*)
   (givenName=domain admins*)
   (legacyExchangeDN=domain admins)
   (physicalDeliveryOfficeName=domain admins*)
   (proxyAddresses=domain admins*)
   (Name=domain admins*)
   (sAMAccountName=domain admins*)
   (sn=domain admins*)
)
```

No “*”



Active Directory: Ambiguous

https://learn.microsoft.com/en-us/archive/technet-wiki/22653.active-directory-ambiguous-na...

Attributes in ANR Set

By default, the following naming-related attributes are supported by Ambiguous Name Resolution in Active Directory (the table lists the lDAPDisplayName's of the attributes):

Windows Server	2000	AD LDS	2003 (and R2)	2008/2012 (and R2)
Schema Version	**13**	**All**	**30, 31**	**44, 47, 56, 69**
displayName	X	X	X	X
givenName (First Name)	X		X	X
legacyExchangeDN	X		X	X
msDS-AdditionalSamAccountName			X	X
msDS-PhoneticCompanyName				X
msDS-PhoneticDepartment				X
msDS-PhoneticDisplayName				X
msDS-PhoneticFirstName				X
msDS-PhoneticLastName				X
Name (RDN)	X	X	X	X
physicalDeliveryOfficeName	X	X	X	X
proxyAddresses	X	X	X	X
sAMAccountName	X		X	X
sn (Last Name)	X		X	X
mail	X	X	X	X
mailNickname	X	X	X	X
msExchResourceSearchProperties	X	X	X	X

Obfuscation::Filter::Attribute

```
(
  |
  (displayName=domain admins*)
  (givenName=domain admins*)
  (legacyExchangeDN=domain admins)
  (physicalDeliveryOfficeName=domain admins*)
  (proxyAddresses=domain admins*)
  (Name=domain admins*)
  (sAMAccountName=domain admins*)
  (sn=domain admins*)
  (givenName=domain*)
  (sn=admins*)
  (givenName=admins*)
  (sn=domain*)
)
```

No “*”

First*Last*

Last*First*



Active Directory: Ambiguous

https://learn.microsoft.com/en-us/archive/technet-wiki/22653.active-directory-ambiguous-na...

Attributes in ANR Set

By default, the following naming-related attributes are supported by Ambiguous Name Resolution in Active Directory (the table lists the lDAPDisplayName's of the attributes):

Windows Server	2000	AD LDS	2003 (and R2)	2008/2012 (and R2)
Schema Version	**13**	**All**	**30, 31**	**44, 47, 56, 69**
displayName	X	X	X	X
givenName (First Name)	X		X	X
legacyExchangeDN	X		X	X
msDS-AdditionalSamAccountName			X	X
msDS-PhoneticCompanyName				X
msDS-PhoneticDepartment				X
msDS-PhoneticDisplayName				X
msDS-PhoneticFirstName				X
msDS-PhoneticLastName				X
Name (RDN)	X	X	X	X
physicalDeliveryOfficeName	X	X	X	X
proxyAddresses	X	X	X	X
sAMAccountName	X		X	X
sn (Last Name)	X		X	X
mail	X	X	X	X
mailNickname	X	X	X	X
msExchResourceSearchProperties	X	X	X	X

Obfuscation::Filter::Attribute



(aNR=krbtgt)

(aNR=krb)

(aNR=krb*Appeal/AnythingYouWant!!!)



Active Directory: Ambiguous

https://learn.microsoft.com/en-us/archive/technet-wiki/22653.active-directory-ambiguous-na...

Attributes in ANR Set

By default, the following naming-related attributes are supported by Ambiguous Name Resolution in Active Directory (the table lists the lDAPDisplayName's of the attributes):

Windows Server	2000	AD LDS	2003 (and R2)	2008/2012 (and R2)
Schema Version	13	All	30, 31	44, 47, 56, 69
displayName	X	X	X	X
givenName (First Name)	X		X	X
legacyExchangeDN	X		X	X
msDS-AdditionalSamAccountName				
			X	X
mail	X	X	X	X
mailNickname	X	X	X	X
msExchResourceSearchProperties	X	X	X	X

Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing
  - OID (Object Identifiers) notation
  - aNR (Ambiguous Name Resolution)

```
(aNR=Domain Admins)
```

```
(aNR==Domain Admins)
```

You can force ANR to require an exact match on any of the attributes in the table by starting the value with the equal sign, "=" (so the filter has two equal signs).



Attributes in ANR Set

By default, the following naming-related attributes are supported by Ambiguous Name Resolution in Active Directory (the table lists the lDAPDisplayName's of the attributes):

Windows Server	2000	AD LDS	2003 (and R2)	2008/2012 (and R2)
Schema Version	**13**	**All**	**30, 31**	**44, 47, 56, 69**
displayName	X	X	X	X
givenName (First Name)	X		X	X
legacyExchangeDN	X		X	X
msDS-AdditionalSamAccountName			X	X
msDS-PhoneticCompanyName				X
msDS-PhoneticDepartment				X
msDS-PhoneticDisplayName				X
msDS-PhoneticFirstName				X
msDS-PhoneticLastName				X
Name (RDN)	X	X	X	X
physicalDeliveryOfficeName	X	X	X	X
proxyAddresses	X	X	X	X
sAMAccountName	X		X	X
sn (Last Name)	X		X	X
mail	X	X	X	X
mailNickname	X	X	X	X
msExchResourceSearchProperties	X	X	X	X

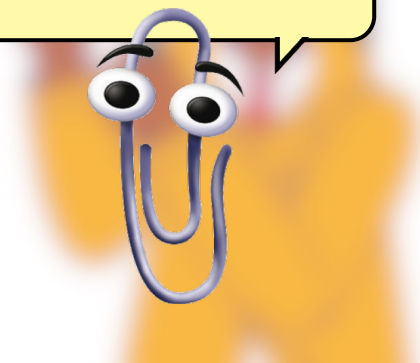
Obfuscation::Filter::Attribute

- Attribute obfuscation
  - Casing
    - `(name=Domain Admins)`
    - `(nAmE=Domain Admins)`
  - OID (Object Identifiers) notation
    - `(1.2.840.113556.1.4.1=Domain Admins)`
    - `(OID.1.2.840.113556.1.4.1=Domain Admins)`
    - `(oId.001.02.000840.00113556.000001.04.001=Domain Admins)`
  - aNR (Ambiguous Name Resolution)
    - `(aNR=Domain Admins)`
    - `(aNR==Domain Admins)`



Obfuscation::Filter::Attribute

Do you know where else Ambiguous Name Resolution exists?

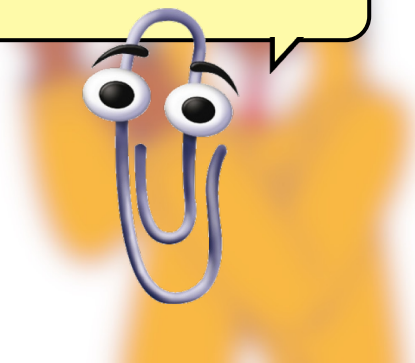


Obfuscation::Filter::Attribute

Do you know where else Ambiguous Name Resolution exists?

Microsoft!

What is more ambiguous than resolving to rename your products every 6 months, amirite?



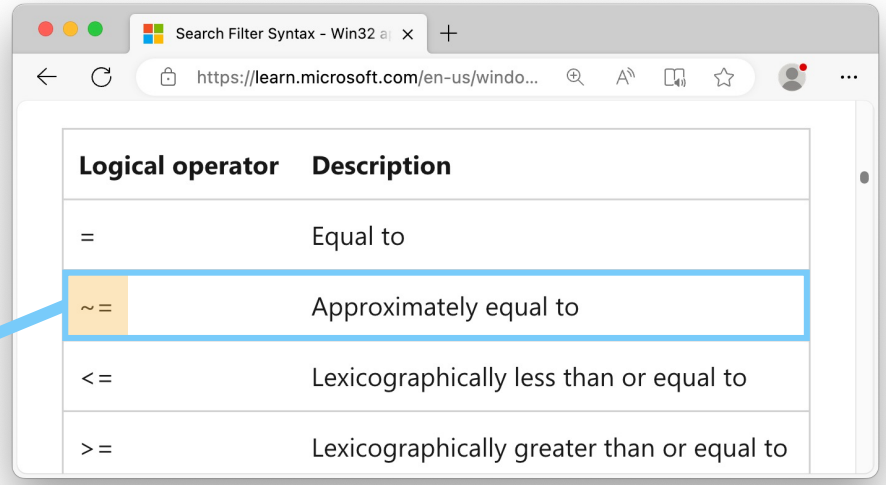
Obfuscation::Filter::ComparisonOperator

- ComparisonOperator obfuscation

```
(name=Domain Admins)
```

Obfuscation::Filter::ComparisonOperator

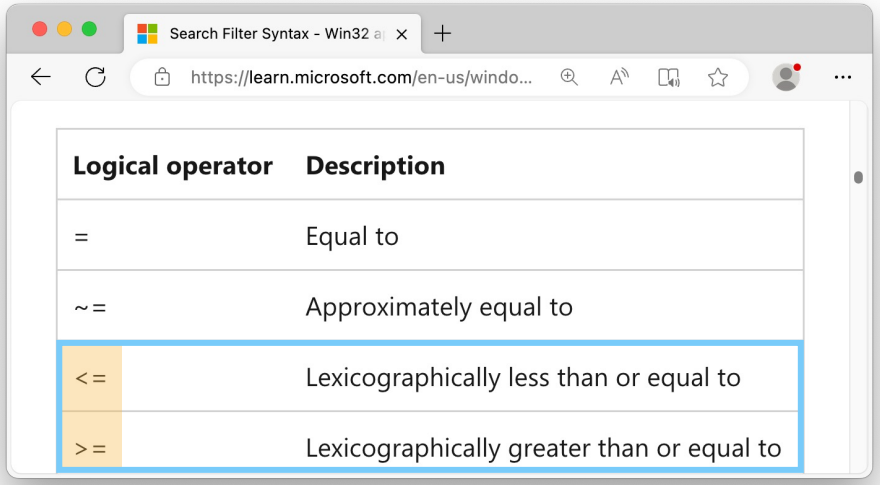
- ComparisonOperator obfuscation
  - Approximately equal to
    - Case still doesn't matter
    - Wildcards non-functional




```
(name~=Domain Admins)
```

Obfuscation::Filter::ComparisonOperator

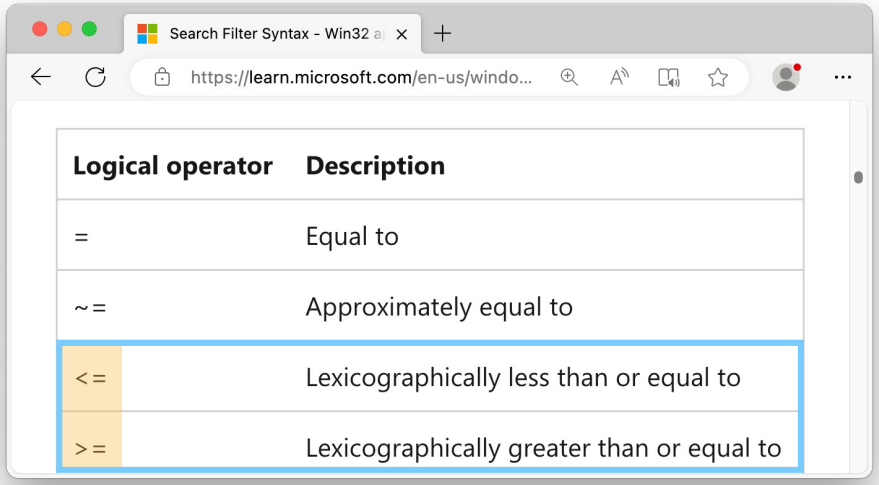
- ComparisonOperator obfuscation
  - Approximately equal to
  - Logically equivalent presence filter




```
(servicePrincipalName=*)
```

Obfuscation::Filter::ComparisonOperator

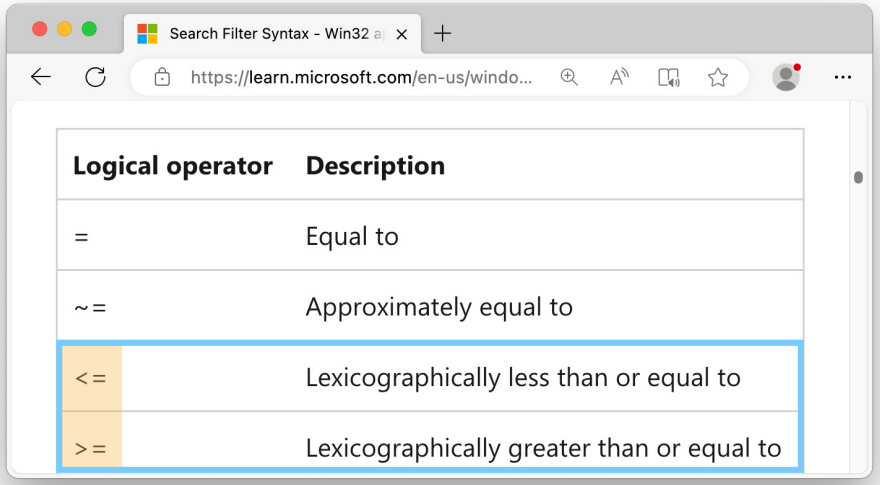
- ComparisonOperator obfuscation
  - Approximately equal to
  - Logically equivalent presence filter




```
(servicePrincipalName>==)
```

Obfuscation::Filter::ComparisonOperator

- ComparisonOperator obfuscation
  - Approximately equal to
  - Logically equivalent presence filter




```
(servicePrincipalName>==)
(servicePrincipalName>=!)
(servicePrincipalName<=zzz)
```

!"#$%&'()*+,-./ 0-9 :;<=>?@ A-Z [\]^_` a-z {|}~

(servicePrincipalName>==)

(servicePrincipalName>=!)

(servicePrincipalName<=zzz)

!"#$%&'()*+,-./ 0-9 :;<=>?@ A-Z [\]^_` a-z {|}~

0-9 a-z 

(servicePrincipalName>==)

(servicePrincipalName>=!)

(servicePrincipalName<=zzz)

!"#$%&'()*+,-./ 0-9 :;<=>?@ A-Z [\]^_` a-z {|}~

'-

+<=> 0-9 a-z



(servicePrincipalName>==)

(servicePrincipalName>=!)

(servicePrincipalName<=zzz)

!"#$%&'()*+,-./ 0-9 :;<=>?@ A-Z [\]^_` a-z {|}~

'- !"#$%&()*,./:;?@[\]^_`{|}~+<=> 0-9 a-z



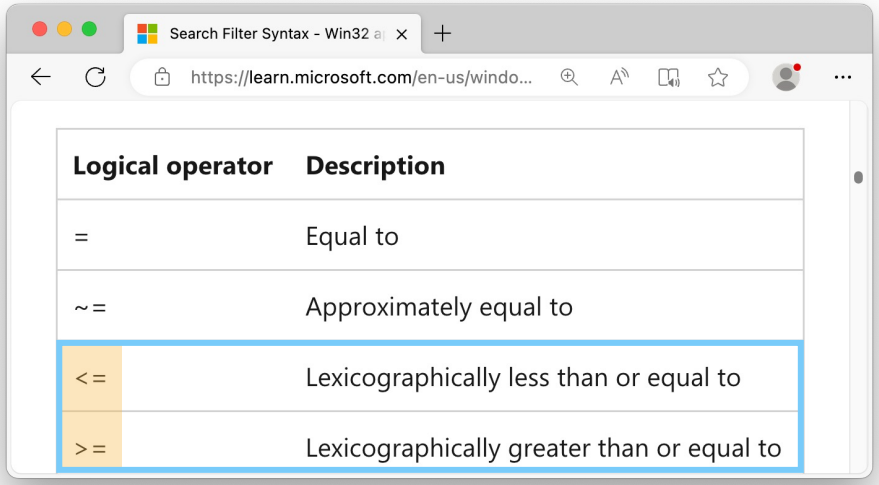
(servicePrincipalName>==)

(servicePrincipalName>=!)

(servicePrincipalName<=zzz)

Obfuscation::Filter::ComparisonOperator

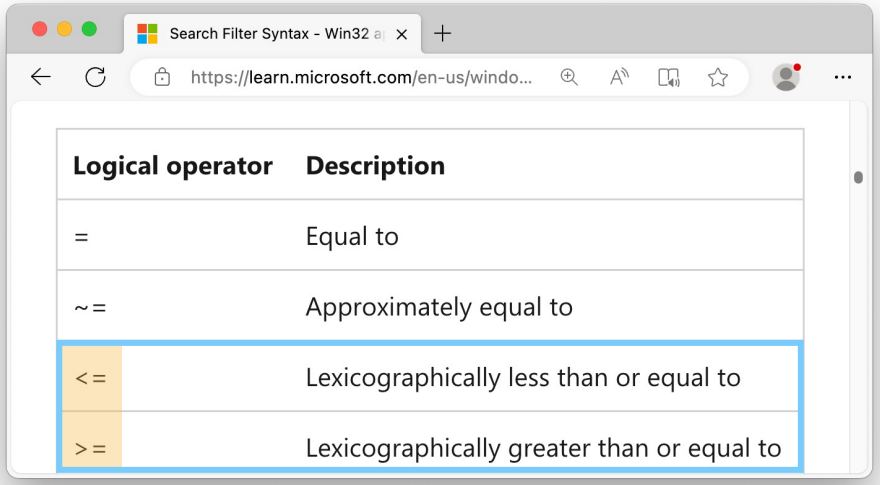
- ComparisonOperator obfuscation
  - Approximately equal to
  - Logically equivalent presence filter
  - Logically equivalent range filters




```
(samaccounttype=805306368)
```

Obfuscation::Filter::ComparisonOperator

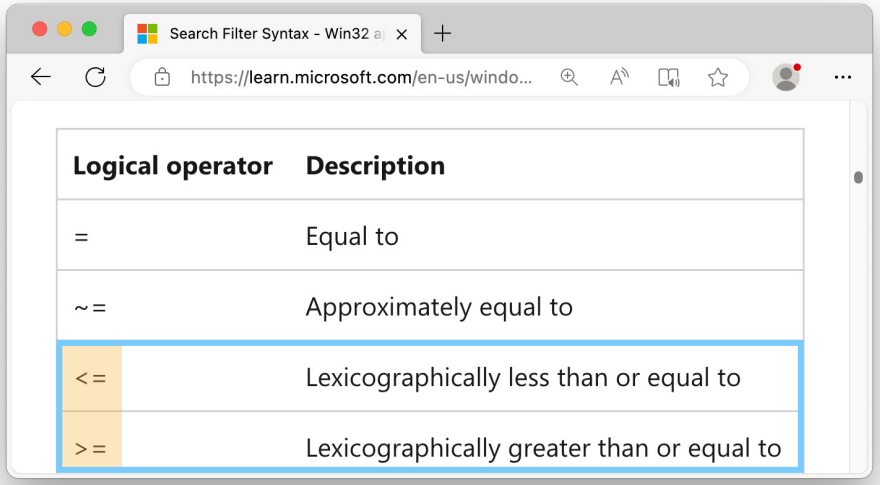
- ComparisonOperator obfuscation
  - Approximately equal to
  - Logically equivalent presence filter
  - Logically equivalent range filters




```
(
    &
    (samaccounttype>=805306367)
    (samaccounttype<=805306369)
)
```

Obfuscation::Filter::ComparisonOperator

- ComparisonOperator obfuscation
  - Approximately equal to
  - Logically equivalent presence filter
  - Logically equivalent range filters

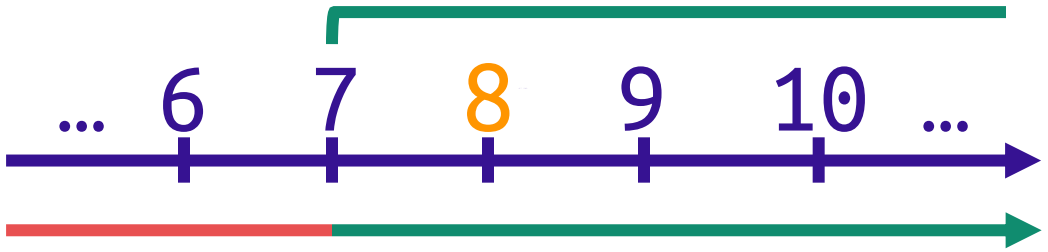



```
(
        &
        (samaccounttype>=805306367)
        (samaccounttype<=805306369)
        (!(samaccounttype=805306367))
        (!(samaccounttype=805306369))
)
```



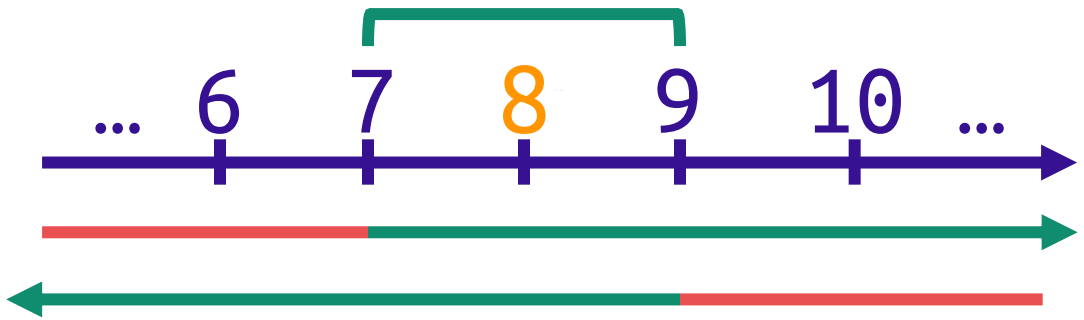
>=7
<=9
!=7
!=9

```
(
    &
    (samaccounttype>=805306367)
    (samaccounttype<=805306369)
    (!(samaccounttype=805306367))
    (!(samaccounttype=805306369))
)
```



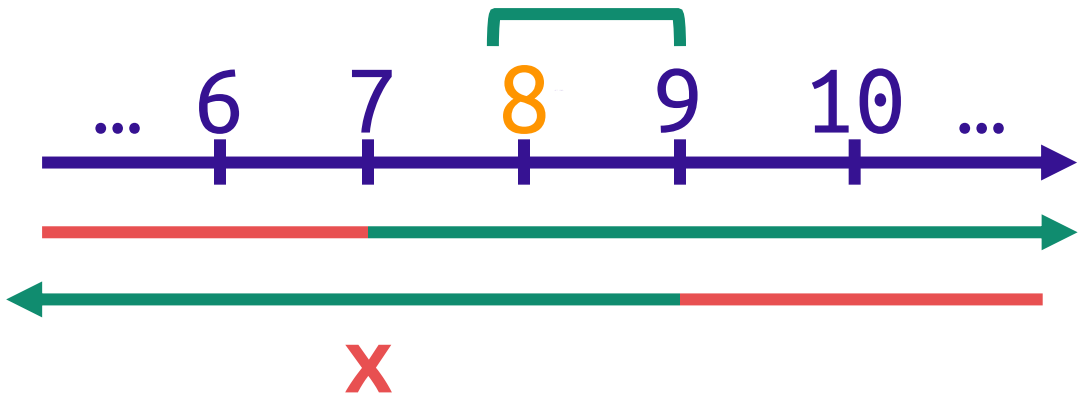
>=7
<=9
!=7
!=9

```
(
  &
  (samaccounttype>=805306367)
  (samaccounttype<=805306369)
  (!(samaccounttype=805306367))
  (!(samaccounttype=805306369))
)
```



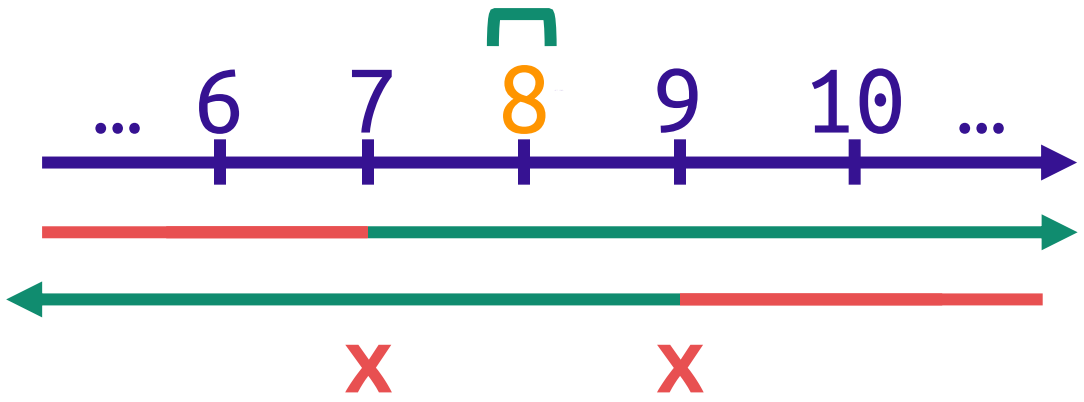
>=7
<=9
!=7
!=9

```
(
  &
  (samaccounttype>=805306367)
  (samaccounttype<=805306369)
  (!(samaccounttype=805306367))
  (!(samaccounttype=805306369))
)
```



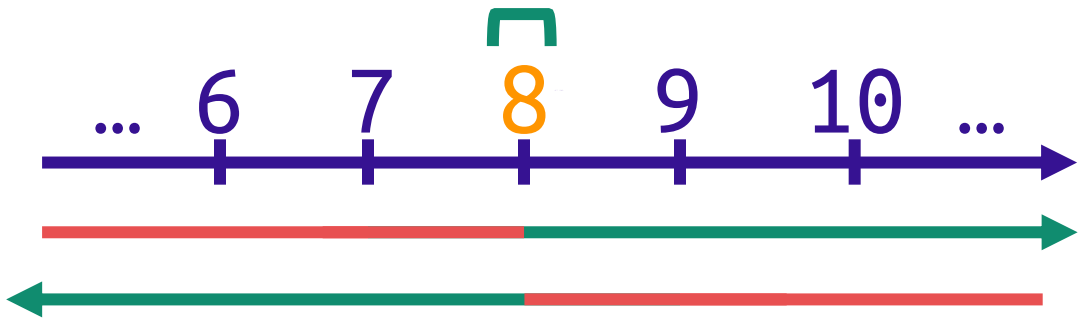
>=7
<=9
!=7
!=9

```
(
  &
  (samaccounttype>=805306367)
  (samaccounttype<=805306369)
  (!(samaccounttype=805306367))
  (!(samaccounttype=805306369))
)
```



>=7
<=9
!=7
!=9

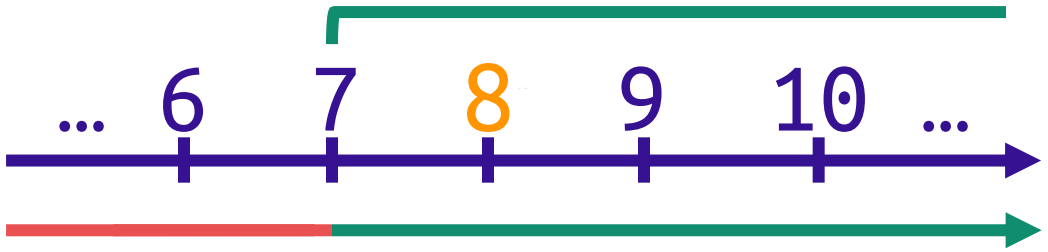
```
(
&
(samaccounttype>=805306367)
(samaccounttype<=805306369)
(!(samaccounttype=805306367))
(!(samaccounttype=805306369))
)
```



>=8	>=7
<=8	<=9
	!=7
	!=9

```
(
  &
  (samaccounttype>=805306367)
  (samaccounttype<=805306369)
  (!(samaccounttype=805306367))
  (!(samaccounttype=805306369))
)
```

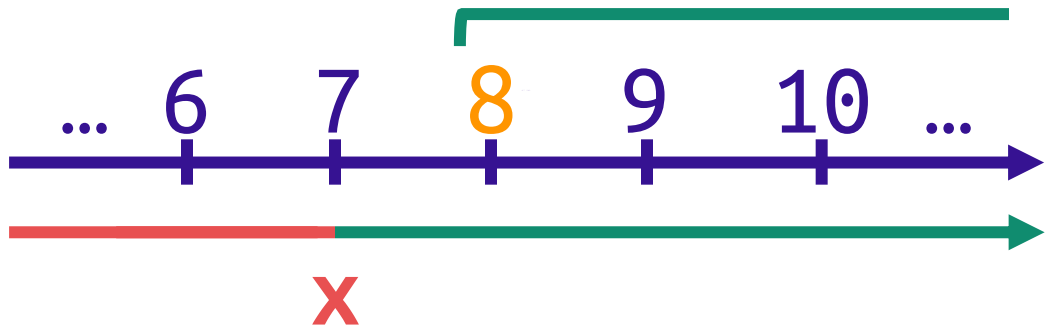
>=7
!=7
!>=9



>=8	>=7
<=8	<=9
	!=7
	!=9

```
(
  &
  (samaccounttype>=805306367)
  (samaccounttype<=805306369)
  (!(samaccounttype=805306367))
  (!(samaccounttype=805306369))
)
```

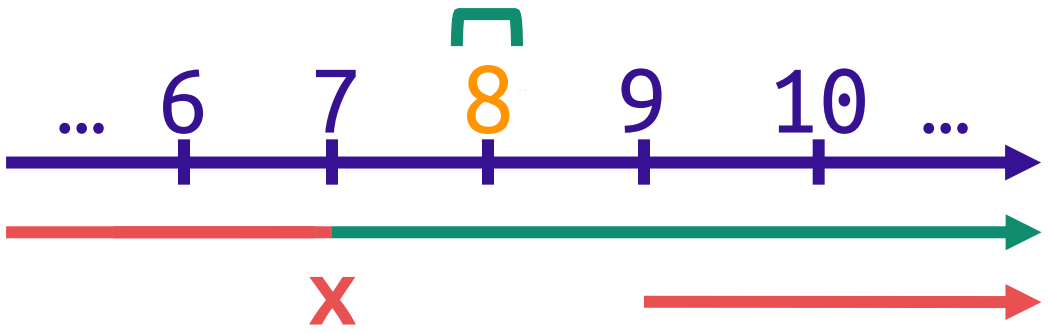
>=7
!=7
!>=9



>=8	>=7
<=8	<=9
	!=7
	!=9

```
(
  &
  (samaccounttype>=805306367)
  (samaccounttype<=805306369)
  (!(samaccounttype=805306367))
  (!(samaccounttype=805306369))
)
```

>=7
!=7
!>=9



>=8	>=7
<=8	<=9
	!=7
	!=9

```
(
  &
  (samaccounttype>=805306367)
  (samaccounttype<=805306369)
  (!(samaccounttype=805306367))
  (!(samaccounttype=805306369))
)
```

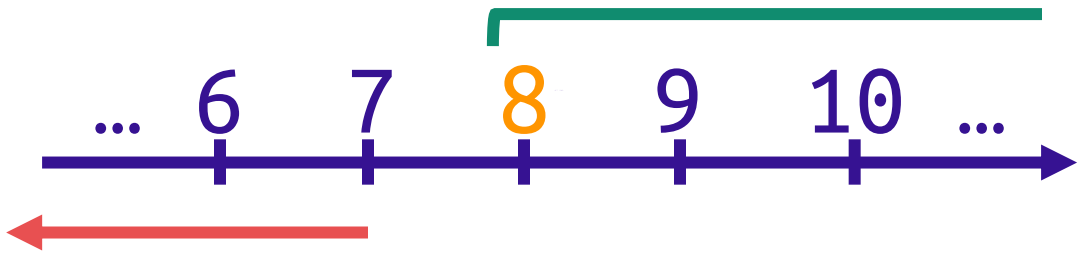
>=7	=*
!=7	!<=7
!>=9	!>=9



>=8	>=7
<=8	<=9
	!=7
	!=9

```
(
    &
    (samaccounttype>=805306367)
    (samaccounttype<=805306369)
    (!(samaccounttype=805306367))
    (!(samaccounttype=805306369))
)
```

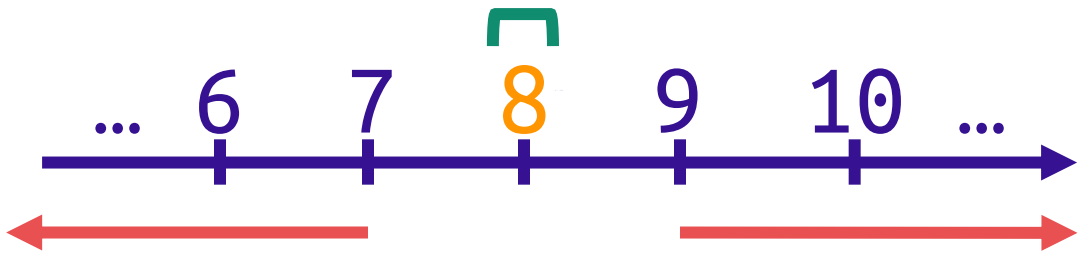
>=7	=*
!=7	!<=7
!>=9	!>=9



>=8	>=7
<=8	<=9
	!=7
	!=9

```
(
    &
    (samaccounttype>=805306367)
    (samaccounttype<=805306369)
    (!(samaccounttype=805306367))
    (!(samaccounttype=805306369))
)
```

>=7	=*
!=7	!<=7
!>=9	!>=9

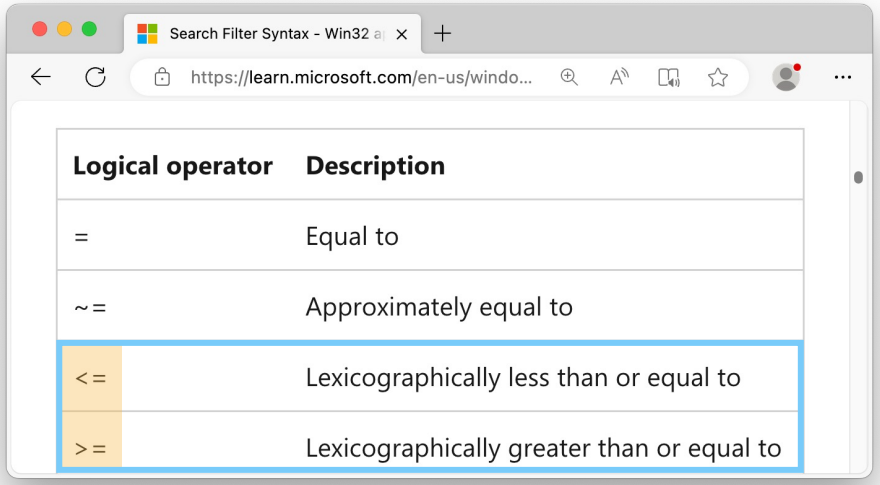


>=8	>=7
<=8	<=9
	!=7
	!=9

```
(
    &
    (samaccounttype>=805306367)
    (samaccounttype<=805306369)
    (!(samaccounttype=805306367))
    (!(samaccounttype=805306369))
)
```

Obfuscation::Filter::ComparisonOperator

- ComparisonOperator obfuscation
  - Approximately equal to
  - Logically equivalent presence filter
  - Logically equivalent range filters




```
(
    &
    (samaccounttype>=805306367)
    (samaccounttype<=805306369)
    (!(samaccounttype=805306367))
    (!(samaccounttype=805306369))
)
```

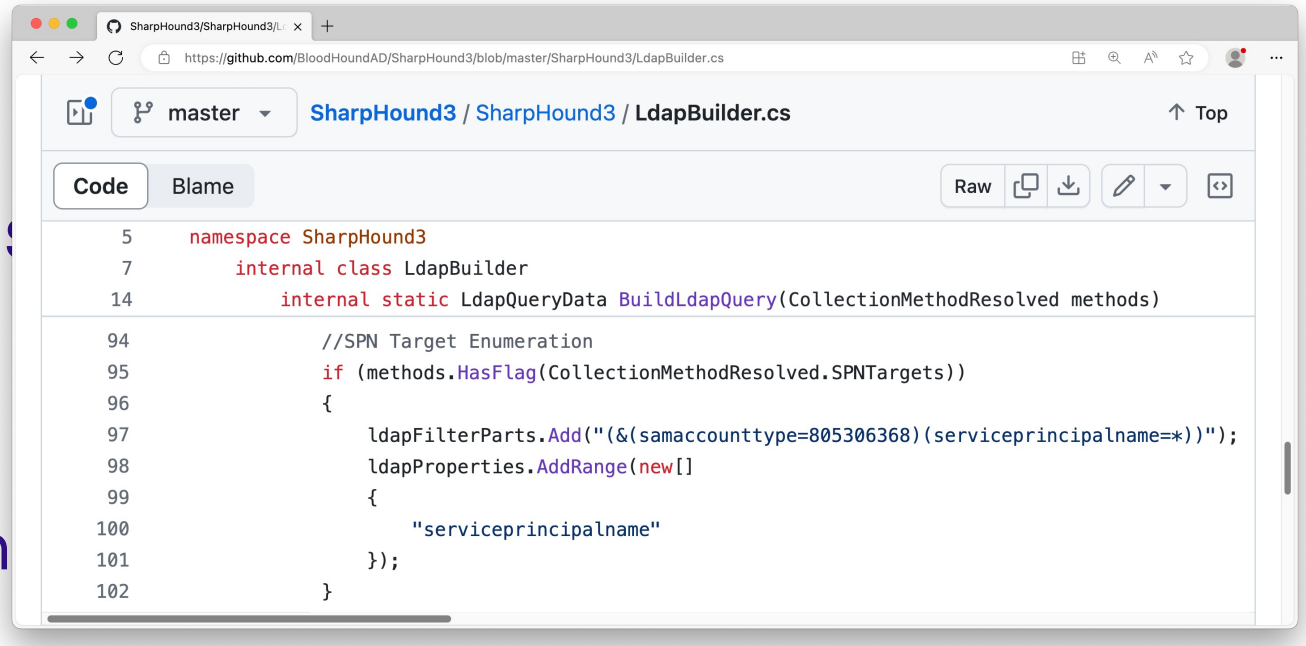
```
(
    &
    (name>=Domain Ad)
    (name<=Domain Ae)
    (name=*mins)
)
```

Obfuscation::Filter::ComparisonOperator

- ComparisonOperator obfuscation
  - Approximately equal to
    - `(name=Domain Admins)`
    - `(name~=Domain Admins)`
  - Logically equivalent presence filter
    - `(servicePrincipalName=*)`
    - `(servicePrincipalName>==)`
    - `(servicePrincipalName>=!)`
    - `(servicePrincipalName<=zzz)`
  - Logically equivalent range filters
    - `(samaccounttype=805306368)`
    - `(&(samaccounttype>=805306367)(samaccounttype<=805306369))`
    - `(&(name>=Domain Ad)(name<=Domain Ae))`

Obfuscation::Filter::ComparisonOperator

- ComparisonOperator obfu
  - Approximately equal to
    - (name=Domain Admins)
    - (name~=Domain Admins)
  - Logically equivalent presen
    - (servicePrincipalName=*)
    - (servicePrincipalName>==)
    - (servicePrincipalName>=!)
    - (servicePrincipalName<=zzz)
  - Logically equivalent range filters
    - (samaccounttype=805306368)
    - (&(samaccounttype>=805306367)(samaccounttype<=805306369))
    - (&(name>=Domain Ad)(name<=Domain Ae))



BLOODHOUND

Obfuscation::Filter::ComparisonOperator

```
//SPN Target Enumeration
if (methods.HasFlag(CollectionMethodResolved.SPNTargets))
{                                (samaccounttype=805306368)(servicePrincipalName=*)
    ldapFilterParts.Add("(&(samaccounttype=805306368)(serviceprincipalname=*))");
    ldapProperties.AddRange(new[]
    {
        "serviceprincipalname"
    });
}
```

- Logically equivalent range filters

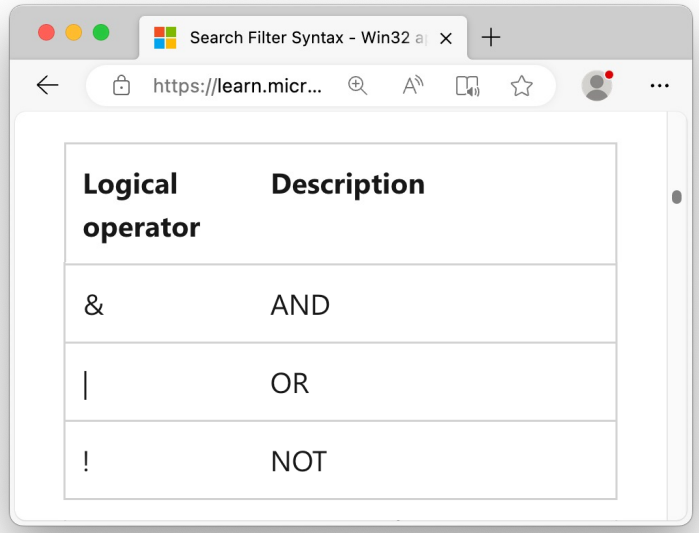


Obfuscation::Filter::BooleanOperator

AND → &

OR → |





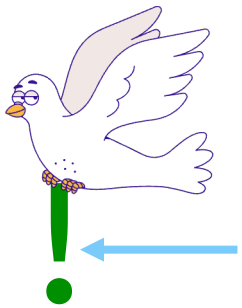
Logical operator	Description
&	AND
|	OR
!	NOT

Obfuscation::Filter::BooleanOperator

AND → &

OR → |





! ← NOT

Search Filter Syntax - Win32 a

https://learn.micr...

Logical operator	Description
&	AND
|	OR
!	NOT

Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive

```
(co=Albania)
```



Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive

```
(|(|(&(((((((|(|(co=Albania))))))))))))
```



Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
  - Double negation

```
(|(&(!(!(|(|(co=Albania))))))))))
```



Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
  - Double negation




```
(|(&(!(!(|(|(co=Albania))))))))))
```

```
(!(!(l=Kukës)))
```

```
(!(!l=Kukës))
```

Active Directory: LDAP Syntax

https://learn.microsoft.com/en-us/archive/technet-wiki/5392....

20. According to RFC 2254, the NOT operator, "!", should operate on a clause in parentheses (similar to the operators | and &). Although a clause similar to (!cn=*Smith) works in almost all cases, it would be more correct to use (!(cn=*Smith)).

Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
  - Double negation



```
(
   &
   (co=Albania)
   (
      |
      (l=Kukës)
      (l=Tiranë)
   )
)
```

Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
  - Double negation



```
(
  &1
  (co=Albania)
  (  2
     |
     (l=Kukës)
     (l=Tiranë)
  )
)
```

Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
  - Double negation

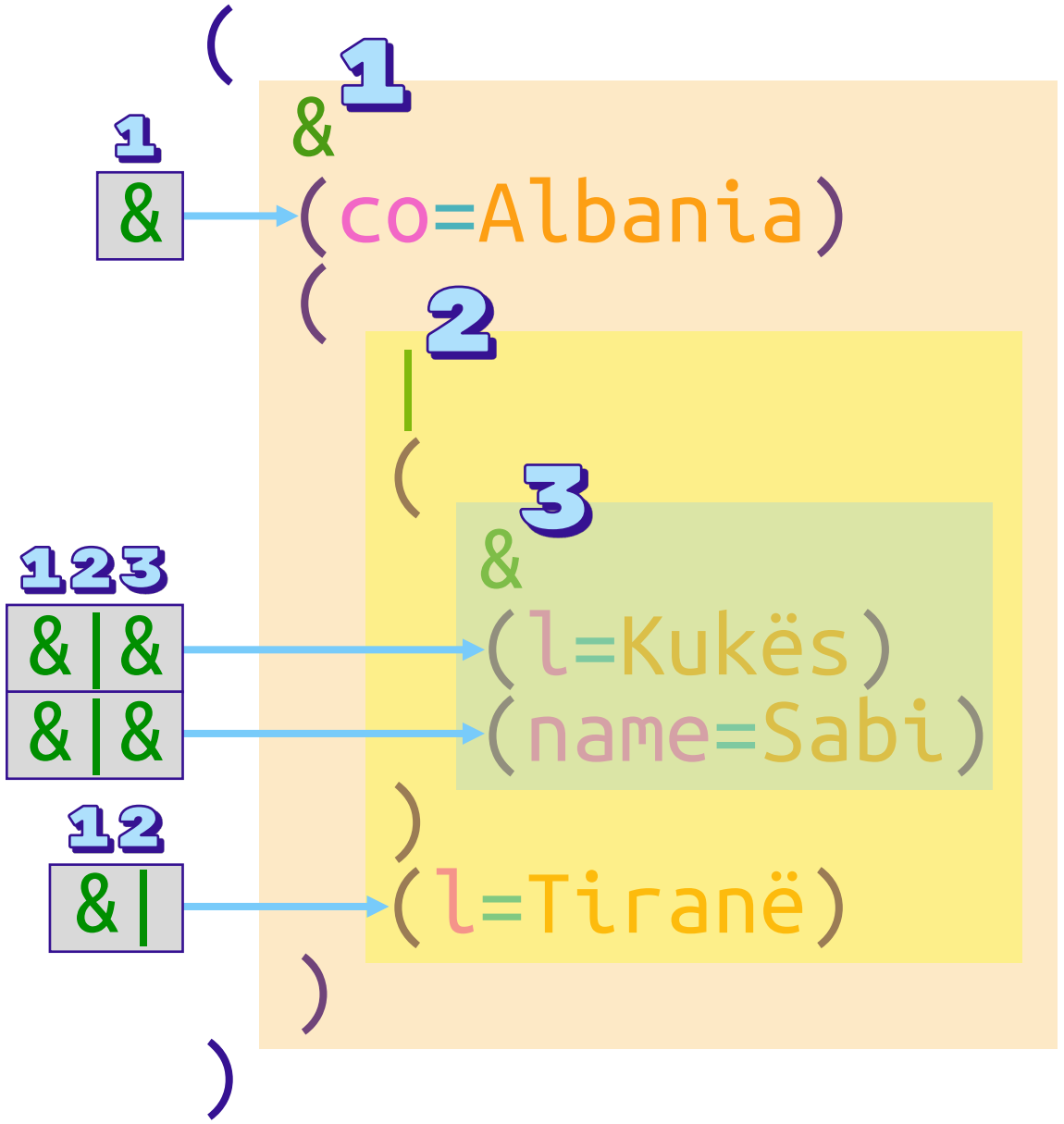


```
(
  &1
  (co=Albania)
  (
    2
    |
    (
      3
      &
      (l=Kukës)
      (name=Sabi)
    )
    (l=Tiranë)
  )
)
```

Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
  - Double negation

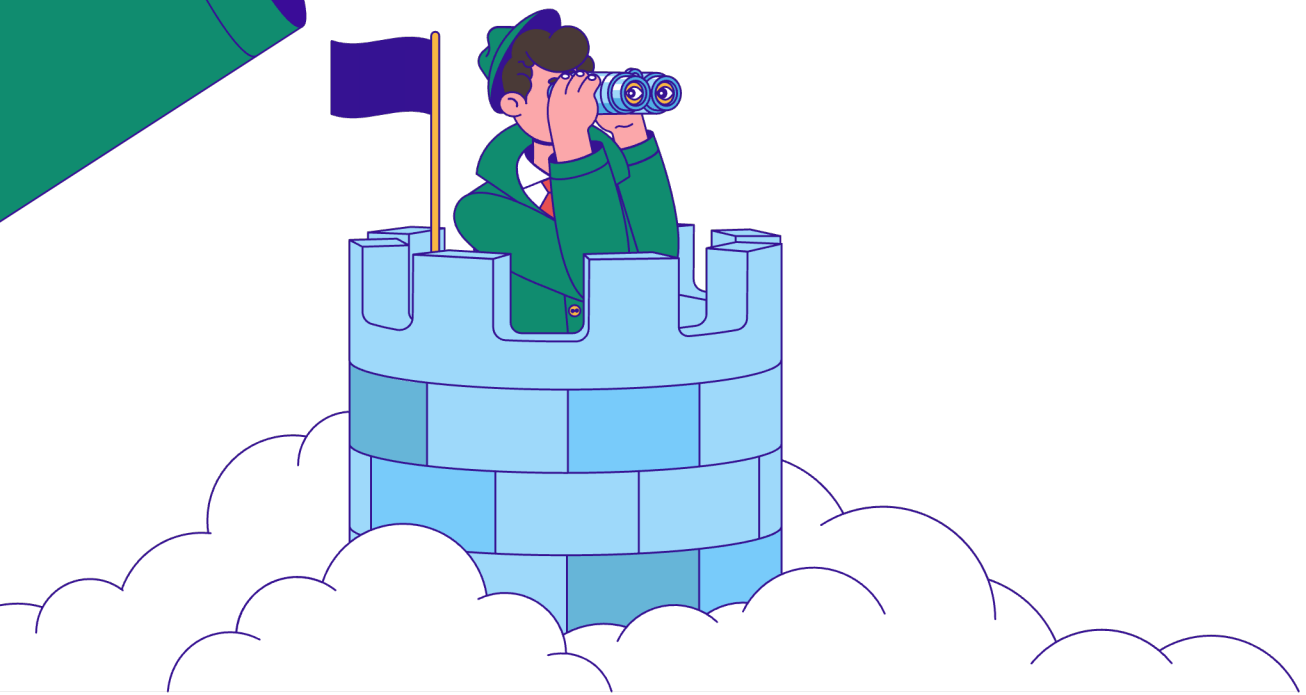




Obfuscation::Filter::BooleanOperator

- Bool

! ONLY affects 1st BooleanOperator OR Filter (not FilterList).



```
(
  1 &
  (co=Albania)
  (
    2 !
    (
      (l=Kukës)
      (name=Sabi)
    )
  )
)
```

1 &

1 2 &!

&

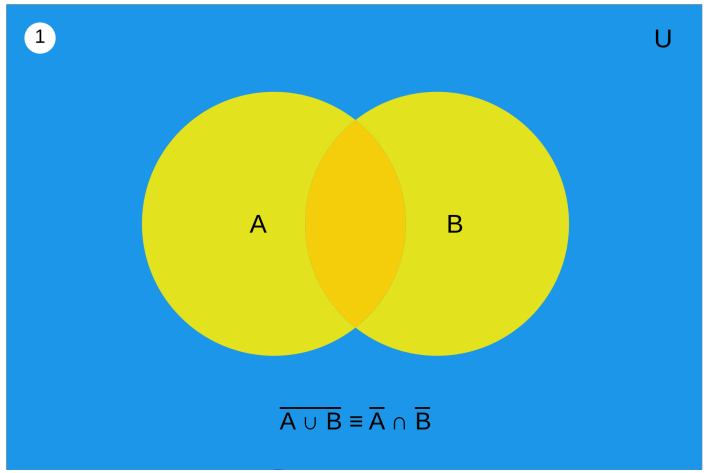
1 &

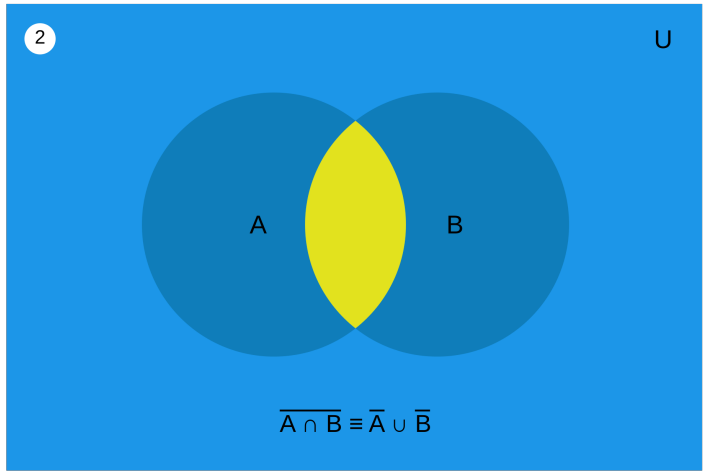
Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
  - Double negation
  - Logical inversion

```
(
   &
   (co=Albania)
   (
      |
      (l=Kukës)
      (l=Tiranë)
   )
)
```

Obfuscation::Filter::BooleanOperator





obfuscation

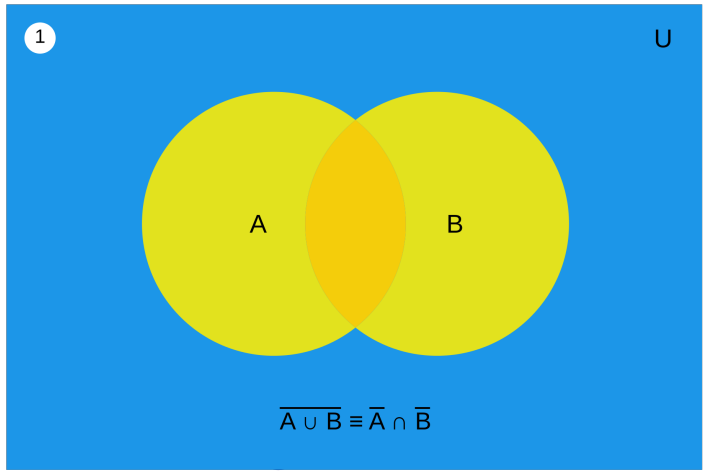
(De Morgan's laws)

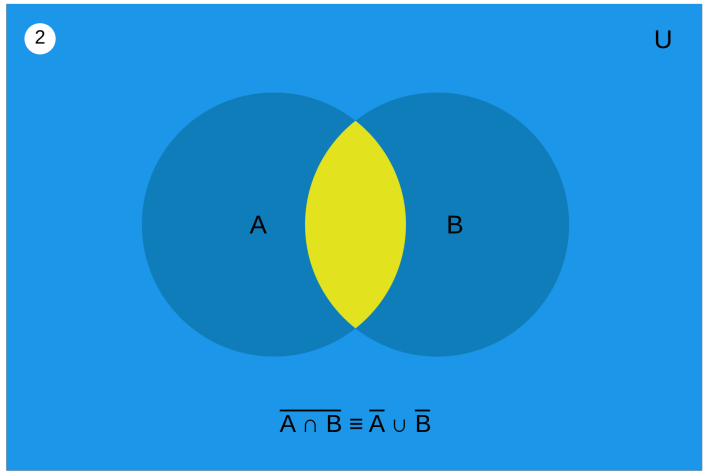
```
(
    &
    (co=Albania)
    (
        |
        (l=Kukës)
        (l=Tiranë)
    )
)
```

not (A or B) = (not A) and (not B)

not (A and B) = (not A) or (not B)

Obfuscation::Filter::BooleanOperator





obfuscation

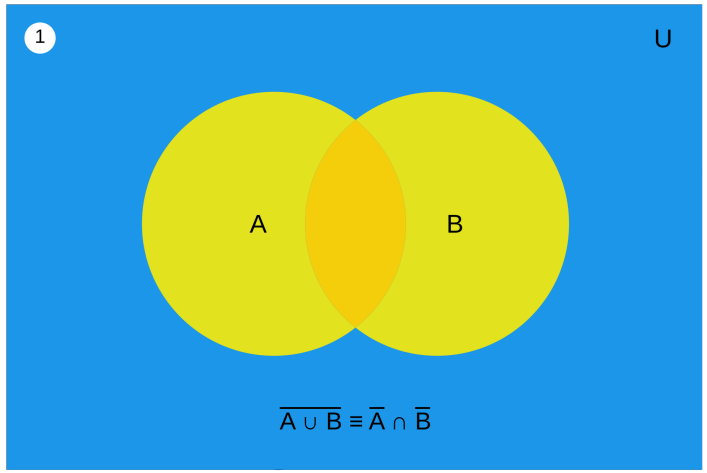
(De Morgan's laws)

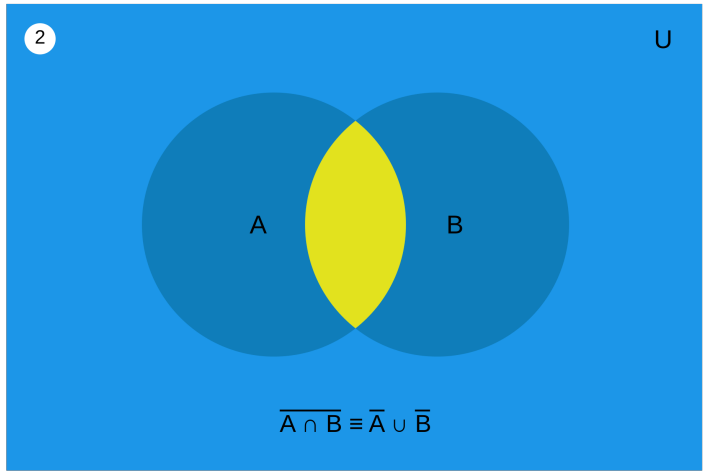
```
(
  !
1 (
        &
        (co=Albania)
2       (
          |
          (l=Kukës)
3         (l=Tiranë)
        )
      )
    )
```

not (A or B) = (not A) and (not B)

not (A and B) = (not A) or (not B)

Obfuscation::Filter::BooleanOperator





obfuscation

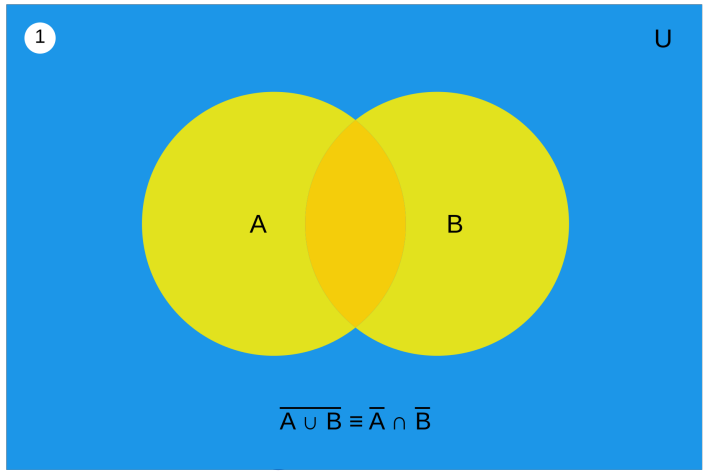
(De Morgan's laws)

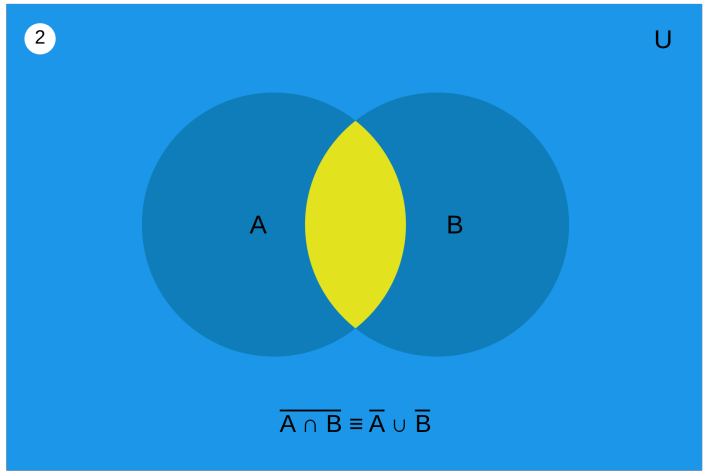
```
(
    !
1   (
        &
        (co=Albania)
2       (
            |
            (l=Kukës)
3           (l=Tiranë)
        )
    )
)
```

not (A or B) = (not A) and (not B)

not (A and B) = (not A) or (not B)

Obfuscation::Filter::BooleanOperator





obfuscation

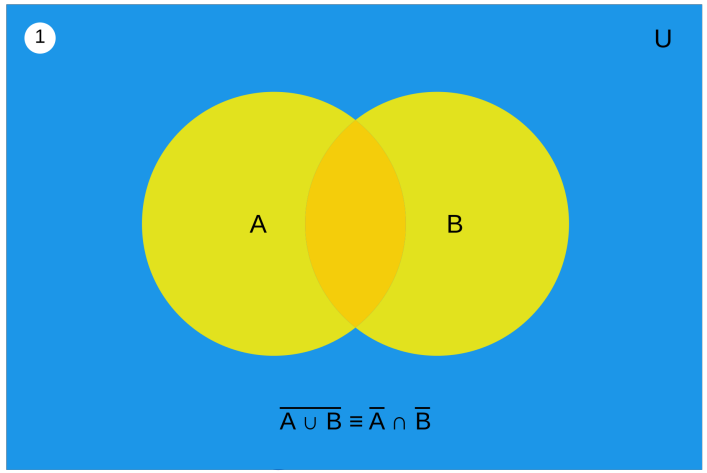
(De Morgan's laws)

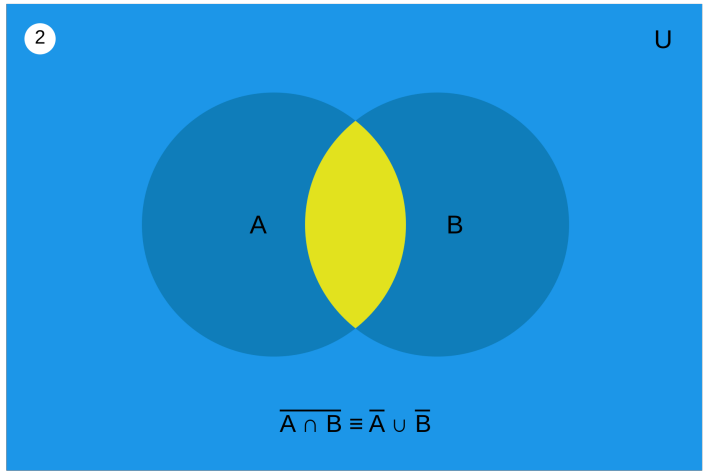
```
(
  !
1 (
    |
    (co=Albania)
2   (
      &
      (l=Kukës)
3     (l=Tiranë)
    )
  )
)
```

not (A or B) = (not A) and (not B)

not (A and B) = (not A) or (not B)

Obfuscation::Filter::BooleanOperator





obfuscation

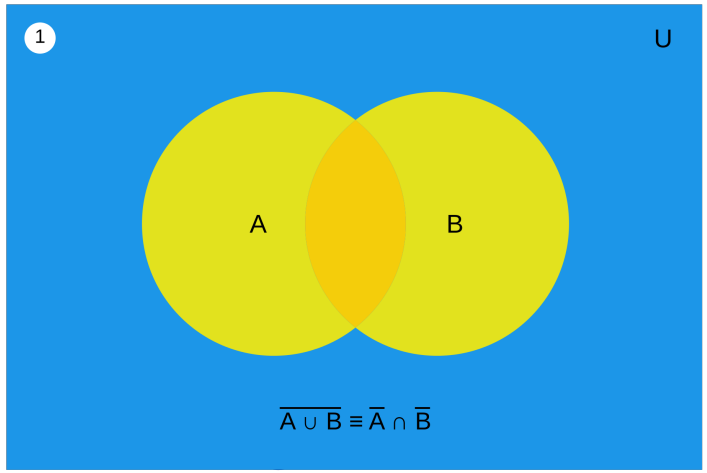
(De Morgan's laws)

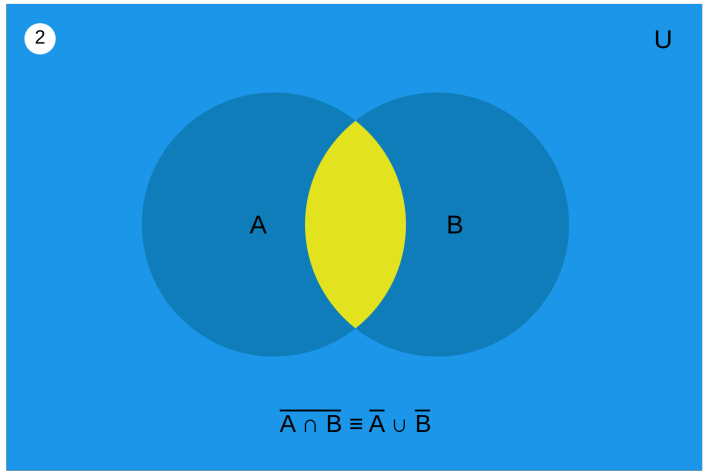
```
(
    !
1   (
        |
        (co=Albania)
2       (
            &
            (l=Kukës)
3           (l=Tiranë)
        )
    )
)
```

not (A or B) = (not A) and (not B)

not (A and B) = (not A) or (not B)

Obfuscation::Filter::BooleanOperator





obfuscation

(De Morgan's laws)

```
(
    !
1   (
        |
        (!co=Albania)
2       (
            &
            (!l=Kukës)
3           (!l=Tiranë)
        )
    )
)
```

not (A or B) = (not A) and (not B)

not (A and B) = (not A) or (not B)

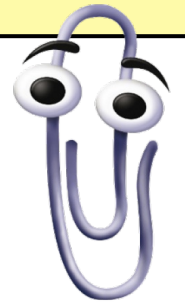
Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
    - `(co=Albania)`
    - `(|(&(((((|(|(co=Albania))))))))))`
  - Double negation
    - `(|(&((!((!(|(|(co=Albania))))))))))`
    - `(!(!l=Kukës))`
  - Logical inversion (De Morgan's laws)
    - `(&(co=Albania)(|(l=Kukës)(l=Tiranë)))`
    - `(!(|(!co=Albania)(&(!l=Kukës)(!l=Tiranë))))`

Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
    - (cn=Albanla)
    - (&(&(&(&(&(&(&(&(&(cn=Albanla))))))))))
  - Double Negation
    - (!(!(&(!(!(!(!(cn=Albanla)))))))))
    - (&(l=Kubds))
  - Logical Inversion (De Morg
    - (&(cn=Albanla)(!(l=Kubds)(!
    - (&(&(cn=Albanla)(&(l=Kubds)(&(

What is my LEAST favorite LDAP token type?

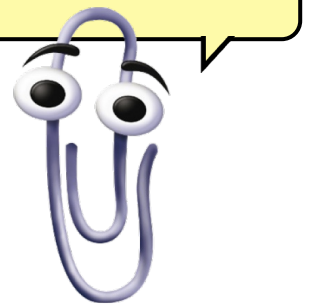


Obfuscation::Filter::BooleanOperator

- BooleanOperator obfuscation
  - Additive
    - (cn=Albania)
    - (!(!(&(!(!(cn=Albania))))))
  - Double Negation
    - (!(!(&(!(!(!(!(cn=Albania))))))))
    - (&(l=Kubds))
  - Logical Inversion (De Morg
    - (&(cn=Albania)(!(l=Kubds)(!
    - (&(|(cn=Albania)(&(l=Kubds)(&

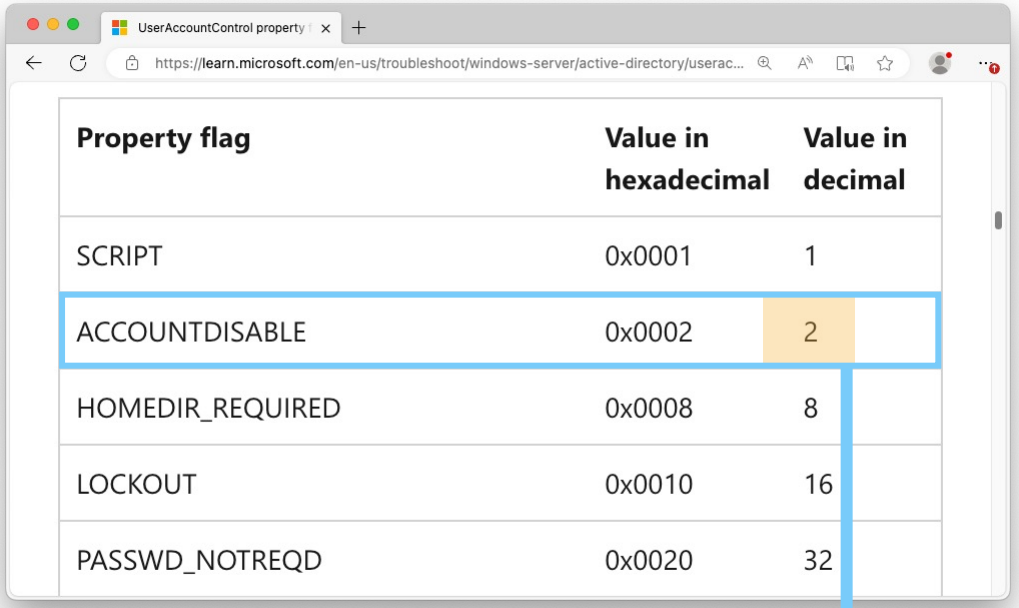
What is my LEAST favorite LDAP token type?

The BooleanOperator – because I'm afraid of getting tied up in NOTs (!)



Obfuscation::Filter::ExtensibleMatchFilter

- ExtensibleMatchFilter obfuscation



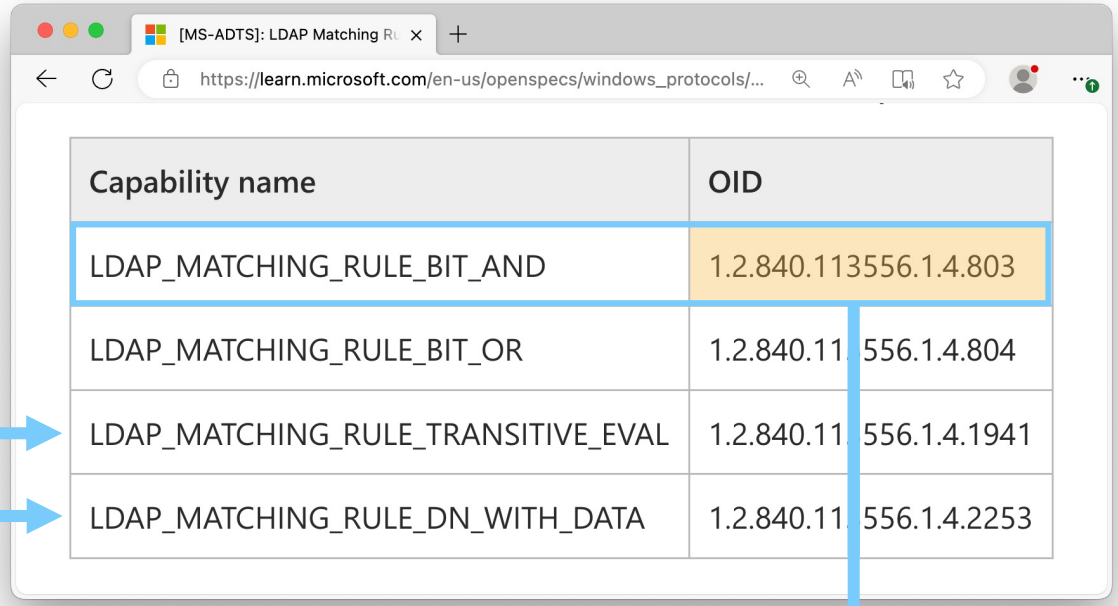
Property flag	Value in hexadecimal	Value in decimal
SCRIPT	0x0001	1
ACCOUNTDISABLE	0x0002	2
HOMEDIR_REQUIRED	0x0008	8
LOCKOUT	0x0010	16
PASSWD_NOTREQD	0x0020	32

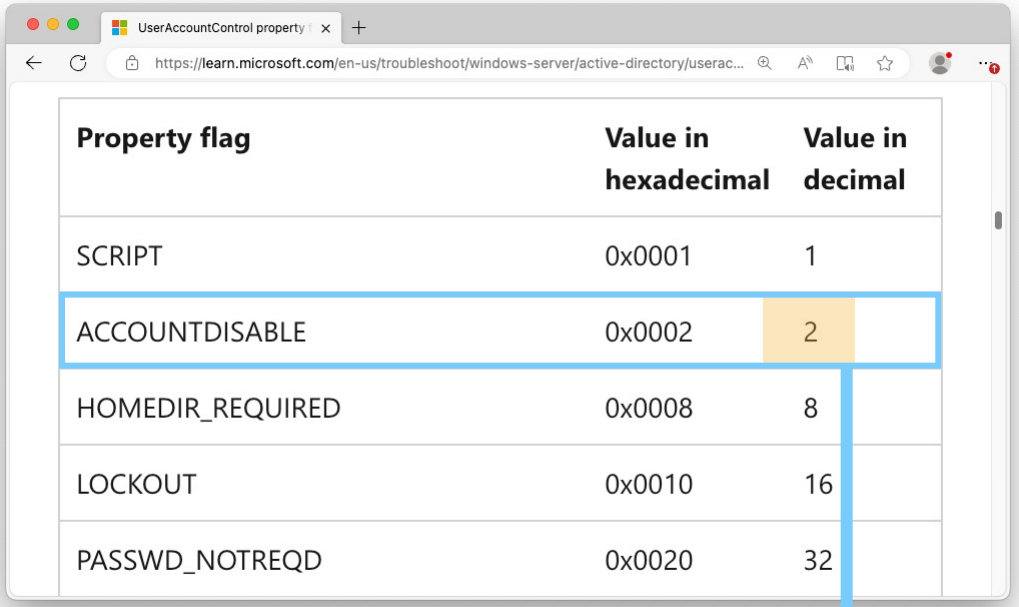
"Object is NOT disabled."

```
(!(userAccountControl:1.2.840.113556.1.4.803:=2))
```

Obfuscation::Filter::ExtensibleMatchFilter

- ExtensibleMatchFilter obfuscation



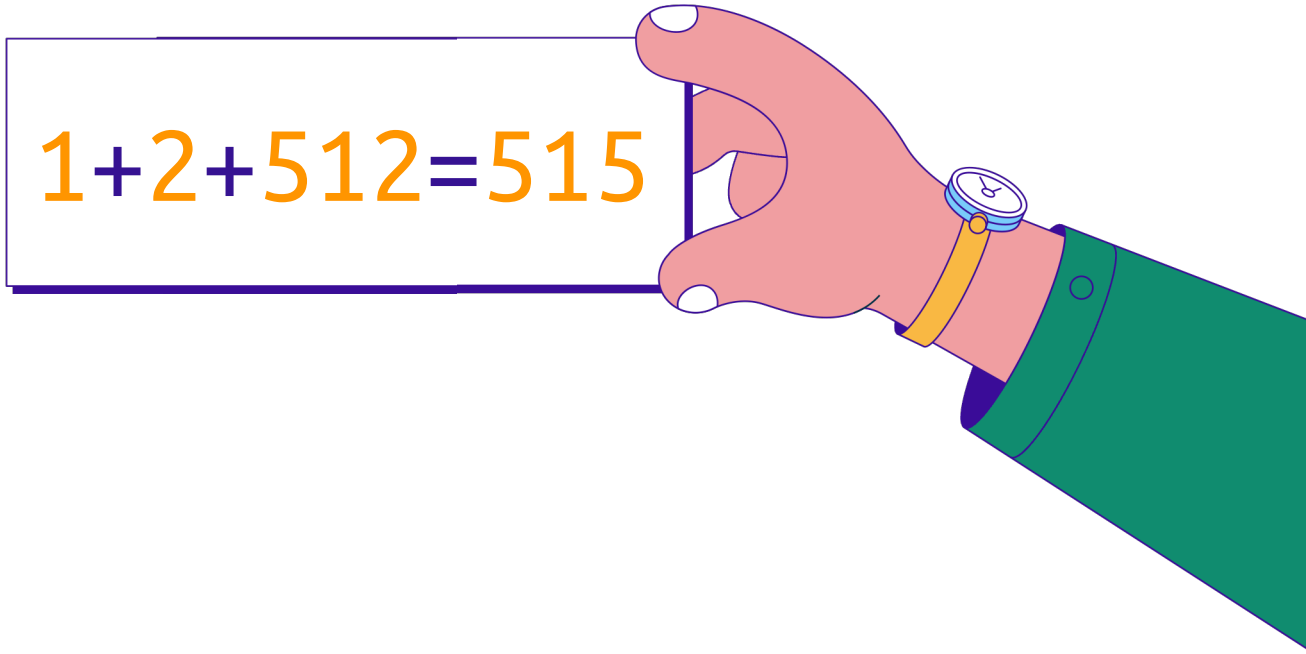


"Object is NOT disabled."

```
(!(userAccountControl:1.2.840.113556.1.4.803:=2))
```

Obfuscation::Filter::ExtensibleMatchFilter

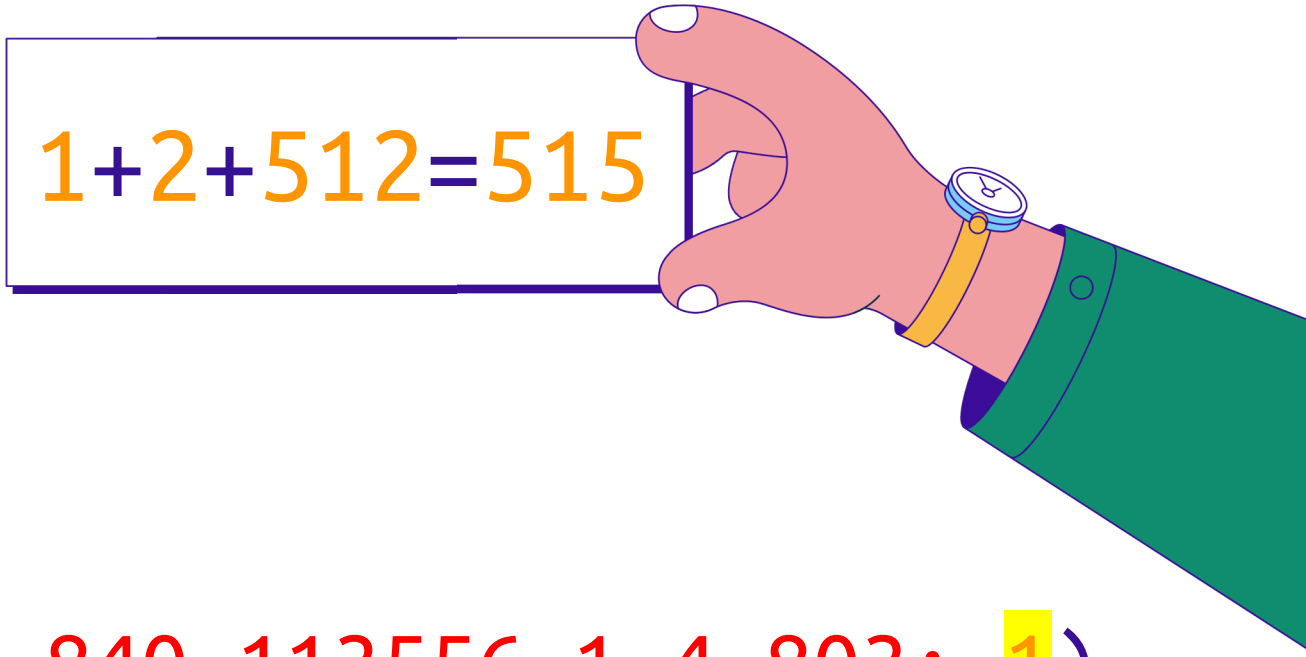
- ExtensibleMatchFilter obfuscation
  - Bitwise breakout (AND/OR)




```
(userAccountControl:1.2.840.113556.1.4.803:=515)
```

Obfuscation::Filter::ExtensibleMatchFilter

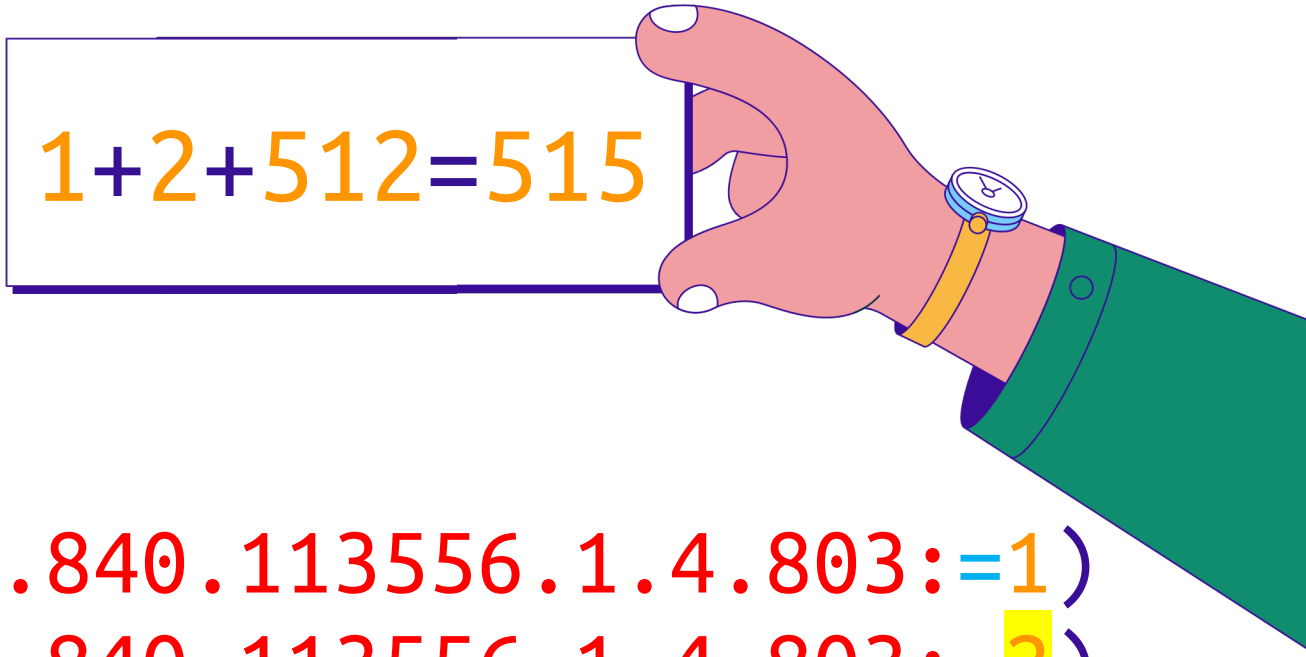
- ExtensibleMatchFilter obfuscation
  - Bitwise breakout (AND/OR)




```
(
   &
   (userAccountControl:1.2.840.113556.1.4.803:=1)
   (userAccountControl:1.2.840.113556.1.4.803:=514)
)
```

Obfuscation::Filter::ExtensibleMatchFilter

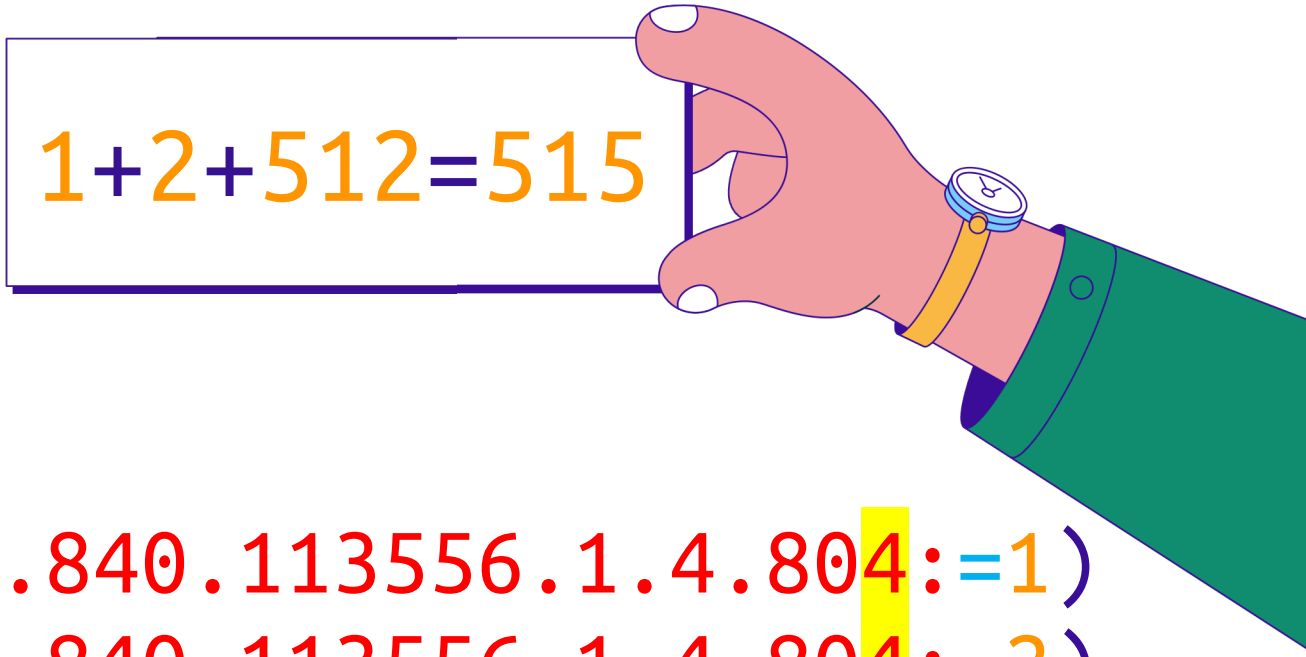
- ExtensibleMatchFilter obfuscation
  - Bitwise breakout (AND/OR)




```
(
   &
   (userAccountControl:1.2.840.113556.1.4.803:=1)
   (userAccountControl:1.2.840.113556.1.4.803:=2)
   (userAccountControl:1.2.840.113556.1.4.803:=512)
)
```

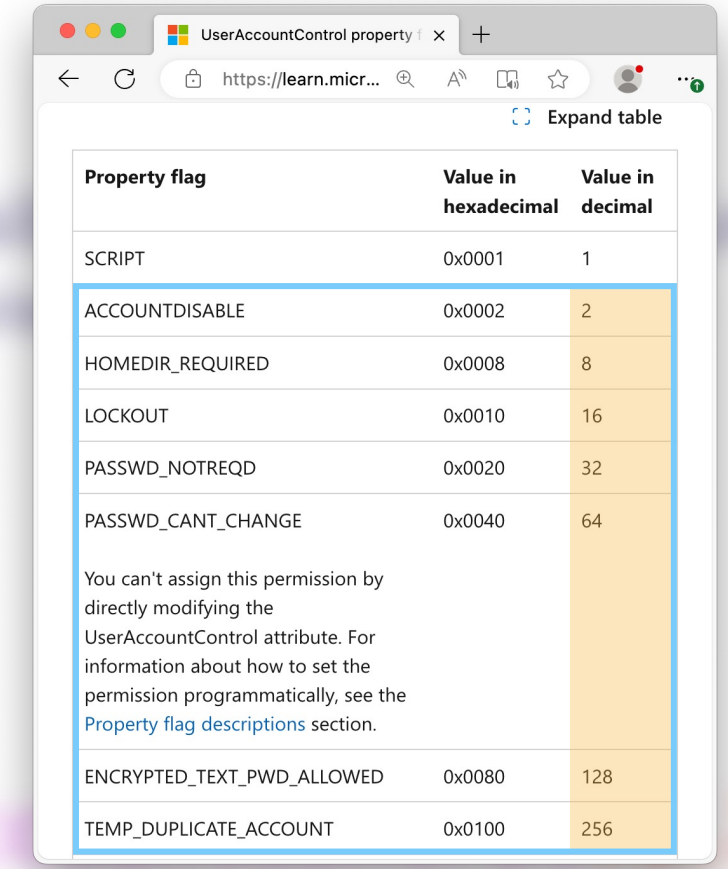
Obfuscation::Filter::ExtensibleMatchFilter

- ExtensibleMatchFilter obfuscation
  - Bitwise breakout (AND/OR)

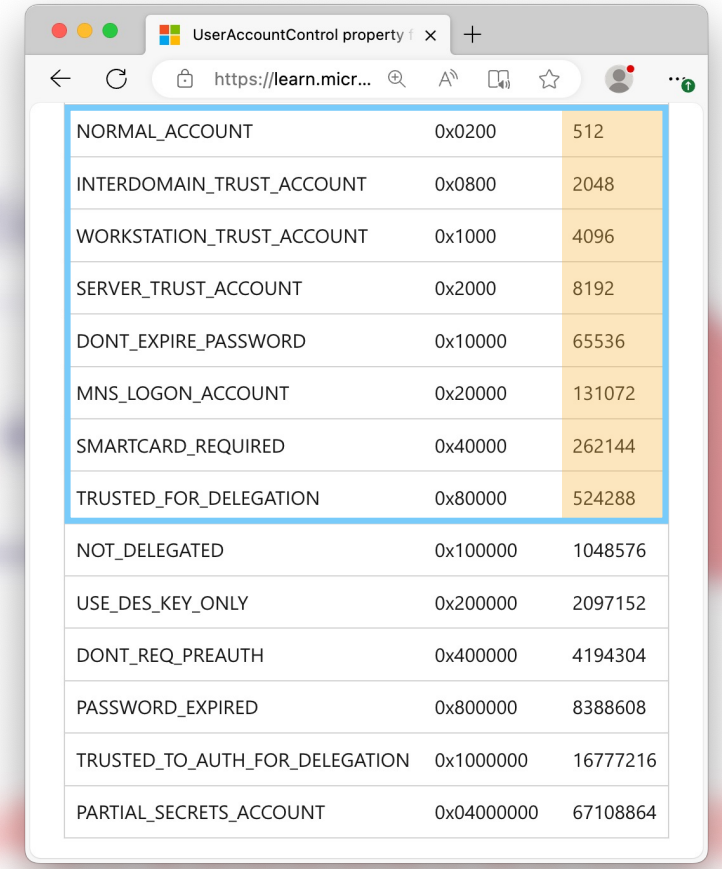



```
(
   |
   (userAccountControl:1.2.840.113556.1.4.804:=1)
   (userAccountControl:1.2.840.113556.1.4.804:=2)
   (userAccountControl:1.2.840.113556.1.4.804:=512)
)
```

Obfuscation::Filter::ExtensibleMatchFilter



Property flag	Value in hexadecimal	Value in decimal
SCRIPT	0x0001	1
ACCOUNTDISABLE	0x0002	2
HOMEDIR_REQUIRED	0x0008	8
LOCKOUT	0x0010	16
PASSWD_NOTREQD	0x0020	32
PASSWD_CANT_CHANGE You can't assign this permission by directly modifying the UserAccountControl attribute. For information about how to set the permission programmatically, see the Property flag descriptions section.	0x0040	64
ENCRYPTED_TEXT_PWD_ALLOWED	0x0080	128
TEMP_DUPLICATE_ACCOUNT	0x0100	256



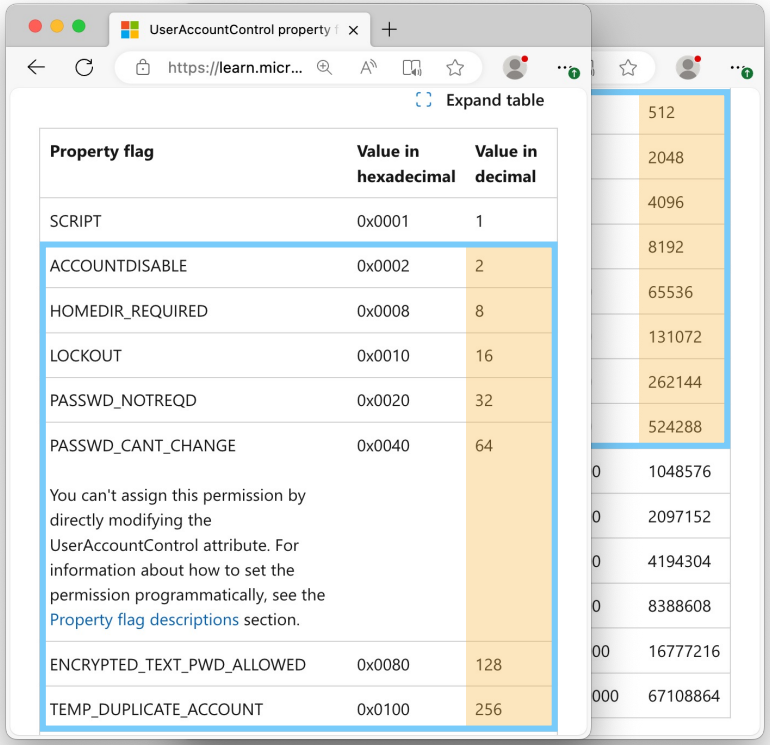
NORMAL_ACCOUNT	0x0200	512
INTERDOMAIN_TRUST_ACCOUNT	0x0800	2048
WORKSTATION_TRUST_ACCOUNT	0x1000	4096
SERVER_TRUST_ACCOUNT	0x2000	8192
DONT_EXPIRE_PASSWORD	0x10000	65536
MNS_LOGON_ACCOUNT	0x20000	131072
SMARTCARD_REQUIRED	0x40000	262144
TRUSTED_FOR_DELEGATION	0x80000	524288
NOT_DELEGATED	0x100000	1048576
USE_DES_KEY_ONLY	0x200000	2097152
DONT_REQ_PREAUTH	0x400000	4194304
PASSWORD_EXPIRED	0x800000	8388608
TRUSTED_TO_AUTH_FOR_DELEGATION	0x1000000	16777216
PARTIAL_SECRETS_ACCOUNT	0x04000000	67108864

```
Write-Verbose '[Get-DomainUser] Searching for users who can be delegated'
# negation of "Accounts that are sensitive and not trusted for delegation"
$Filter += '(!(userAccountControl:1.2.840.113556.1.4.803:=1048574))'
```

(!(userAccountControl:1.2.840.113556.1.4.803:=1048574))

Obfuscation::Filter::ExtensibleMatchFilter

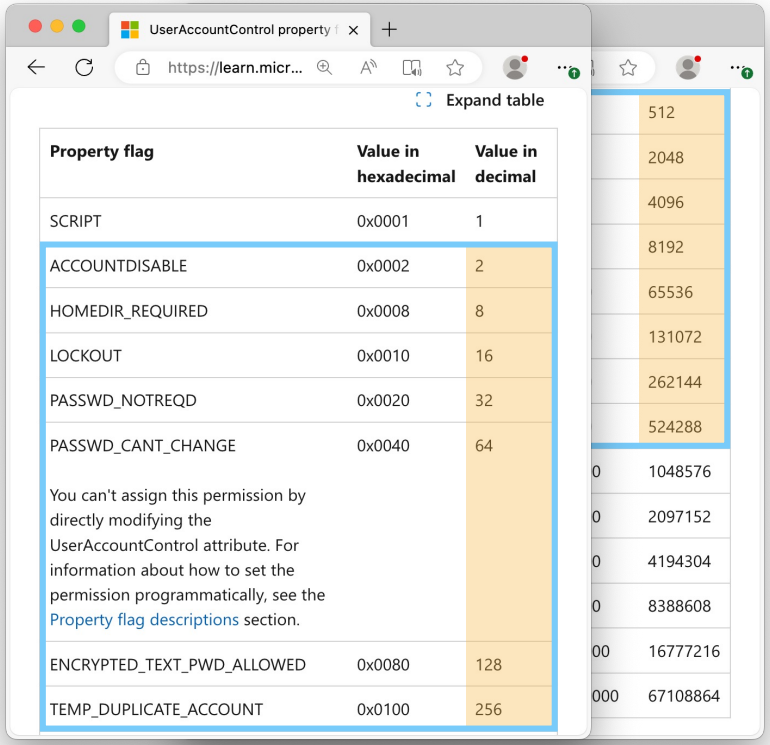
- ExtensibleMatchFilter obfuscation
  - Bitwise breakout (AND/OR)
  - Bitwise breakout (exact)
    - **AND**
      - **AND'd 1 bits**
      - **NOT OR'd 0 bits**




```
(userAccountControl=1048574)
```

Obfuscation::Filter::ExtensibleMatchFilter

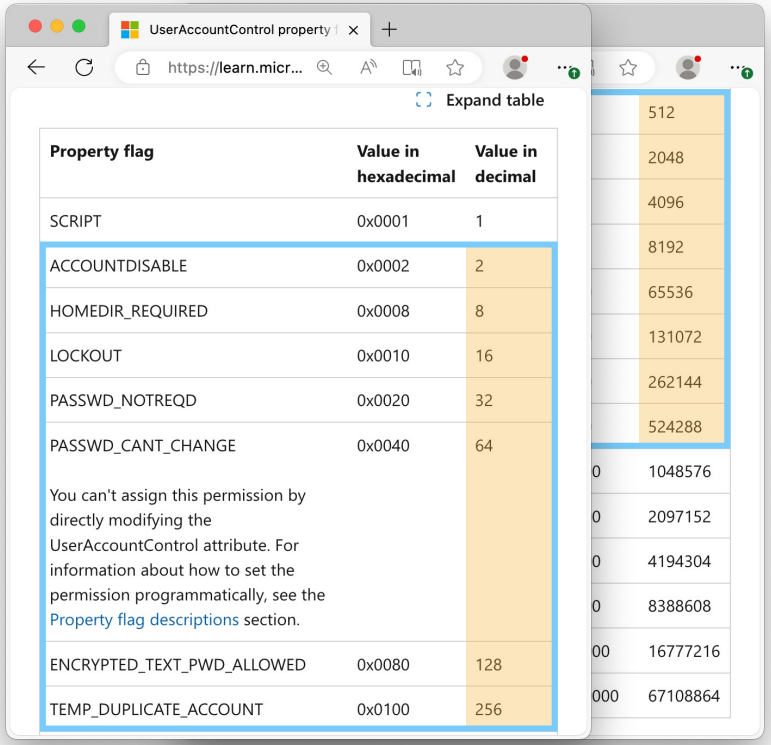
- ExtensibleMatchFilter obfuscation
  - Bitwise breakout (AND/OR)
  - Bitwise breakout (exact)
    - **AND**
      - **AND'd 1 bits**
      - **NOT OR'd 0 bits**




```
(
   &
   (userAccountControl:1.2.840.113556.1.4.803:=1048574)
   (!(userAccountControl:1.2.840.113556.1.4.804:=4293918721))
)
```

Obfuscation::Filter::ExtensibleMatchFilter

- ExtensibleMatchFilter obfuscation
  - Bitwise breakout (AND/OR)
  - Bitwise breakout (exact)
    - **AND**
      - **AND'd 1 bits**
      - **NOT OR'd 0 bits**




```
(
   &
   (userAccountControl:1.2.840.113556.1.4.803:=342294)
   (userAccountControl:1.2.840.113556.1.4.803:=706280)
   (!(userAccountControl:1.2.840.113556.1.4.804:=328204289))
   (!(userAccountControl:1.2.840.113556.1.4.804:=3965714432))
)
```

Obfuscation::Filter::ExtensibleMatchFilter

- ExtensibleMatchFilter obfuscation
  - Bitwise breakout (AND/OR)
    - `(userAccountControl:1.2.840.113556.1.4.803:=515)`
    - `(&(userAccountControl:1.2.840.113556.1.4.803:=1)`
      `(userAccountControl:1.2.840.113556.1.4.803:=514))`
  - Bitwise breakout (exact)
    - `(userAccountControl=1048574)`
    - `(&(userAccountControl:1.2.840.113556.1.4.803:=342294)`
      `(userAccountControl:1.2.840.113556.1.4.803:=706280)`
      `(!(userAccountControl:1.2.840.113556.1.4.804:=328204289))`
      `(!(userAccountControl:1.2.840.113556.1.4.804:=3965714432)))`



Obfuscation::Filter::Value

- Value obfuscation
  - Casing
    - Excluding Attributes with Value formats **Boolean**, **SID**, **HexObjectReplicaLink**, etc.

```
(name=krbtgt)
```

Obfuscation::Filter::Value

- Value obfuscation
  - Casing
    - Excluding Attributes with Value formats **Boolean**, **SID**, **HexObjectReplicaLink**, etc.

```
(name=kRBtgT)
```

Obfuscation::Filter::Value

- Value obfuscation
  - Casing
  - Prepended 0's

```
(sAMAccountName=805306368)
  (logonCount>=-1)
```

Obfuscation::Filter::Value



- Value obfuscation
  - Casing
  - Prepended 0's

```
(sAMAccountName=0000805306368)
  (logonCount>=-0000001)
```

Obfuscation::Filter::Value

- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps

```
(whenCreated=20080217112855.000Z)
```

Obfuscation::Filter::Value



- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps



(whenCreated=20080217112855.1337KaZanPlehrash)

Obfuscation::Filter::Value



- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps



(whenCreated=20080217112855.1337KaZanPlehrash)

\5A
\5a



Obfuscation::Filter::Value

- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps
  - Hex encoding

Character	Hex Representation
*	\2A
(	\28
)	\29
\	\5C
Nul	\00

Character	Hex Representation
á	\E1
é	\E9
í	\ED
ó	\F3
ú	\FA
ñ	\F1

```
(name=krbtgt)
```

Obfuscation::Filter::Value

- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps
  - Hex encoding

Character	Hex Representation
*	\2A
(	\28
)	\29
\	\5C
Nul	\00

Character	Hex Representation
á	\E1
é	\E9
í	\ED
ó	\F3
ú	\FA
ñ	\F1

```
(name=krb\74g\74)
```

Obfuscation::Filter::Value



- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps
  - Hex encoding

Character	Hex Representation
*	\2A
(	\28
)	\29
\	\5C
Nul	\00

Character	Hex Representation
á	\E1
é	\E9
í	\ED
ó	\F3
ú	\FA
ñ	\F1

(name=krb\74g\74)

In LDAP filters these characters should be escaped with the backslash escape character, followed by the **two character** ASCII hexadecimal representation of the character.

\00 → \0

\0A → \A

Obfuscation::Filter::Value

- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps
  - Hex encoding
  - Wildcards

```
(name=krbtgt)
```

Obfuscation::Filter::Value

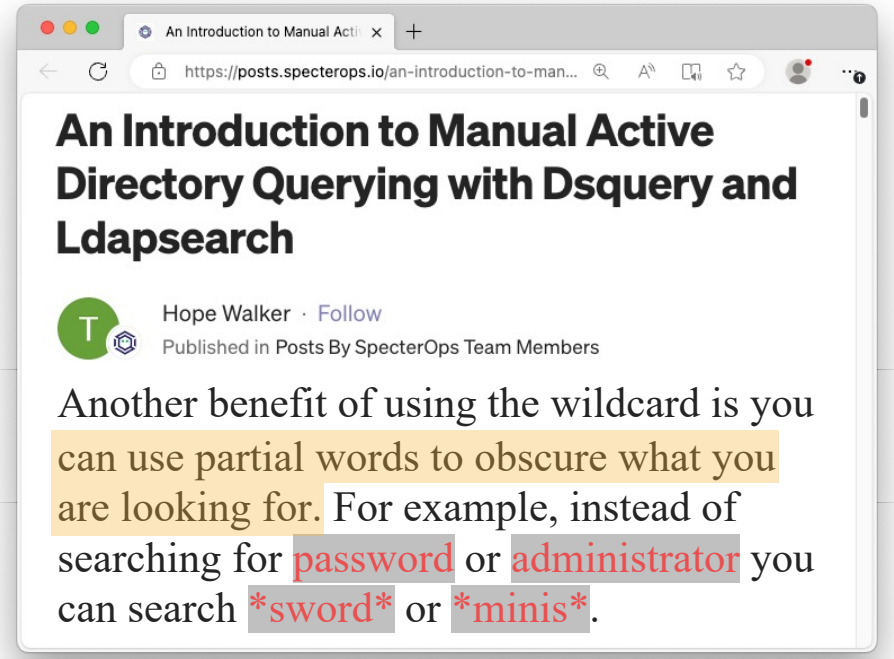
- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps
  - Hex encoding
  - Wildcards

```
(name=*rb*g*)
```



Obfuscation::Filter::Value

- Value obfuscation
  - Casing
  - Prepended 0's
  - Timestamps
  - Hex encoding
  - Wildcards




```
(name=*rb*g*)
```



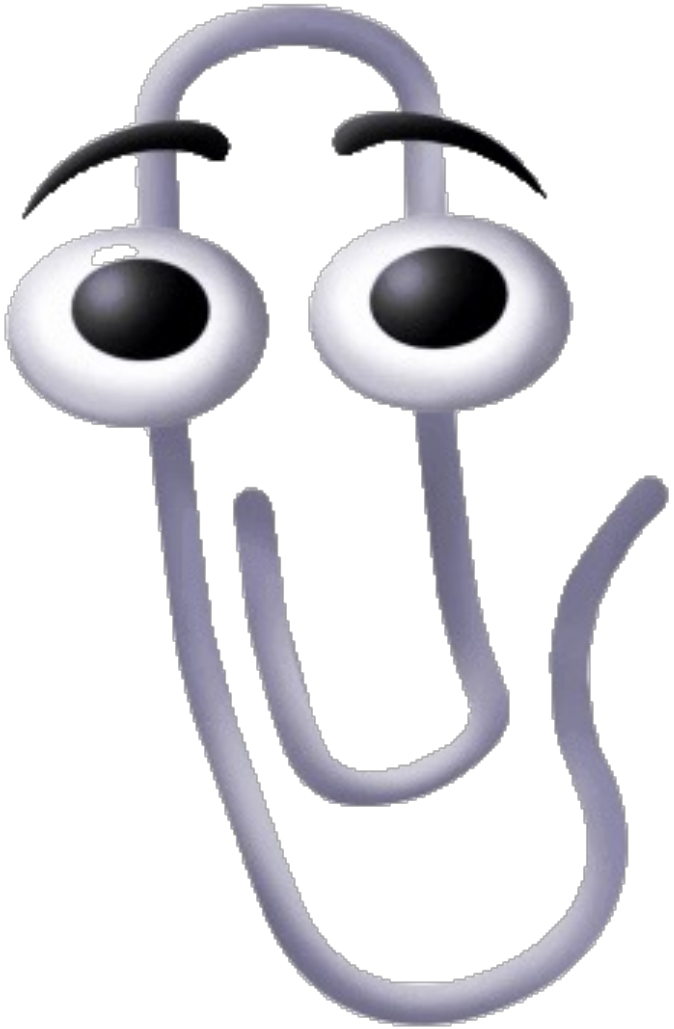
- **Hope Walker** (@Icemoonhsv) - SpecterOps
  - https://posts.specterops.io/an-introduction-to-manual-active-directory-querying-with-dsquery-and-ldapsearch-84943c13d7eb
  - https://posts.specterops.io/manual-ldap-querying-part-2-8a65099e12e3

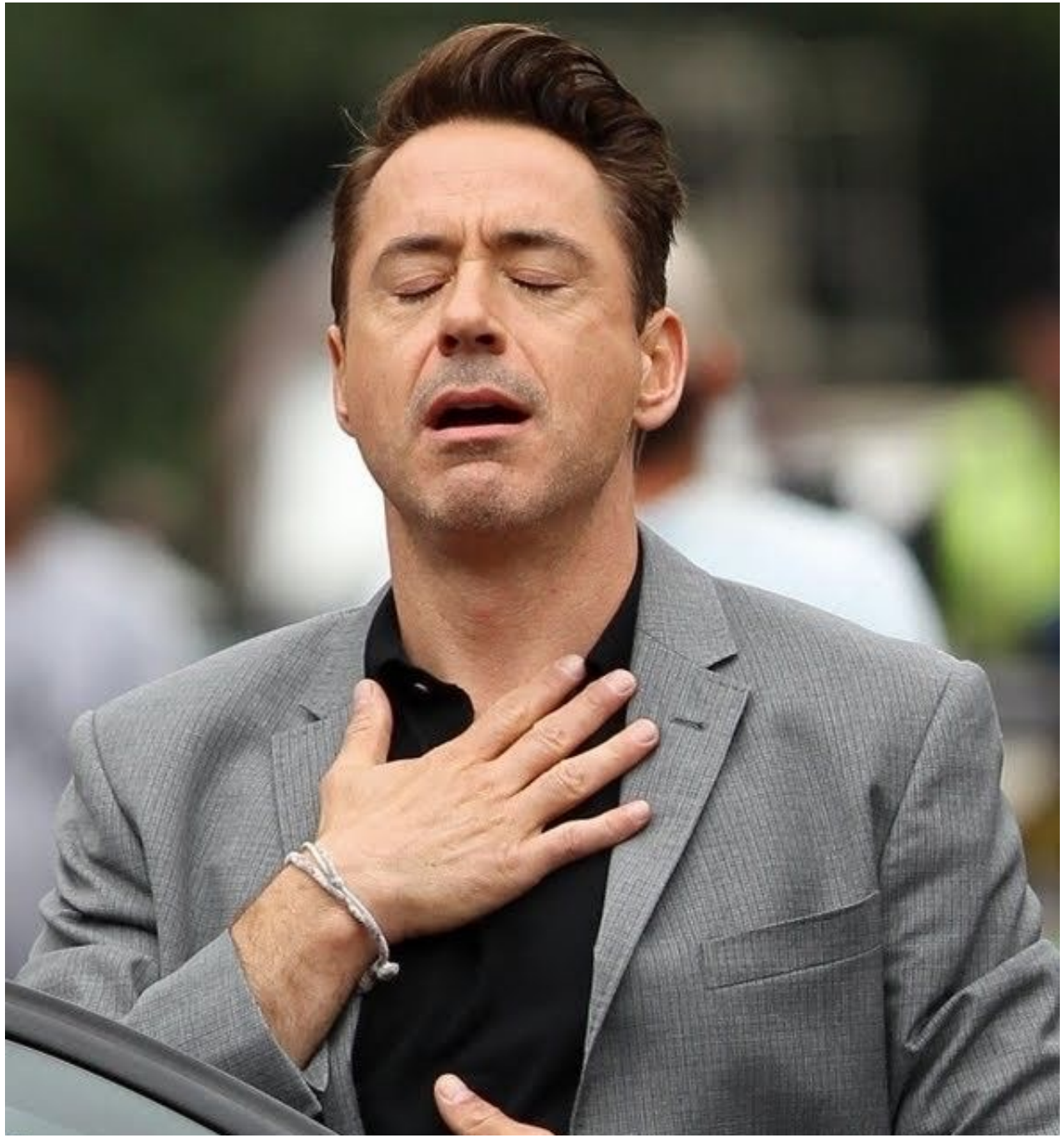
Obfuscation::Filter::Value

- Value obfuscation
  - Casing
    - (name=kRBtgT)
  - Prepended 0's
    - (sAMAccountName=0000805306368)
    - (logonCount>=-0000001)
  - Timestamps
    - (whenCreated=20080217112855.1337KaZanPlehrash)
  - Hex encoding
    - (name=krb\74g\74)
  - Wildcards
    - (name=*rb*g*)



Obfuscation::Filter::That'sAll!





Obfuscation::Filter::Additional




```
(!(userAccountControl:1.2.840.113556.1.4.803:=2))
```

BooleanOperator

ExtensibleMatchFilter

3

4

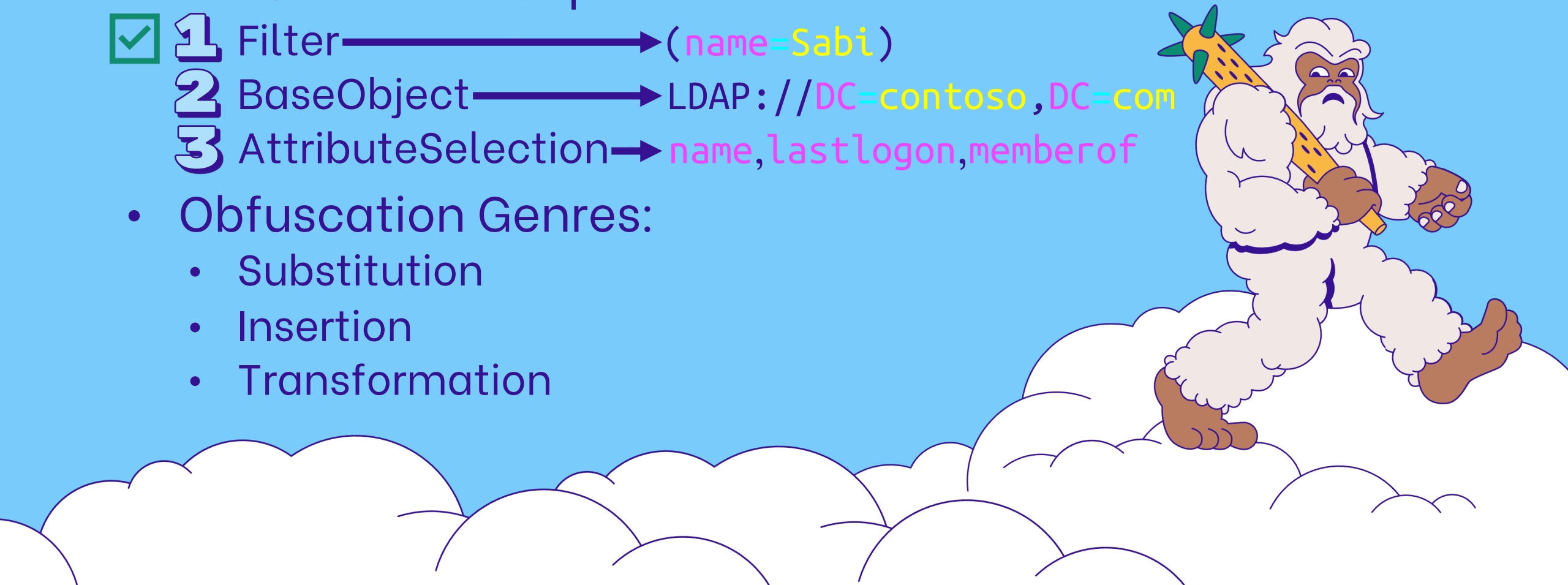
Obfuscation::Filter::Additional

- Filter-wide obfuscation
  - GroupStart/GroupEnd
    - `(((((name=krbtgt)))))`
  - Whitespace (newline, tab, whitespace, etc.)
    - `   (    (    (  (  (    name=  krbtgt)  )  )    )  )  `
  - Garbage filters
    - `(&(name=KushJamUne?)(name=krbtgt)(name=NukEDi!))`
    - `(|(un(e:J((am:=)s)a(b))i)(name=krbtgt)($}_=+;/@-))`
  - Filter/FilterList reordering



Obfuscating LDAP: 3 Deep Dives * 3 Genres

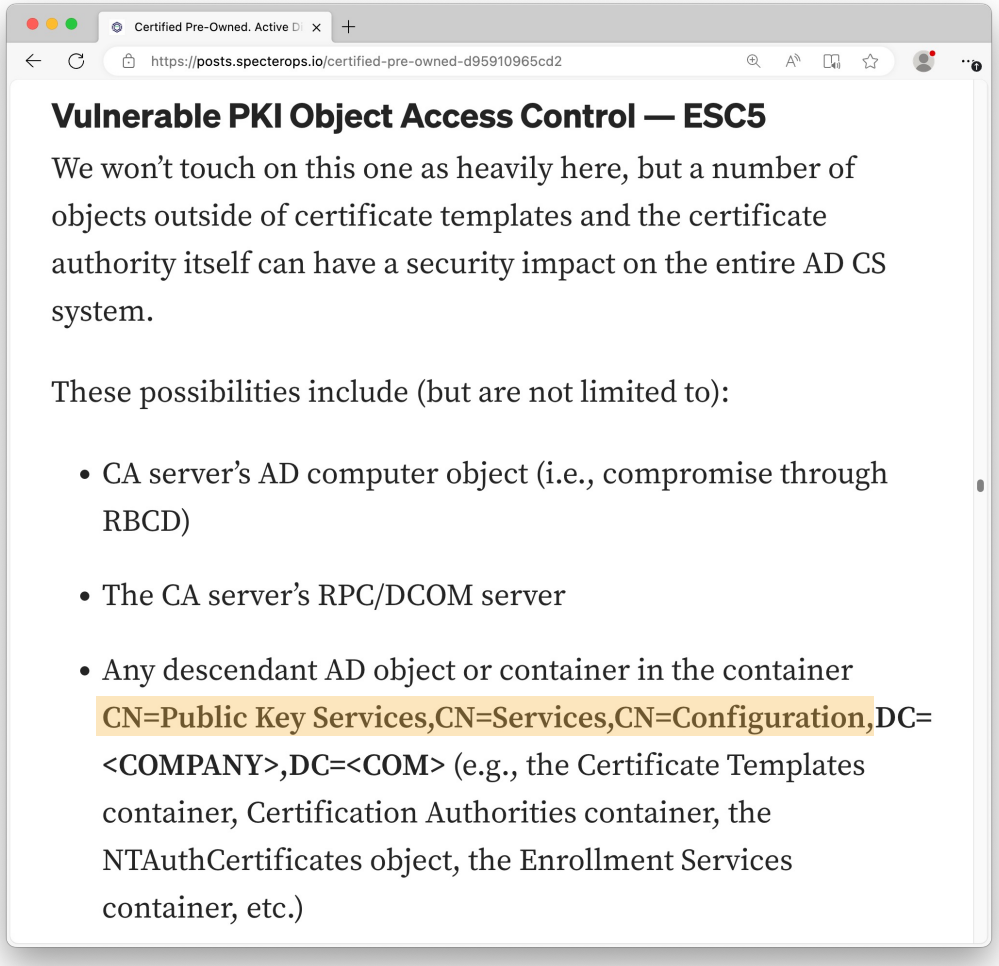
- Obfuscation Deep Dives:
  - ☑ 1 Filter → `(name=Sabi)`
  - 2 BaseObject → `LDAP://DC=contoso,DC=com`
  - 3 AttributeSelection → `name,lastlogon,memberof`
- Obfuscation Genres:
  - Substitution
  - Insertion
  - Transformation



Obfuscation::BaseObject

- BaseObject
  - LDAP://DC=contoso,DC=com
  - LDAP://CN=Public Key Services,CN=Services,CN=Configuration,DC=contoso,DC=com






```
// Container location per MS-WCCE 2.2.2.11.2 Enrollment Services Container
// - https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-wcce/3ec073ec-9b91-4bee-964e-56f22a93a28c
var root = new DirectoryEntry($"LDAP://{LdapServer}CN=Public Key Services,CN=Services,{ConfigurationPath}");
```

Obfuscation::BaseObject

- BaseObject

```
LDAP://DC=contoso,DC=com
```

Obfuscation::BaseObject

- BaseObject
  - Casing

LDAP://dC=coNToSo,dc=cOM

Obfuscation::BaseObject

- BaseObject
  - Casing
  - Hex encoding

LDAP://dC=co\4eToS\6F,dc=\63\4fM

Obfuscation::BaseObject

- BaseObject
  - Casing
  - Hex encoding
  - OID (Object Identifiers) notation

```
LDAP://0.9.2342.19200300.100.1.25=co\4eToS\6F,0.9.2342.19200300.100.1.25=\63\4fM
```

Obfuscation::BaseObject

- BaseObject
  - Casing
  - Hex encoding
  - OID (Object Identifiers) notation
  - Prepended 0's

```
LDAP://0.9.0002342.19200300.100.1.25=co\4eToS\6F,OId.000.9.2342.19200300.100.0001.25=\63\4fM
```

Obfuscation::BaseObject

- BaseObject
  - Casing
  - Hex encoding
  - OID (Object Identifiers) notation
  - Prepended 0's
  - Whitespace

```
LDAP: / / 0.9.0002342.19200300.100.1.25 = co\4eToS\6F ,OId.000.9.2342.19200300.100.0001.25 = \63\4fM
```

Obfuscation::BaseObject



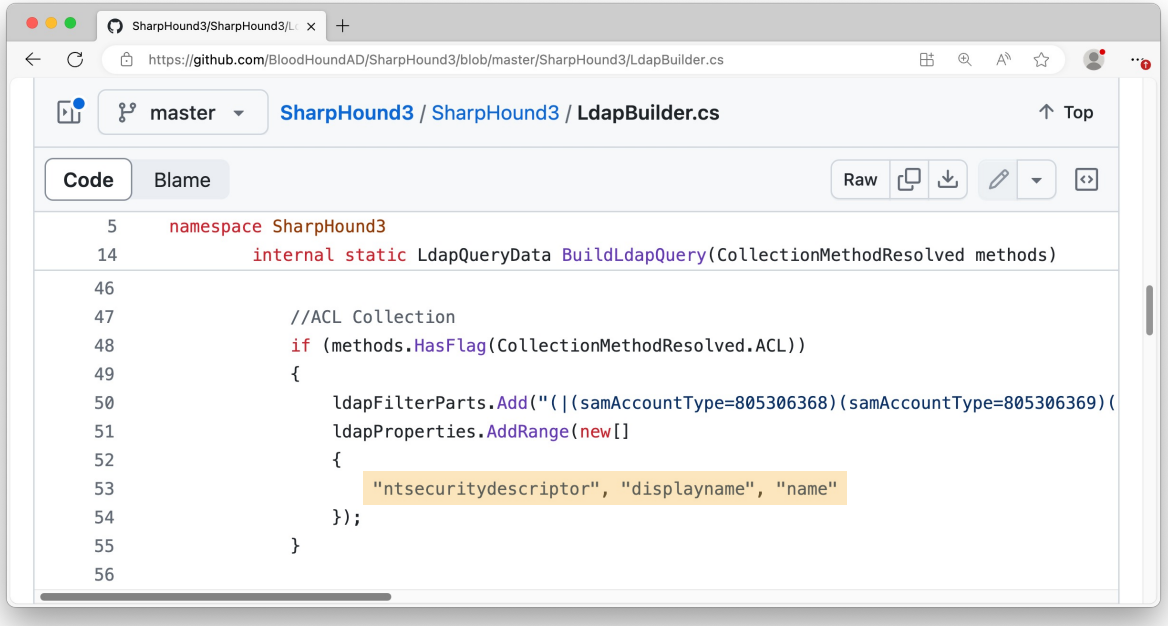
- BaseObject
  - Casing
  - Hex encoding
  - OID (Object Identifiers) notation
  - Prepended 0's
  - Whitespace
  - Double quotes



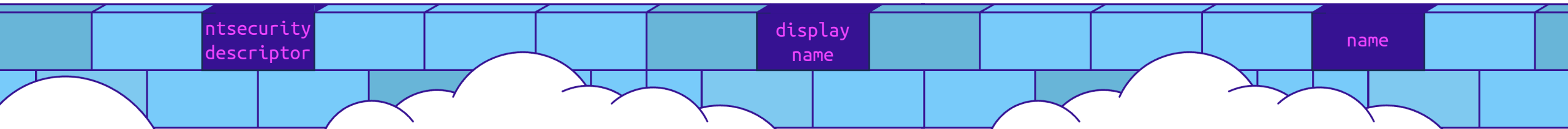
```
LDAP: / /  0.9.0002342.19200300.100.1.25 = "coNToSo" , OId.000.9.2342.19200300.100.0001.25 =  \63\4fM
```

Obfuscation::AttributeSelection

- AttributeSelection

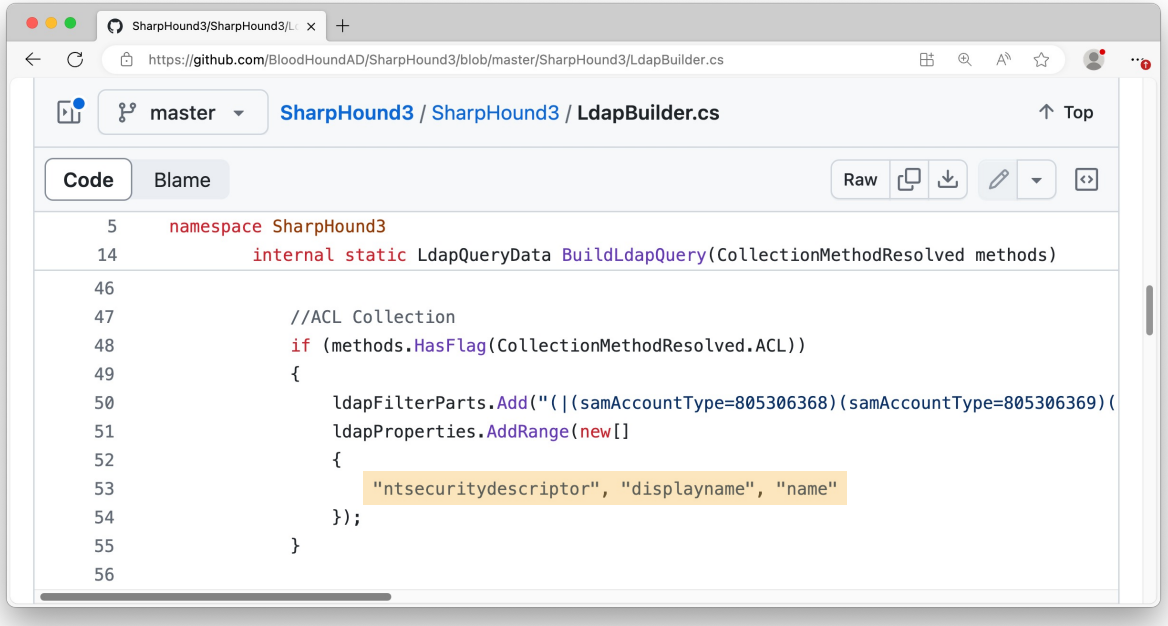


ntsecuritydescriptor,displayname,name

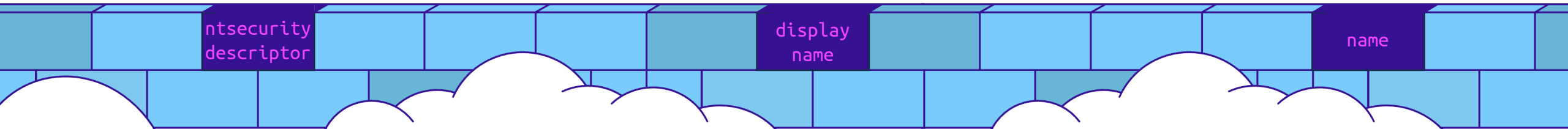


Obfuscation::AttributeSelection

- AttributeSelection
  - Casing

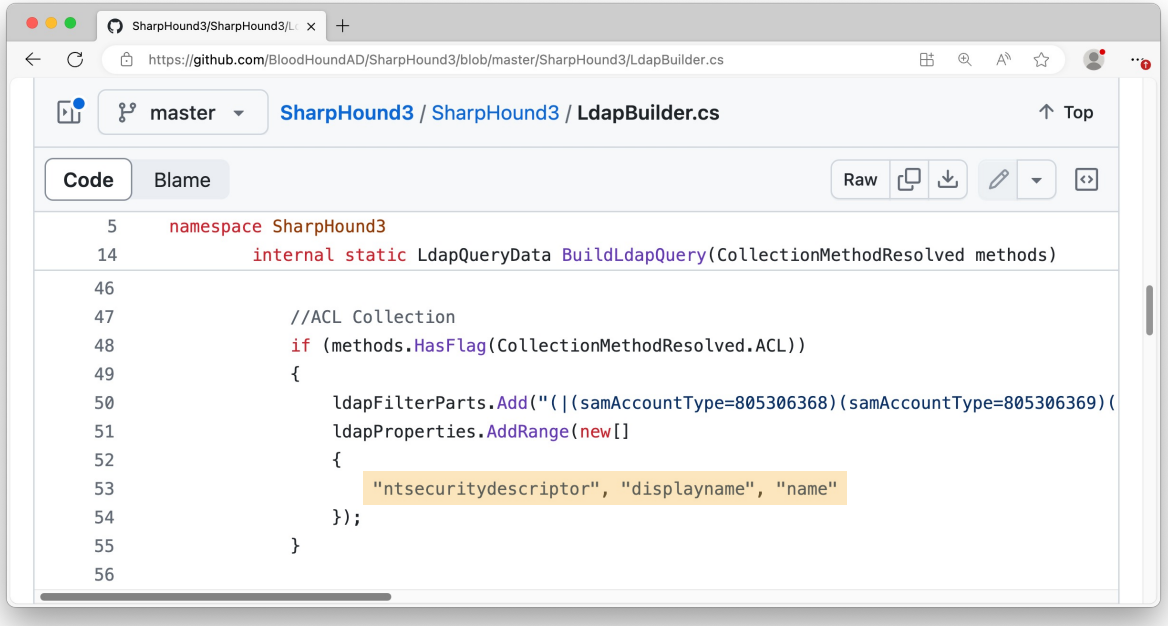


ntSECurITydeSCRIPTor,displayNAme,naMe

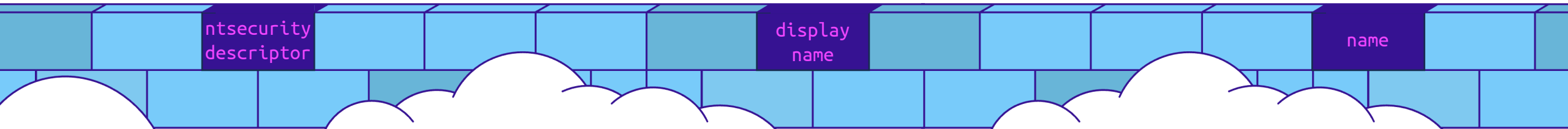


Obfuscation::AttributeSelection

- AttributeSelection
  - Casing
  - OID (Object Identifiers) notation



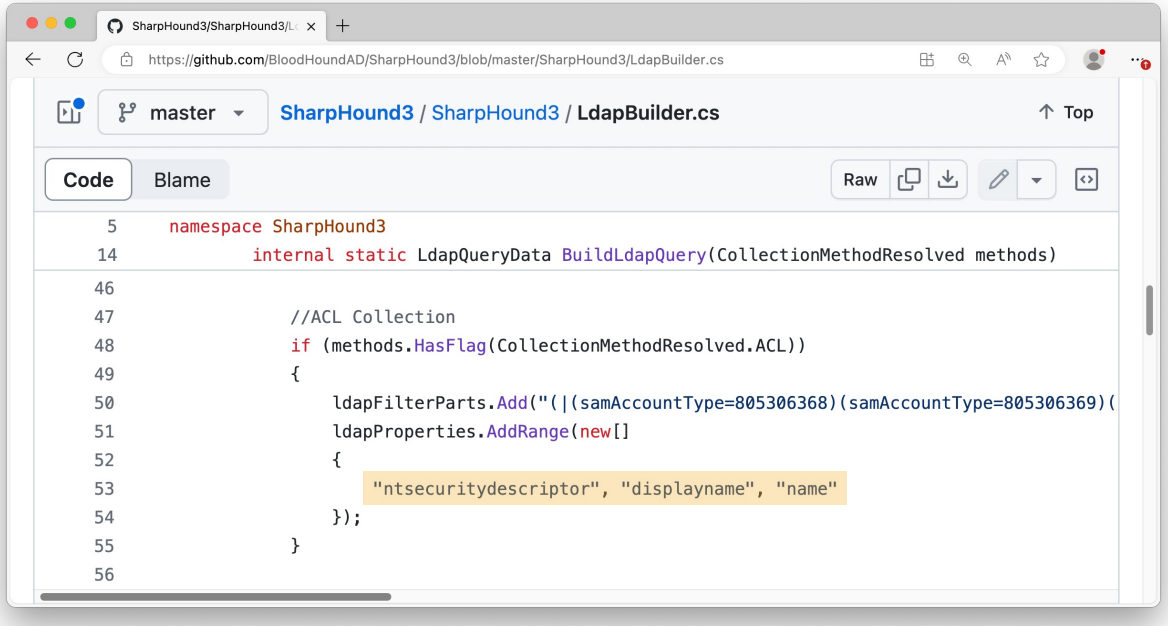
1.2.840.113556.1.2.281,displayNAme,naMe



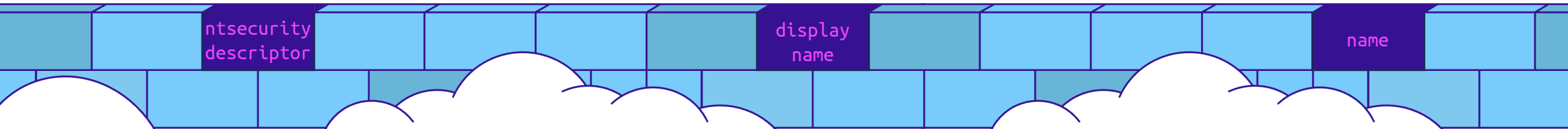
Obfuscation::AttributeSelection



- AttributeSelection
  - Casing
  - OID (Object Identifiers) notation
  - Prepended 0's



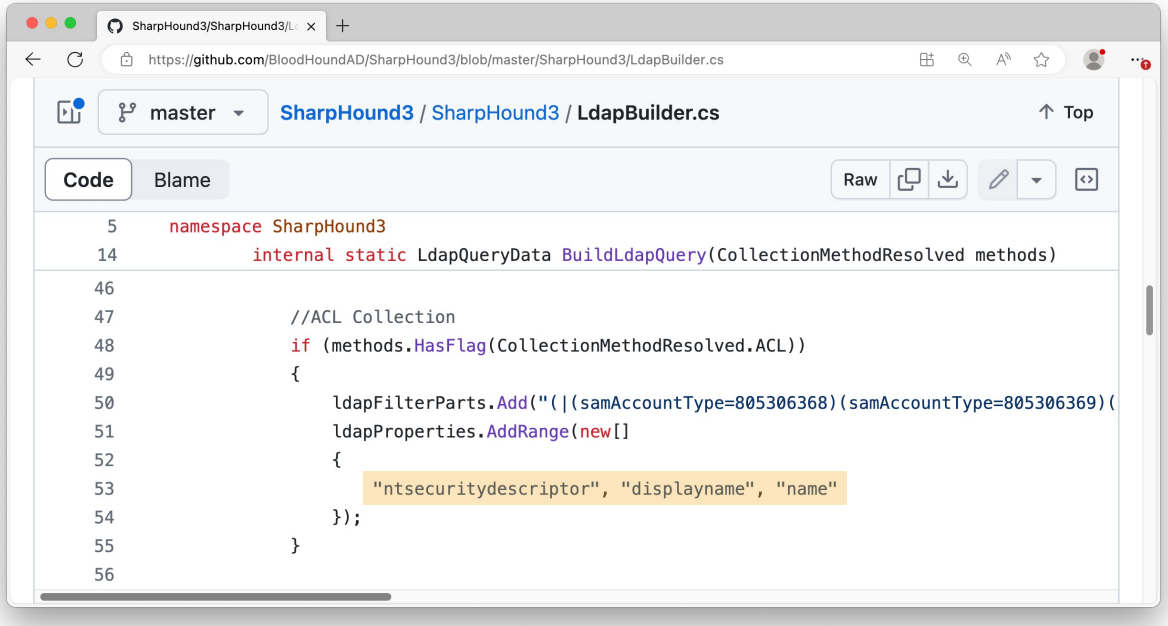
oID.0000**1**.2.840.113556.1.2.**000**281,displayNAme,naMe



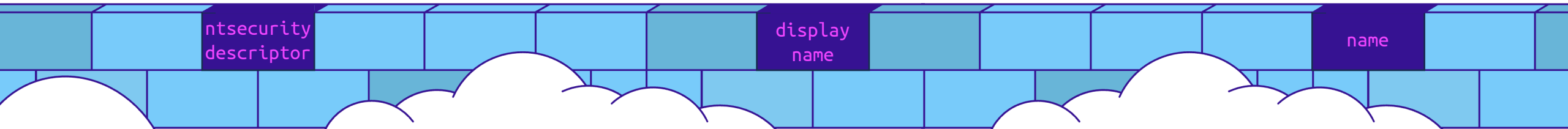
Obfuscation::AttributeSelection



- AttributeSelection
  - Casing
  - OID (Object Identifiers) notation
  - Prepended 0's
  - Whitespace

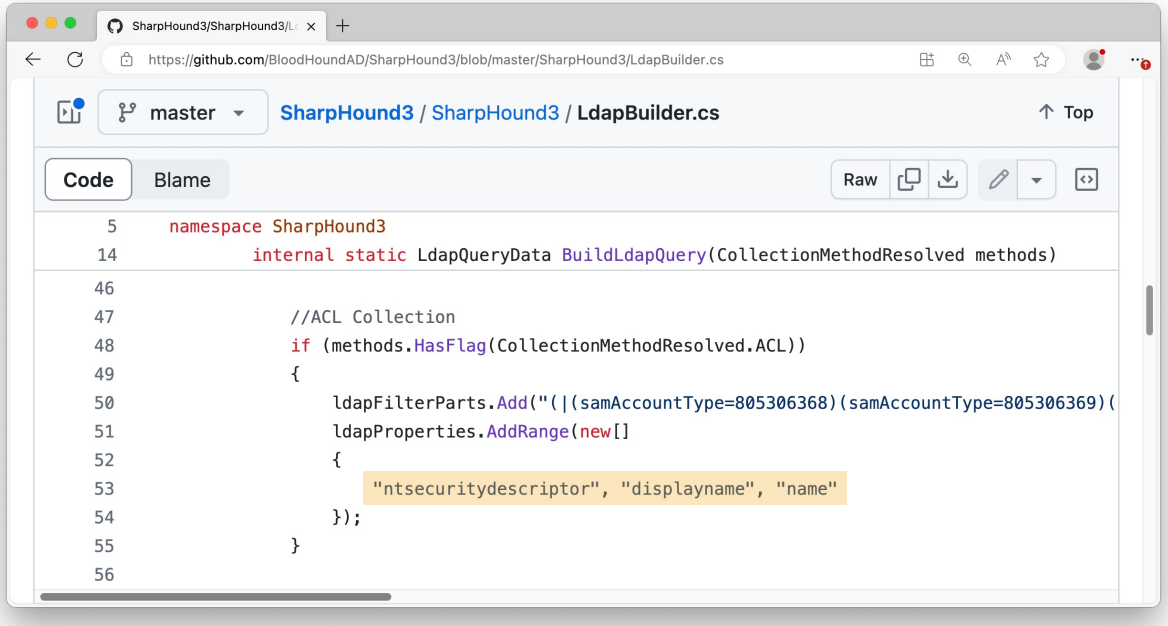


oID.00001.2.840.113556.1.2.000281 ,displayNAme,naMe

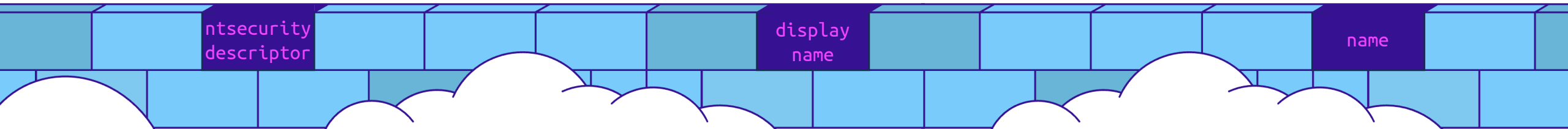


Obfuscation::AttributeSelection

- AttributeSelection
  - Casing
  - OID (Object Identifiers) notation
  - Prepended 0's
  - Whitespace
  - Duplicate attributes

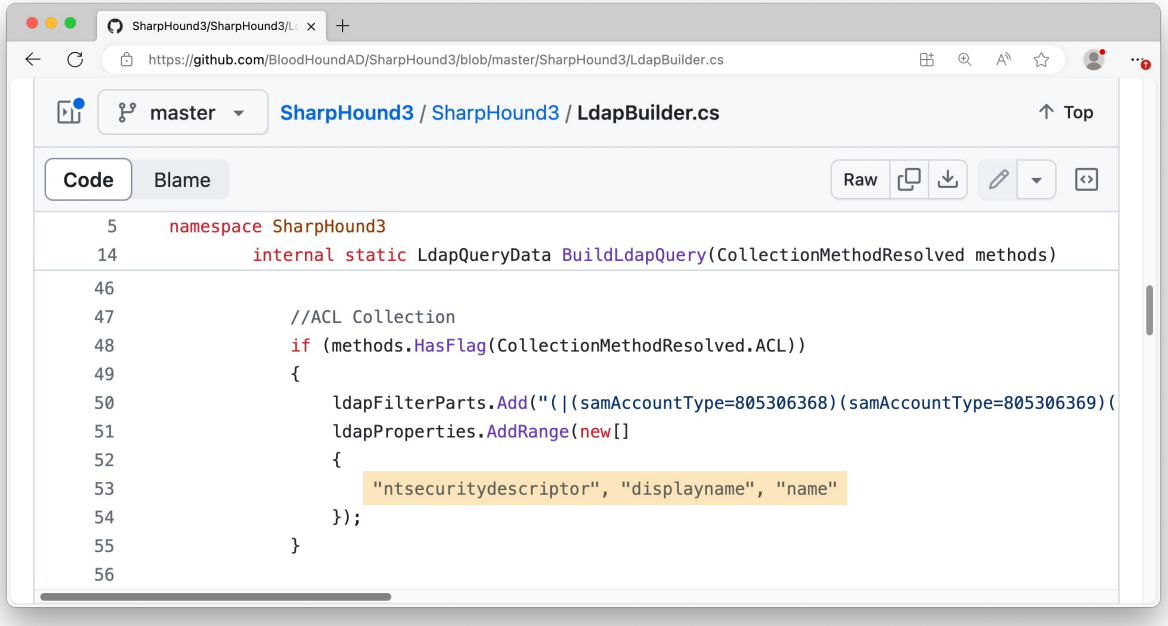


oID.00001.2.840.113556.1.2.000281 ,displayNAme,naMe,NAME

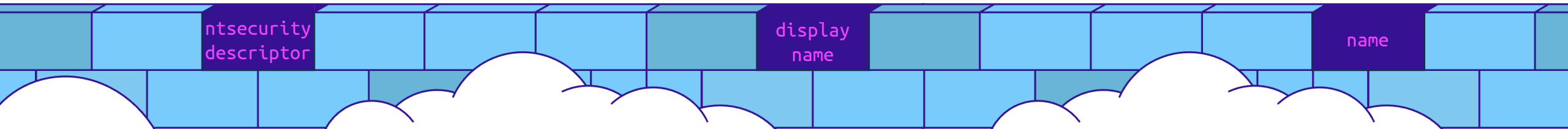


Obfuscation::AttributeSelection

- AttributeSelection
  - Casing
  - OID (Object Identifiers) notation
  - Prepended 0's
  - Whitespace
  - Duplicate attributes
  - Garbage attributes

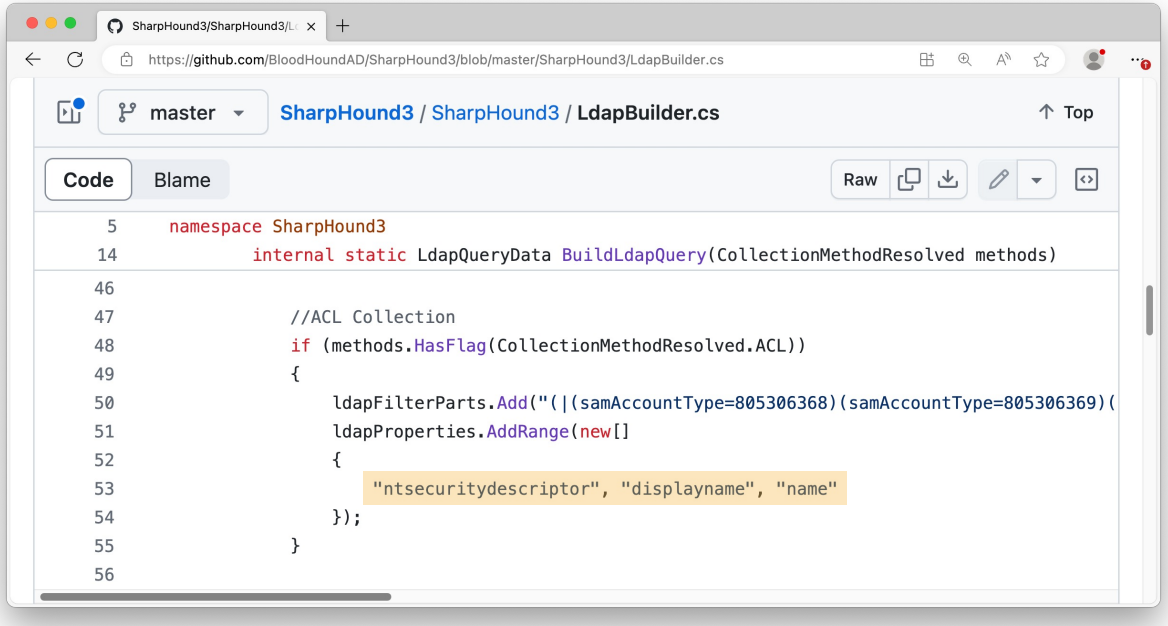



```
oID.00001.2.840.113556.1.2.000281    ,displayNAme,naMe,NAME,AnythingYouWant
```

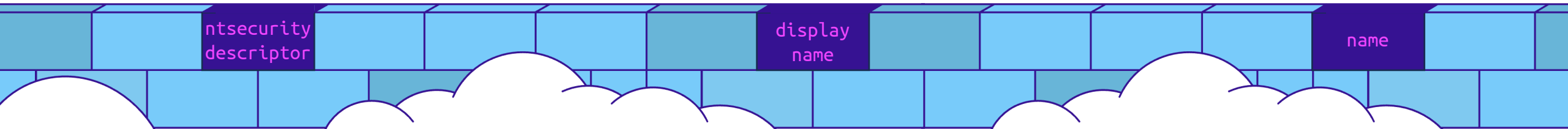


Obfuscation::AttributeSelection

- AttributeSelection
  - Casing
  - OID (Object Identifiers) notation
  - Prepended 0's
  - Whitespace
  - Duplicate attributes
  - Garbage attributes

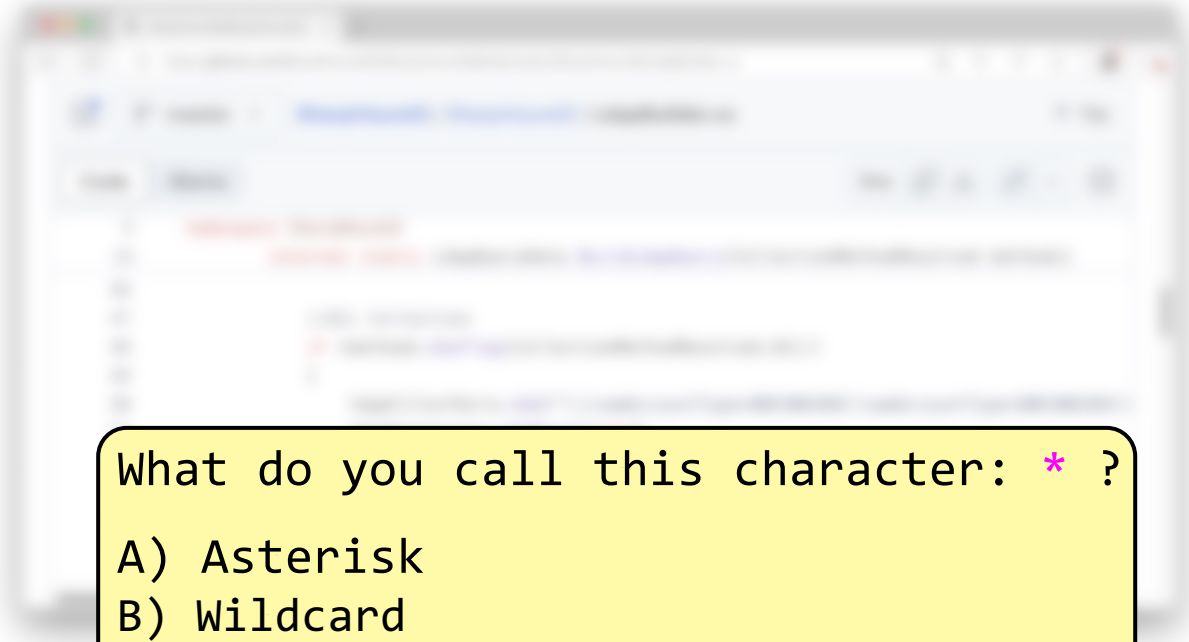


`oID.00001.2.840.113556.1.2.000281` `,displayNAme,naMe,NAME,AnythingYouWant,*`



Obfuscation::AttributeSelection

- AttributeSelection
  - Casing
  - OID (Object Identifiers) notation
  - Prepended 0's
  - Whitespace
  - Duplicate attributes
  - Garbage attributes



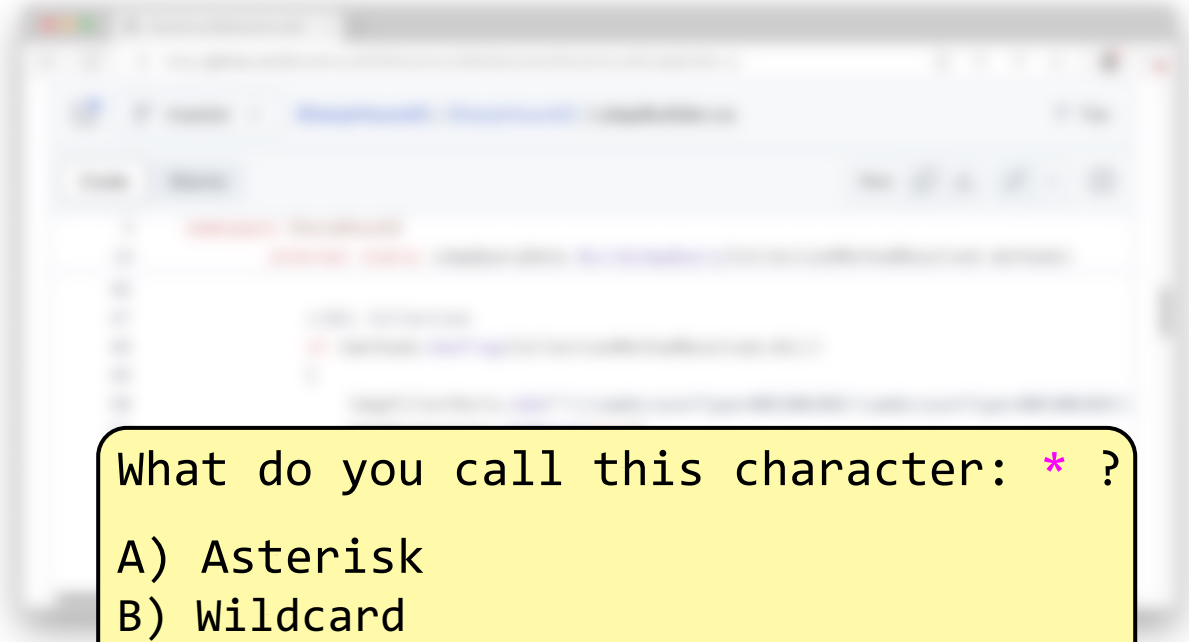
What do you call this character: * ?

A) Asterisk
B) Wildcard
C) Splat



Obfuscation::AttributeSelection

- AttributeSelection
  - Casing
  - OID (Object Identifiers) notation
  - Prepended 0's
  - Whitespace
  - Duplicate attributes
  - Garbage attributes



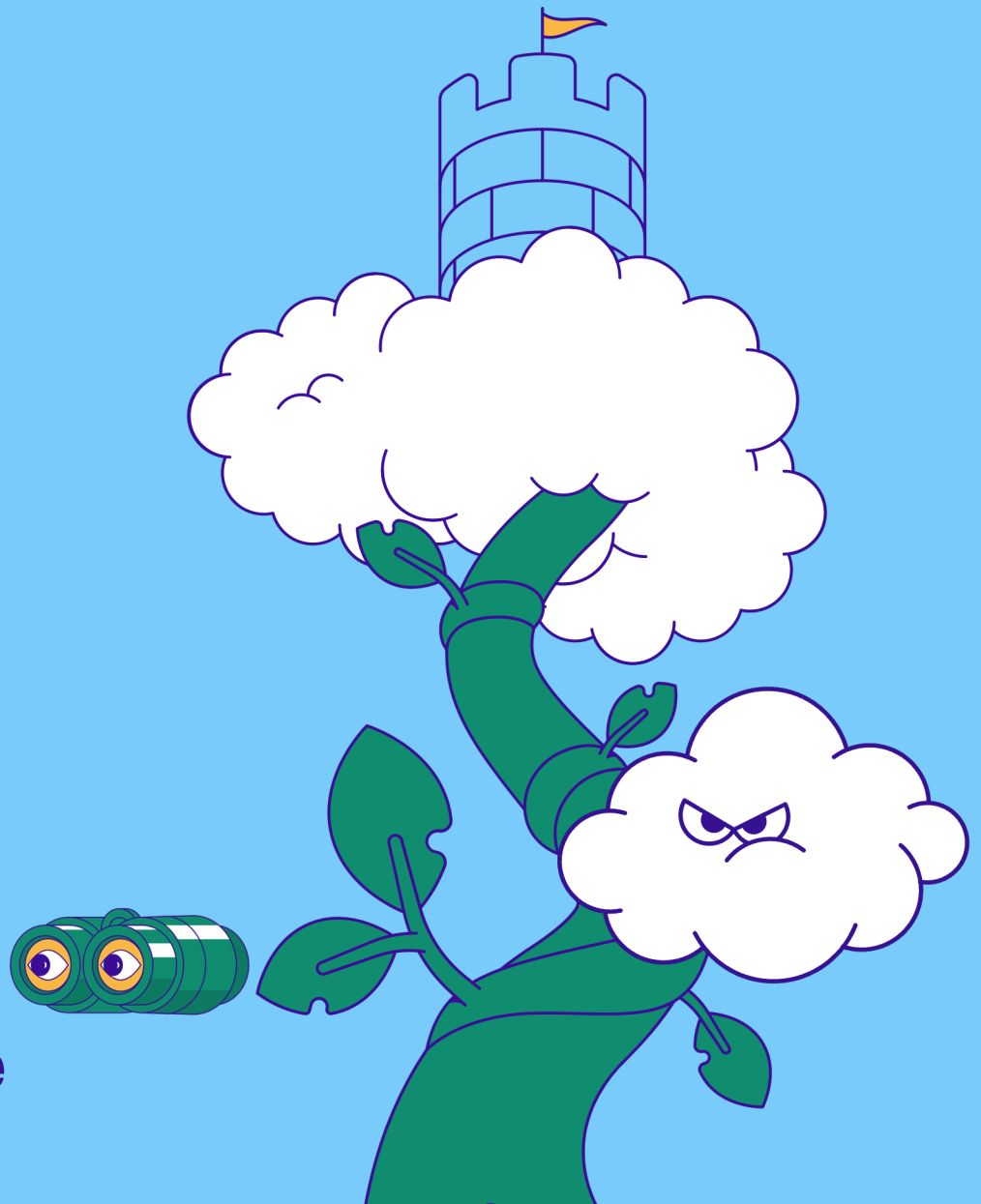
What do you call this character: * ?

A) Asterisk
B) Wildcard
C) Splat
D) Logically same as **null** (RFC-4522)
E) Logging resolves to **[all_with_list]**
F) All of the above



AGENDA

- Introduction
- LDAP Overview
- PROBLEM: Obfuscating LDAP
- SOLUTION: Parse, Enrich, Detect
- MaLDAPtive Tool Demo + Release



Solution::Parse

- Wrote custom parser for LDAP SearchFilters
  - C# State Machine
  - Tokenizer



Solution::Parse

- Wrote custom parser for LDAP SearchFilters
  - C# State Machine
  - Tokenizer





Solution::Parse

- Wrote custom parser for LDAP SearchFilters
  - C# State Machine
  - Tokenizer
  - Syntax Tree





Solution::Enrich



1) String

MaLDAPtive — pwsh

```
(|(name=sabi)(1.2.840.113556.1.4.1=\64\62\6F))
```

2) Tokens

MaLDAPtive — pwsh

Type	Content	ContentDecoded	Start	Length	Depth
GroupStart	(	(	0	1	0
BooleanOperator	|	|	1	1	1
GroupStart	(	(	2	1	1
Attribute	name	name	3	4	1
ComparisonOperator	=	=	7	1	1
Value	sabi	sabi	8	4	1
GroupEnd	)	)	12	1	1
GroupStart	(	(	13	1	1
Attribute	1.2.840.113556.1.4.1	name	14	20	1
ComparisonOperator	=	=	34	1	1
Value	\64\62\6F	dbo	35	9	1
GroupEnd	)	)	44	1	1
GroupEnd	)	)	45	1	0

Decoding

3) Filters & Branches

MaLDAPtive — pwsh

Content	ContentDecoded	Start	Length	Depth
(name=sabi)	(name=sabi)	2	11	1
(1.2.840.113556.1.4.1=\64\62\6F)	(name=dbo)	13	32	1

Solution::Detect



- **Find-Evil**
  - Parses
  - Enriches
  - Detects
  - Scores
  - Explains

65+ Rules

MaLDAPtive — pwsh

```
PS /> '(1.2.840.113556.1.4.8:1.2.840.113556.1.4.803:=1234)' | Find-Evil

Type           : Filter
Author         : Official_MaLDAPtive_Ruleset
Date           : 7/4/2024 12:00:00AM
ID             : SPECIFIC_BITWISE_ADDEND_FOR_DEFINED_ATTRIBUTE_USERACCOUNTCONTROL
Name           : Filter Contains Defined Attribute userAccountControl With
                 Specific Bitwise Addend ENCRYPTED_TEXT_PWD_ALLOWED (128) In
                 Actual Value: '1234'
Example        : (userAccountControl:1.2.840.113556.1.4.804:=65929)
Score          : 20
Depth          : 0
Start          : 0
Content        : (1.2.840.113556.1.4.8:1.2.840.113556.1.4.803:=1234)
ContentDecoded : (userAccountControl:1.2.840.113556.1.4.803:=1234)
```

MaLDAPtive — pwsh

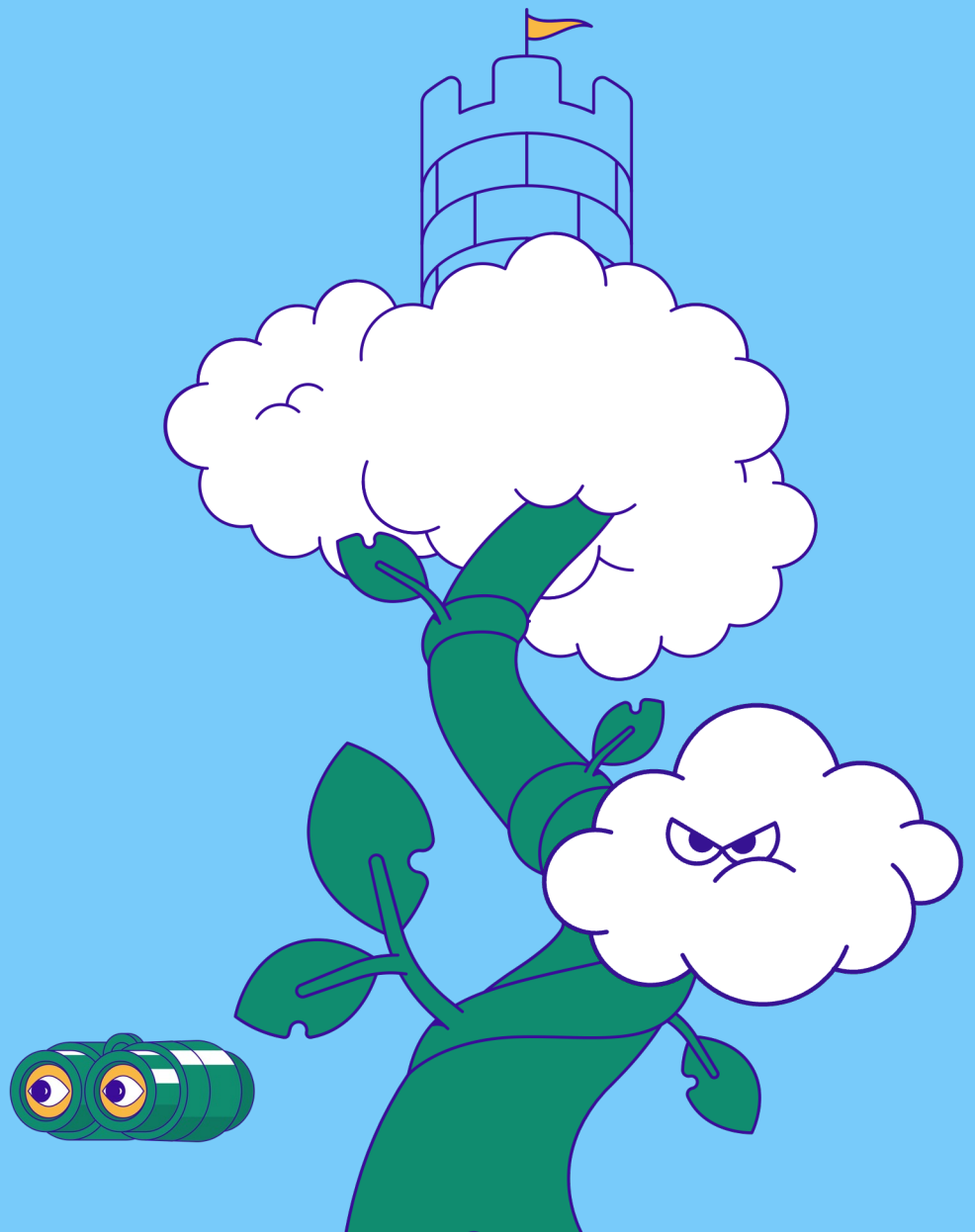
```
PS /> $sfArr = @('(&(objectCategory=Person)(|(name=sabi)(name=dbo)))')*100000;
Measure-Command { $hits = $sfArr | Find-Evil } | Select-Object TotalSeconds

TotalSeconds
------------
   4.3747311
```



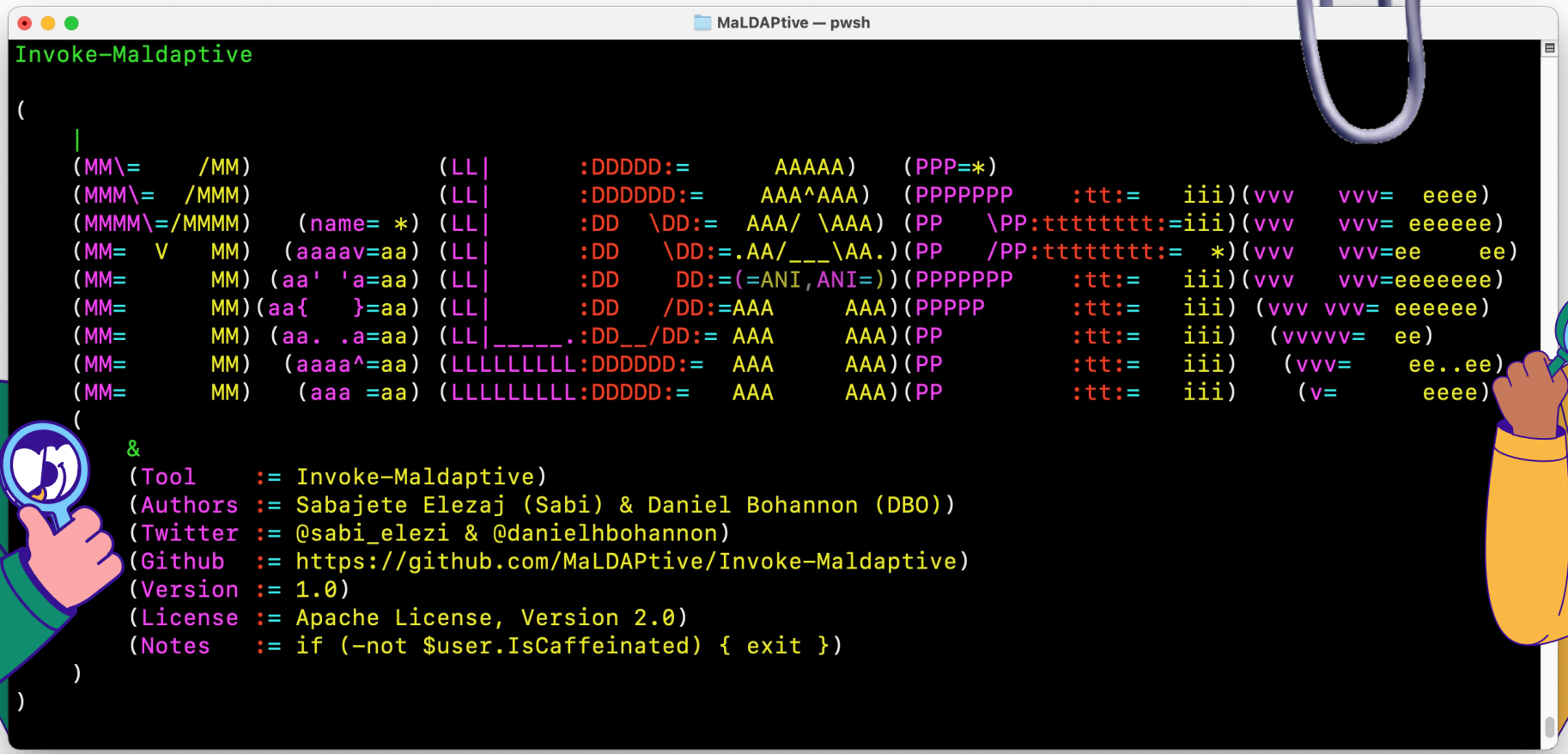


AGENDA

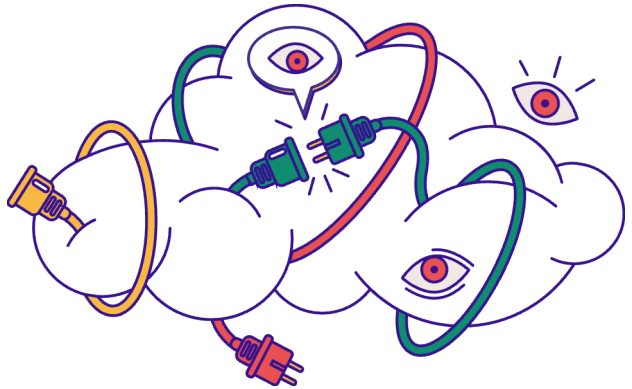
- Introduction
- LDAP Overview
- PROBLEM: Obfuscating LDAP
- SOLUTION: Parse, Enrich, Detect
- MaLDAPtive Tool Demo + Release



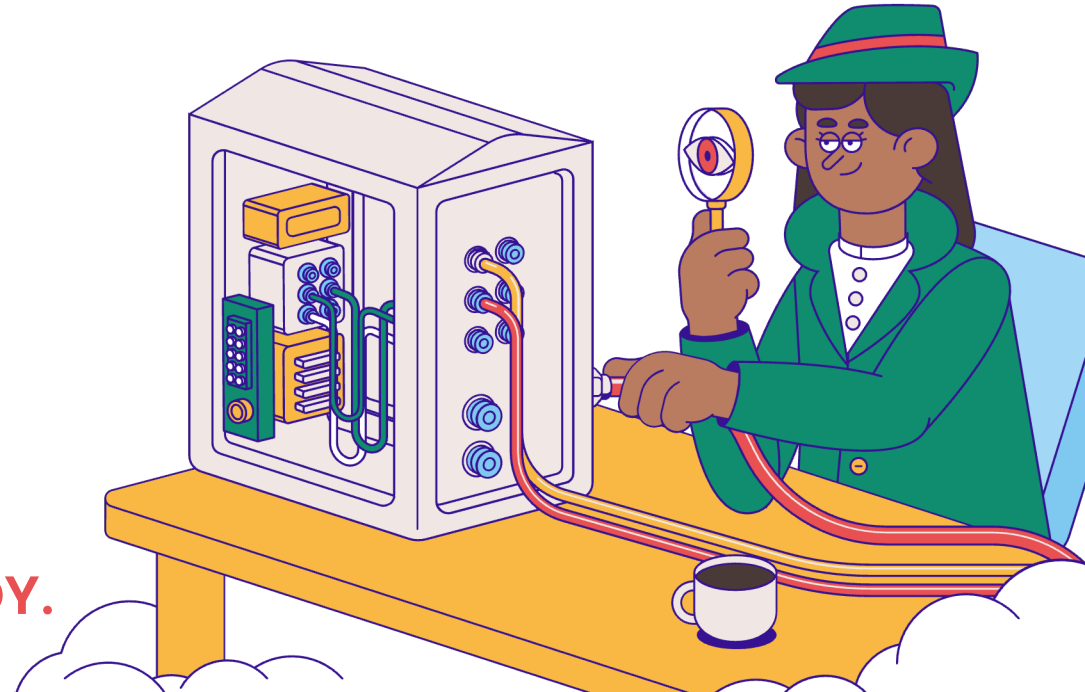
DEMO + Public Tool Release



What We Released Today



1. REMEMBER…we are **defenders**
2. From the beginning we decided on a **non-standard** release format
3. Today's **MaLDAPtive** release includes:
    1. Parser (tokenizer & syntax tree parser)
    2. *Obfuscation module
    3. Deobfuscation module
    4. Detection module
    5. Detection ruleset
    6. Obfuscation corpus



*** Obfuscation module complete w/delayed release by EOY.**

Black Hat Sound Bytes



1. **LDAP SearchRequests are frequently used by attackers** in internal reconnaissance & privilege escalation phases of the attack lifecycle.
2. **Defensive awareness & visibility of LDAP SearchRequests are still relatively immature** with string-based detections being vulnerable to numerous classes of obfuscation & evasion.
3. **Brand new MaLDAPtive open-source framework can improve visibility, detection & deobfuscation of obfuscated LDAP SearchRequests** via custom LDAP parser & detection ruleset.




Rrofsh sa malet.



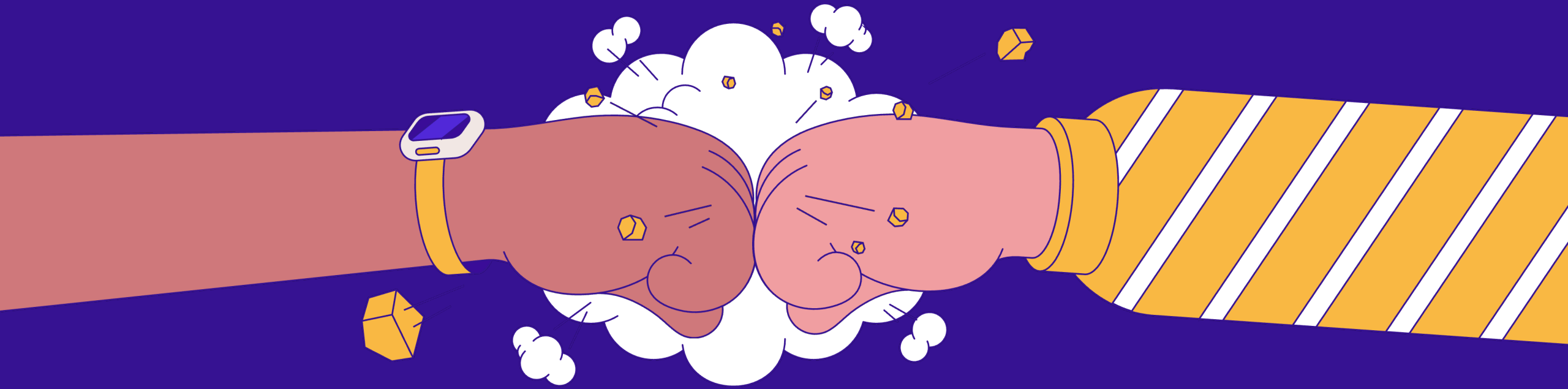
Rrofsh sa malet.

Live long like a mountain.


Rrofsh sa malet.
Live long like a mountain.
Të lutem jo më kapëse letrash.


Rrofsh sa malet.
Live long like a mountain.
Të lutem jo më kapëse letrash.
Please, no more paperclips.

THANKS FOR YOUR TIME!







Solaris

PERMISO

MaLDAPtive — pwsh

```
Invoke-Maldaptive

(
    |
    (MM\=    /MM)            (LL|      :DDDDD:=      AAAAA)   (PPP=*)
    (MMM\=  /MMM)            (LL|      :DDDDDD:=    AAA^AAA)  (PPPPPPP    :tt:=   iii)(vvv   vvv=  eeee)
    (MMMM\=/MMMM)   (name= *) (LL|      :DD  \DD:=  AAA/ \AAA) (PP   \PP:tttttttt:=iii)(vvv   vvv= eeeeee)
    (MM=  V   MM)  (aaaav=aa) (LL|      :DD   \DD:=.AA/___\AA.)(PP   /PP:tttttttt:=  *)(vvv   vvv=ee    ee)
    (MM=      MM) (aa' 'a=aa) (LL|      :DD    DD:=(=ANI,ANI=))(PPPPPPP    :tt:=   iii)(vvv   vvv=eeeeeee)
    (MM=      MM)(aa{   }=aa) (LL|      :DD   /DD:=AAA     AAA)(PPPPP      :tt:=   iii) (vvv vvv= eeeeee)
    (MM=      MM) (aa. .a=aa) (LL|_____.:DD__/DD:= AAA     AAA)(PP         :tt:=   iii)  (vvvvv=  ee)
    (MM=      MM)  (aaaa^=aa) (LLLLLLLLL:DDDDDD:=  AAA     AAA)(PP         :tt:=   iii)   (vvv=    ee..ee)
    (MM=      MM)   (aaa =aa) (LLLLLLLLL:DDDDD:=   AAA     AAA)(PP         :tt:=   iii)    (v=      eeee)
    (
        &
        (Tool    := Invoke-Maldaptive)
        (Authors := Sabajete Elezaj (Sabi) & Daniel Bohannon (DBO))
        (Twitter := @sabi_elezi & @danielhbohannon)
        (Github  := https://github.com/MaLDAPtive/Invoke-Maldaptive)
        (Version := 1.0)
        (License := Apache License, Version 2.0)
        (Notes   := if (-not $user.IsCaffeinated) { exit })
    )
)
```





“SABI”
SABAJETE ELEZAJ

“DBO”
DANIEL BOHANNON

sabajete_elezaj

danielhbohannon

@sabi_elezi

@danielhbohannon





https://github.com/MaLDAPtive/**Invoke-Maldaptive**

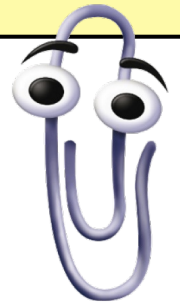
References

1. https://learn.microsoft.com/en-us/archive/technet-wiki/5392.active-directory-ldap-syntax-filters
2. https://learn.microsoft.com/en-us/windows/win32/ad/active-directory-schema
3. https://www.rfc-editor.org/rfc/rfc4511
4. https://ldap.or.kr/ldap-a-to-z/
5. https://www.binarydefense.com/resources/blog/uncovering-adversarial-ldap-tradecraft/
6. https://posts.specterops.io/an-introduction-to-manual-active-directory-querying-with-dsquery-and-ldapsearch-84943c13d7eb
7. https://posts.specterops.io/manual-ldap-querying-part-2-8a65099e12e3

Clippy Jokes Cut for Time (You're Welcome)

If I became an LDAP SearchRequest
then what kind of Filter would I be?

A Presence Filter – because I'm
ALWAYS HERE!



Okay, here's a non-LDAP joke:
Where do I live when not on screen?

C:\Windows\System32\clip.exe



Why do I exist?

To help you identify the cutoff between
millennials & GenZ without saying a word.

