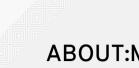


#### GLOBAL CATALOG SERVICE IMPLEMENTATION IN FREEIPA

Alexander Bokovoy Red Hat Inc. May 4th, 2017



### **ABOUT:ME**

Sr. Principal Software Engineer at Red Hat

- Samba Team member since 2003
- Core FreeIPA developer since 2011



### WHAT IS THIS TALK ABOUT?



### WHAT IS THIS ABOUT?

FreeIPA is a directory service for Linux and POSIX clients:

- 389-ds The LDAP directory server (and a lot of plugins)
- Samba as a traditional (NT4-style) domain controller with a twist (smbd and winbindd)
- MIT Kerberos Kerberos KDC
- MS-KKDCP proxy
- Dogtag Certificate Authority
- Custodia (secrets management)
- SSSD client side identity (nss, PAM, D-Bus, ...)
- FreeIPA management framework written in Python and running under Apache



### WHAT IS THIS ABOUT?

FreeIPA supports forest trust to Active Directory:

- Active Directory sees FreeIPA as a "native Active Directory" deployment
- Since Samba 4.5 it is possible to establish a trust between Samba AD and FreeIPA
- Active Directory users can access resources on FreeIPA clients
- FreeIPA users cannot natively access resources in Active Directory



### WHAT IS THIS ABOUT?

FreeIPA users cannot access resources in Active Directory:

- Access control in Active Directory uses SIDs of users/groups in ACLs
- "Security" tab in UI deals with user and group names
- Windows performs user or group SID lookup
- FreeIPA does not provide interfaces expected by Active Directory to perform name to SID lookups



### ANATOMY OF A NAME RESOLUTION



### FOUR STYLES OF CONVERSATION

Active Directory has four ways of discovering SIDs of users/groups:

- Domain controller LDAP ping allows user name validation ([MS-ADTS] 6.3.3.2 Domain Controller Response to an LDAP Ping) but doesn't allow to discover SIDs
- DsCrackNames is part of DRSU API, Directory Replication Service of Active Directory. It is not implemented in smbd, only in Samba AD
- LsaLookupNames and SamLogon families of RPC calls
- LDAP queries to Global Catalog



LDAP ping is used by all Windows clients to discover closest domain controller. As part of it clients may request a user name validation to avoid hitting domain controllers that don't have that user replicated



## Windows UI seems to trigger CLDAP ping requests with random user names instead of the one you entered:

Protects							
Units         Construction         Construction <thconstruction< th="">         Construction</thconstruction<>	_						
V7         UAD         200         searchestrijuge							
Unit         Unit <th< td=""><td></td><td></td><td></td><td></td><td></td><td>succes</td><td>▼ and: (&amp;(&amp;(&amp;(&amp;(&amp;(&amp;(ClosDomain=xs.ipa.cool)(Host=METIS))(User=USER2))(AAC=10:00:00))(DomainSid=S-1-5-21-57)</td></th<>						succes	▼ and: (&(&(&(&(&(&(ClosDomain=xs.ipa.cool)(Host=METIS))(User=USER2))(AAC=10:00:00))(DomainSid=S-1-5-21-57)
1000         2000 <th< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></th<>							
Libre CLAP         222         searchapset[13]						succes	
707         CLAM         308         searchhast(17)213         -Mool?*         samaliest(12)21         -Mool?*         samaliest(12)21         -Mool?*         samaliest(12)21         -Mool?*         samaliest(12)21         -Mool?*         -Mool?*         samaliest(12)21         -Mool?*         samaliest(12)21         -Mool?*         samaliest(12)21         -Mool?*         -Mool?*         samaliest(12)21         -Mool?*         samaliest(12)21         -Mool?*         -Mool?*         samaliest(12)21         -Mool?*         -Mool **							
Libro CLMP         222 searcheschity/D22 - MODF** searcheschite/D22 location         Comment						succes	
100         200         searchest(1)(1)         1000000000000000000000000000000000000							
Link CLMP         223         sarchhastinty/E33         ************************************						succes	
V7 0LW         30 secrethese(1)913 - MODP* secrethese(1)914 secret secrethese(1)914         - and 11es: spat11petts 1:0; - spat2(1)petts							
Line         CAMP         223         searchbestify(19)         ************************************						succes	
PT         CLAM         PT							
Labe         CLAP         223         sarchfastifty/1931         *4007*         sarchfastifty/1931         *4						succes	
V7         CLAW         201         searchbestry(198)         -Memory Second							
Unit         Class         Sol         Sol<						succes	
V70         UAD         SearchBest(17)/179         *MODP**         SearchBest(17)/179							
Labe         CLAP         223         searchleget(19)/197         *4007*         searchleget(19)/197         searchleg						succes	
C102         C102 <thc102< th="">         C102         C102         <thc< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></thc<></thc102<>							
1.08       CLMP       223       searchbest(17)(788)       *4007***       searcbest(19)(788)       *4007***       searcbest(19)(788)       *4007***       searcbest(19)(788)       *4007***       searcbest(19)(788)							
V77         ULM         383         saarchbergers(199)         *MOD**         same/belau/(199)         MOD**						140000	
1.00       CLMP       223       sacrofhenderforty/1093       "-4000"**       sacrofhenderforty/1093       "-4000"***       sa						Jucces	
M27         CLAMP         382         searc/Mesuper(1600)         -400075**         Searc/Mesuper(1600)         -400075**         Searc/Mesuper(1600)         -400075**         Searc/Mesuper(1600)         -400075**         Searc/Mesuper(1600)         -400075**         Searc/Mesuper(1600)         -400075**         -400075*						succes	
Lake       CLAP       223       searchikelöntry(460)       *e0007b*       searchikelöntosi       1000       00							
		223	searchResEntry[d86	) *«ROOT»*	searchResDone(488)	succes	0010 00 00 f8 11 3d 2a 02 27 70 00 17 00 00 02 1a=*. 'p
			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,				
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$							0050 0a 01 00 0a 01 08 02 01 00 02 01 00 01 01 00 a0
0000       0f       <							
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$							
00000       00       14       44       16       16       10       16       10							0090 05 4d 45 54 49 53 a3 84 00 00 0d 04 04 55 73 .METISUs
00-c0         04         09         46         04         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10         16         10							
00000         its 44 7: 15 44 4.3         15 44 10 00 00 40 40 30 44 72 45 80         10.1							
0100 44 64 77 48 64 77 74 46 61 66 65 40 11 66 75 40 005 80 105 74 0110 69 73 24 61 64 26 96 76 61 62 63 96 76 61 65 36 16 53 76 005 16 54 16 16 54 16 16 54 16 16 54 16 16 16 16 16 16 16 16 16 16 16 16 16							00c0 c6 4f 7c fb 44 a3 84 00 00 00 0d 04 05 4c 74 56 .0].DNtV
0110 69 73 26 61 64 28 69 70 12 66 37 66 67 67 66 68 12 66 37 67 67 66 68 12 66 37 67 67 66 68 12 66 36 68 66 68 66 68 68 68 68 57 46 66 67 57 67 66 68 12 56 56 56 68 12 56 56 56 56 56 56 56 56 56 56 56 56 56							
0120 00 00 00 00 00 00 40 00 40 05 74 6c 6f 67 6f 6eNe tlogon							
	4						



## Windows UI seems to trigger CLDAP ping requests with random user names instead of the one you entered:

Protoci         Lungth: Info           697         CLAP         202 searchbergest(198)         -46071-* basebject         + andi (SciEGERGEGEEGEEGEEGEEGEEGEEGEEGEEGEEGEEGEEG	
:1a9e CLDAP 223 searchResEntry(389) " <root>" searchResDone(389) succes + Filter: (DnsDowain+xs.ipa.cool)</root>	
ab27_f10AP	
:1a9e CLDAP 223 searchResEntry(390) " <r00t>" searchResDone(390) succes + equalityMatch</r00t>	
eb77 CLDAP 302 searchRequest(391) " <root>" baseObject Filter: (Host=METIS)</root>	
1a9e CLDAP 223 searchResEntry(391) **R00T>* searchResDone(391) succes * and item; equalityMatch (3)	
eb77 CLDAP 308 searchRequest(392) * <root>* baseObject + equalityMatch</root>	
1a9e CLDAP 223 searchResEntry(392) "-RDOT>" searchResDone(392) succes Filter: [User=ADMIN]	
eb77 CLDAP 302 searchRequest(393) " <root>" baseObject • and item; equalityMatch (3)</root>	
:1a9e CLDAP 223 searchResEntry(393) " <root>" searchResDone(393) succes } equalityMatch</root>	
eb77 CLDAP 302 searchRequest(394) " <r00t>" baseObject Filter: (AAC=10:00:00:00)</r00t>	
:1a9e CLDAP 223 searchResEntry(394) " <root>" searchResDone(394) succes • and item: equalityMatch (3)</root>	
eb77 CLDAP 304 searchReguest(395) " <root>" baseObject FegualityMatch</root>	
:1a9e CLDAP 223 searchResEntry(395) * <root>* searchResDone(395) succes Filter: (DomainSid=5-1-5-21-570121326-3336757064-1157332047 (Domain SID))</root>	
eb77 CLDAP 301 searchRequest(396) " <r00t>" baseObject = and item: equalityMatch (3)</r00t>	
:la9e CLDAP 223 searchResEntry(396) " <root>" searchResDone(396) succes equalityMatch</root>	
eb77 CLDAP 302 searchRequest(397) "«R00T»" baseObject Filter: (NtVer=0x01000016)	
:la9e CLDAP 223 searchResEntry(397) " <root>" searchResDone(397) succes • and item: equalityMatch (3)</root>	
eb77 CLDAP 304 searchRequest(398) "«R00T»" baseObject + equalityMatch	
:la9e CLDAP 223 searchResEntry(398) " <root>" searchResDone(398) succes * Filter: (DnsHostName=metis.ad.ipa.cool)</root>	
eb77 CLDAP 301 searchRequest(399) *«ROOT»* baseObject • and item: equalityMatch (3)	*
:1a9e CLDAP 223 searchResEntry(399) " <r00t>" searchResDone(399) succes</r00t>	•
eb77 CLDAP 302 searchRequest(400) "«ROOT»" baseObject 0000 00 1a 4a 62 eb 77 5c 5e ab 7c ea 81 85 dd 60 003b.v\^	
:109E CLDAP 223 searchResEntry(400) " <r00t>" searchResDone(400) succes(0010 00 00 016 11 13 2a 02 27 70 00 17 00 00 02 1a" p</r00t>	
0030 4a ff fe 62 eb 77 e4 06 01 85 00 f8 64 2d 30 84 J.b.wd.0.	
0040 00 00 00 es 02 02 01 86 63 84 00 00 es 04 00 c	
0050 0a 01 00 0a 01 00 02 01 00 02 01 00 02 01 00 02 01 00 a0	
0060 84 00 00 00 b9 33 84 00 00 00 18 04 09 44 6e 73	
0080 6f 6f 6c a3 84 00 00 00 0d 04 04 48 6f 73 74 04 ool	
0090 05 4d 45 54 49 53 a3 84 00 00 0d 04 04 55 73 .METISUs	
0030 65 72 04 05 41 44 46 49 4e a3 84 00 00 00 0b 4e er, AOHIN 00b0 03 41 41 43 40 40 40 00 00 00 a3 84 00 00 02 53 AAC	
0000 0 04 1 41 43 94 04 10 00 00 08 84 00 00 02 5 AAC	
00d0 00 00 00 00 05 15 00 00 06 5c fb 21 48 df e2	
00e0 c6 4f 7c fb 44 a3 84 00 00 00 04 04 05 4e 74 56 .0].DNtV	
0070 05 72 04 04 16 00 00 01 33 84 00 00 00 20 04 0b er	
0110 0 44 00 73 40 01 73 74 10 01 00 05 44 11 00 05 74 0 Distribute and Not 0110 0 05 73 26 01 04 26 05 70 01 22 63 36 06 05 30 44 is adj as accold.	
0120 00 00 00 0a 04 08 4e 65 74 6c 6f 67 6f 6eNe tlogon	



## Windows UI seems to trigger CLDAP ping requests with random user names instead of the one you entered:

	Protocol		Info		The barrent in the		# and: (&{&{&{&{DSDBWAIN=XS.ipa.cool}(Most=METIS)}(User=ALICE)}(AAC=10:00:00:00))(DowaInSId=S-1-5-21-576
							▼ and: 7 items
	CLDAP				DOT>" searchResDone(3)	<li>succes</li>	
	CLDAP				OT>" baseObject		<ul> <li>and item: equalityMatch (3)</li> </ul>
	CLDAP				DOT>" searchResDone(3	<li>io) succes</li>	
	CLDAP		searchRequest				<ul> <li>Filter: (Host=METIS)</li> </ul>
	CLDAP				DOT>" searchResDone(3	<li>succes</li>	and item: equalityMatch (3)
	CLDAP		searchRequest				equalityMatch
	CLDAP				DOT>" searchResDone(3	<li>32) succes</li>	Filter: (User=ALICE)
	CLDAP				)T>" baseObject		<ul> <li>and item: equalityMatch (3)</li> </ul>
	CLDAP				DOT>" searchResDone(3	<li>33) succes</li>	equalityMatch
	CLDAP		searchRequest				<ul> <li>Filter: (AAC=10:00:00:00)</li> </ul>
	CLDAP				DOT>" searchResDone(3	<li>34) succes</li>	<ul> <li>and item; equalityMatch (3)</li> </ul>
	CLDAP		searchRequest				equalityMatch
	CLDAP				DOT>" searchResDone(3	35) succes	<ul> <li>Filter: (DomainSideS-1-5-21-570121326-3336757064-1157332047 (Domain SID))</li> </ul>
	CLDAP		searchRequest				<ul> <li>and item; equalityMatch (3)</li> </ul>
	CLDAP				DOT>" searchResDone(3	<li>36) succes</li>	equalityMatch
	CLDAP		searchRequest				<ul> <li>Filter: (NtVer=0x01000016)</li> </ul>
	CLDAP				DOT>" searchResDone(3	<li>97) succes</li>	<ul> <li>and item: equalityMatch (3)</li> </ul>
	CLDAP		searchRequest				<pre>equalityMatch</pre>
	CLDAP				DOT>" searchResDone(3	<li>38) succes</li>	<ul> <li>Filter: (DnsHostName=metis.ad.ipa.cool)</li> </ul>
	CLDAP		searchRequest				<ul> <li>and item: equalityMatch (3)</li> </ul>
	CLDAP				DOT>" searchResDone(3	<li>39) succes</li>	
	CLDAP		searchRequest				0000 00 1a 4a 62 eb 77 5c 5e ab 7c ea 81 86 dd 60 00)b.w\^
:1a9e	CLDAP	223	searchResEntry	(400) " <r0< td=""><td>DOT&gt;" searchResDone(4)</td><td>10) succes</td><td>0010 00 00 00 f8 11 3d 2a 02 27 70 00 17 00 00 02 1a=*. 'p 0020 4a ff fe e5 1a 9e 2a 02 27 70 00 09 00 00 21 a J*. 'p</td></r0<>	DOT>" searchResDone(4)	10) succes	0010 00 00 00 f8 11 3d 2a 02 27 70 00 17 00 00 02 1a=*. 'p 0020 4a ff fe e5 1a 9e 2a 02 27 70 00 09 00 00 21 a J*. 'p
							0030 4a 11 f f 6 05 1a 96 2a 02 27 70 00 00 00 00 00 02 1a 3
							0040 00 00 00 ea 02 02 01 89 63 84 00 00 00 e8 84 00 c
							0050 0a 01 00 0a 01 00 02 01 00 02 01 00 01 01 00 a0
							0060 84 08 08 08 b9 a3 84 00 00 00 18 04 09 44 6e 73Dns 0070 44 6f 6d 61 69 6e 04 0b 78 73 2e 69 70 61 2e 63 Domain., xs.ipa.c
							0000 61 61 62 63 84 00 00 00 64 64 48 67 73 74 80 001
							0090 05 4d 45 54 49 53 a3 84 00 00 00 0d 04 84 55 73 .METISUs
							00x0 65 72 04 05 41 4c 49 43 45 a3 84 00 00 00 0b 04 er. ALIC E
							00000 03 41 41 43 04 04 10 00 00 00 a3 84 00 00 08 25 .AAC
							0000 00 00 00 00 00 05 15 00 00 00 06 55 15 21 40 07 e2
							00e0 c6 4f 7c fb 44 a3 84 00 00 00 0d 04 05 4e 74 56 .0].DNtV
							00f0 65 72 84 84 16 80 80 01 a3 84 00 00 82 84 8b er
							0100 44 6e 73 48 6f 73 74 4e 61 6d 65 04 11 6d 65 74 DnsHostN amemet 0110 69 73 2e 61 64 2e 69 70 61 2e 63 6f 6f 6c 30 84 is.ad.ip a.cool0.
							0120 09 09 09 09 09 04 04 05 74 65 77 67 6 6 Ne tloon
						,	



### WHO IS ALICE?



Alice is a fine random name, along with Marvin, Heather, Student, User2, Test, and many others seen on the wire The state of user inquiries in Windows is ... interesting



### **GLOBAL CATALOG SERVICE**

#### If Global Catalog service is available, Windows will attempt to connect to it:

	Protocol L	enath	Info		<ul> <li>sname-string: 2 items</li> </ul>
	CLDAP		searchRequest(3487) " <root>" baseObject</root>		SNameString: krbtgt
	CLDAP		searchResEntry(3487) " <root>" searchResDone(3487) success [1 resul</root>		SNameString: XS.IPA.COOL
	CLDAP		searchRequest(3488) " <root>" baseObject</root>		enc.part
	CLDAP		<pre>searchResEntry(3488) "<root>" searchResDone(3488) success [1 resul</root></pre>		<ul> <li>authenticator</li> </ul>
	CLDAP		searchRequest(3489) " <root>" baseObject</root>		etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
	CLDAP		<pre>searchResEntry(3489) "<root>" searchResDone(3489) success [1 result</root></pre>		cipher: bfbed62f2c28f01f632810862bcf9f42ebca06aa555aab0e
62:eb77			NBSS Continuation Message		<ul> <li>req.bdy</li> </ul>
ee5:1a9e			40050 - 445 [ACK] Seg=1 Ack=2 Win=351 Len=0 TSval=3839598215 TSecr=		Padding: 0
	CLDAP		searchRequest(3498) " <root>" baseObject</root>		<ul> <li>kdc-options: 40810000 (forwardable, renewable, canonicalize)</li> </ul>
	CLDAP	208	searchResEntry(3490) " <root>" searchResDone(3490) success [1 result</root>		realm: XS.IPA.COOL
	CLDAP		searchReguest(3491) " <root>" baseObject</root>		▼ shane
	CLDAP	208	<pre>searchResEntry(3491) "<root>" searchResDone(3491) success [1 result</root></pre>		name-type: kRB5-NT-SRV-INST (2)
62:eb77	TCP	86	50822 - 3268 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MSS=1440 MS=256 S.		✓ sname-string: 3 items
005:1090	TCP	86	3268 - 50822 [SYN, ACK] Seq=0 Ack=1 Win=28800 Len=0 MSS=1440 SACK PI		SNameString: tdap
62:eb77	TCP	74	50822 - 3268 [ACK] Seg=1 Ack=1 Win=263424 Len=0		SNameString: nyx.xs.ipa.cool
	TCP	66	50823 - 88 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACI		SNameString: xs.ipa.cool
	TCP	66	88 - 50823 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERU		till: 2037-09-13 02:48:05 (UTC)
	TCP		50823 → 88 [ACK] Seq=1 Ack=1 Win=2102272 Len=0		nonce: 1668017033
					etype: 5 items
	TCP	54	88 - 50823 [ACK] Seq=1 Ack=1581 Win=32384 Len=0		✓ enc-authorization-data
	KRB5		TGS-REP		etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
	TCP		88 - 50823 [FIN, ACK] Seq=1600 Ack=1581 Win=32384 Len=0	4	
	TCP		50823 → 88 [ACK] Seq=1581 Ack=1601 Win=2102272 Len=0	0060	
	TCP		50823 → 88 [FIN, ACK] Seq=1581 Ack=1601 Win=2102272 Len=0	0070	
	TCP		88 - 50823 [ACK] Seq=1601 Ack=1582 Win=32384 Len=0	8898	05 a1 8d 1b 8b 41 44 2e - 49 50 41 2e 43 4f 4f 4cAD. IPA.COOL
:62:eb77			bindRequest(31) " <root>" sasl</root>	00a0	
ee5:1a9e			3268 - 50822 [ACK] Seq=1 Ack=1682 Win=32256 Len=0	00b0 00c0	
ee5:1a9e			bindResponse(31) saslBindInProgress	0000	
62:eb77			SASL GSS-API Integrity:	8868	ec 6d 59 88 ed 48 36 d5 6f ad 65 e7 89 db 56 3f .nYH6. o.eV?
ee5:1a9e			3268 - 50822 [FIN, ACK] Seq=205 Ack=1879 Win=35072 Len=0	00f0 0100	
:62:eb77			50822 - 3268 [ACK] Seq=1879 Ack=286 Win=263168 Len=0	8118	
62:eb77			50822 - 3268 [FIN, ACK] Seq=1879 Ack=206 Win=263168 Len=0	8128	11 11 18 6c a8 b9 18 d0 02 06 10 cc b2 cd 09 Sbl
ee5:1a9e 62:eb77			3268 - 50822 [ACK] Seq=206 Ack=1880 Win=35072 Len=0	0130	
ee5:1a9e			50824 - 3268 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MSS=1440 WS=256 S. 3268 - 50824 [SYN, ACK] Seq=0 Ack=1 Win=28800 Len=0 MSS=1440 SACK PI	8148 8158	
62:eb77			3268 = 58824 [51N, ALK] Seq=0 ACK=1 WIN=28800 Len=0 PSS=1440 SACK P1 50824 → 3268 [ACK] Seq=1 ACK=1 WIN=2108160 Len=0	0160	e8 4c 58 9e d3 cb fd 20 4b ef 2c f8 11 cf 48 df .LX K.,H.
62:eb77			bindRequest(35) " <root>" sasl</root>	8178	
02:0077			Dinomequest(55) "Route sast	<ul> <li>0180</li> <li>8198</li> </ul>	
				0198	



### **GLOBAL CATALOG SERVICE SEARCH RESULTS**

## If search in a Global Catalog returns no results, Windows falls back to netr\_LogonSamLogonWithFlags RPC call:

	Protocol Leng	th Info	Frag Length: 205
62:eb77	TCP	74 50826 - 3268 [FIN, ACK] Seg=1955 Ack=206 Win=2107904 Len=0	Auth Length: 37
ee5:1a9e		74 3268 - 50826 [RST] Seg=206 Win=0 Len=8	Call ID: 109
62:eb77	TCP	86 58827 - 3268 [SYN, ECN, CMR] Seq=0 Win=8192 Len=0 MSS=1440 MS=256 S.	Max Xmit Frag: 5840
ee5:1a9e		86 3268 - 50827 [SYN, ACK] Seg=0 Ack=1 Win=28800 Len=0 MSS=1440 SACK P	Max Recv Frag: 5840
62:eb77		74 50827 - 3268 [ACK] Seg=1 Ack=1 Win=263424 Len=0	Assoc Group: 0x00000000
:62:eb77	LDAP 17	55 bindRequest(47) " <root>" sasl</root>	Num Ctx Items: 3
'ee5:1a9e	TCP	74 3268 - 50827 [ACK] Seg=1 Ack=1682 Win=32256 Len=0	Ctx Item[1]: Context ID:0, RPC_NETLOGON, 32bit NDR
ee5:1a9e	LDAP 2	78 bindResponse(47) saslBindInProgress	Ctx Item[2]: Context ID:1, RPC_NETLOGON, 64bit NDR Ctx Item[3]: Context ID:2, RPC_NETLOGON, Bind Time Feature Negotiation
62:eb77	LDAP 2	71 SASL GSS-API Integrity:	Auth type: NETLOGON Secure Channel (68)
ee5:1a9e		74 3268 - 50827 [FIN, ACK] Seq=205 Ack=1879 Win=35072 Len=0	Auth level: Packet privacy (6)
62:eb77	TCP	74 50827 - 3268 [ACK] Seq=1879 Ack=206 Win=263168 Len=0	Auth pad len: 0
62:eb77	TCP	74 50827 - 3268 [FIN, ACK] Seq=1879 Ack=286 Win=263168 Len=0	Auth pad ten: 0 Auth Rsrvd: 0
ee5:1a9e		74 3268 - 50827 [ACK] Seq=206 Ack=1880 Win=35072 Len=0	Auth Context ID: 0
62:eb77		86 50828 - 3268 [SYN, ECN, CMR] Seq=0 Win=8192 Len=0 MSS=1440 MS=256 S.	Secure Channel NL AUTH MESSAGE
ee5:1a9e		86 3268 - 50828 [SYN, ACK] Seq=0 Ack=1 Win=28800 Len=0 MSS=1440 SACK_P	Message Type: Request (0x00000000)
162:eb77		74 50828 → 3268 [ACK] Seq=1 Ack=1 Win=263424 Len=0	Message Flags: 0x000000017, NetBios Domain, NetBios Host, DNS Domain, NetBios Host(UTF8)
162:eb77		'55 bindRequest(51) " <root>" sasl</root>	NetBios Domain: XS
ee5:1a9e		74 3268 → 50828 [ACK] Seq=1 Ack=1682 Win=32256 Len=0	NetBios Host: METIS
'ee5:1a9e		78 bindResponse(51) saslBindInProgress	DNS Domain: xs.ipa.col
:62:eb77		71 SASL GSS-API Integrity:	NetBios Host(UTFR): NETIS
ee5:1a9e		74 3268 - 50828 [FIN, ACK] Seq=205 Ack=1879 Win=35072 Len=0	
:62:eb77		74 50828 - 3268 [ACK] Seq=1879 Ack=206 Win=263168 Len=0	0000 00 1a 4a 62 eb 77 5c 5e ab 7c ea 81 86 dd 60 00
62:eb77		50 SASL GSS-API Integrity:	0010 00 00 00 01 06 3d 2a 02 27 70 00 17 00 00 02 1a=*. 'p
ee5:1a9e		74 3268 → 50828 [RST] Seq=206 Min=0 Len=0	0020 4a ff fe e5 1a 9e 2a 02 27 70 00 09 00 00 02 1a J*. 'p 0030 4a ff fe 62 eb 77 c6 8d 04 00 55 05 91 46 11 03 Jb.wU.F
		81 searchRequest(3492) " <root>" baseObject</root>	0040 44 11 10 02 00 77 C0 80 04 00 35 05 01 40 11 05 3. 0. 0
and the second second		08 searchResEntry(3492) * <root>* searchResDone(3492) success [1 resul</root>	0050 00 00 cd 00 25 00 6d 00 00 00 d0 16 d0 16 00 00%.m
62:eb77		86 50829 - 1024 [SYN, ECN, CMR] Seq=0 Win=8192 Len=0 MSS=1440 MS=256 S.	0000 00 00 03 00 00 00 00 00 01 00 78 56 34 12 34 12
ee5:1a9e		86 1024 → 50829 [SYN, ACK] Seq=0 Ack=1 Win=28800 Len=0 MSS=1440 SACK_PI 74 50829 = 1024 [ACK] Seq=1 Ack=1 Win=263424 Len=0	0070 cd ab ef 00 01 23 45 67 cf fb 01 00 00 00 04 5d#Eg] 0080 88 8a eb 1c c9 11 9f e8 08 00 2b 10 48 60 02 00
62:eb77		74 50829 = 1824 [ACK] Seq=1 ACK=1 Win=263424 Len=0 79 Bind: call id: 109, Fragment: Single, 3 context items: RPC NETLOGON :	0090 00 00 01 00 01 00 78 56 34 12 34 12 cd ab ef 00xV 4.4
ee5:1a9e		74 1024 - 50829 [ACK] Seg=1 Ack=206 Win=29952 Len=0	0030 01 23 45 67 cf fb 01 00 00 03 05 71 71 ba be .#Eg3.qq
ee5:1a9e		56 Bind ack: call id: 109. Fragment: Single, max xmit: 4280 max recv: -	00b0 37 49 83 19 b5 db ef 9c cc 36 01 00 00 02 00 716 00c0 01 00 78 56 34 12 34 12 cd ab ef 00 01 23 45 67
		00 Bind_ack: call_10: 109, Fragment: Single, max_xmit: 4280 max_recv: - 00 NetrLogonSamLogonWithFlags request	0000 01 00 76 50 54 12 54 12 54 12 50 40 01 00 01 23 45 01 1.544.4
		98 NetrLogonSamLogonWithFlags response	00e0 00 00 00 00 08 08 01 00 00 00 44 05 00 00 08 08
62:eb77		74 50829 - 1824 [ACK] Seq=822 Ack=517 Win=262912 Len=0	00f0 00 00 00 00 00 01 70 00 00 058 53 00 4d 45 54
10212077		82 searchRequest(3493) " <root>" baseObject</root>	0110 05 44 55 44 9 53 00
		08 searchResEntry(3493) * <root>" searchResDone(3493) success [1 resul</root>	
< C		a parente sentity and a sentence beneficially added a transfer	



### **NETLOGON RESULTS**

#### But the name passed to netr\_LogonSamLogonWithFlags is totally different from

#### what is entered in Windows UI

[2017/05/03 13:58:05.818919. 5. pid=17774. effective(0. 0). real(0. 0)] ../source3/lib/smbldap.c:1249(smbld smbldap search ext: base => [dc=xs.dc=ipa.dc=cool], filter => [(&(objectClass=ipaNTUserAttrs)(uid=USER))]. [2017/05/03 13:58:05.858058, 4, pid=17774, effective(99, 99), real(99, 0)] ../source3/smbd/sec ctx.c:439(po pop sec ctx (99, 99) - sec ctx stack ndx = 1[2017/05/03 13:58:05.858094, 3, pid=17774, effective(99, 99), real(99, 0), class=auth] ../source3/auth/chec check sam security: Couldn't find user 'USER' in passdb. [2017/05/03]13:58:05.858119. 5. pid=17774. effective(99. 99). real(99. 0). class=auth] ../source3/auth/auth check ntlm password: sam authentication for user [USER] FAILED with error NT STATUS NO SUCH USER [2017/05/03 ]3:58:05.858171, 10. pid=17774, effective(99, 99), real(99, 0), class=auth]../source3/auth/auth Check auth for: [USER] [2017/05/03 13:58:05.858185, 3. pid=17774, effective(99, 99), real(99, 0), class=auth] ../source3/auth/auth check winbind security: Not using winbind, requested domain [XS] was for this SAM. [2017/05/03 13:58:05.858195, 10, pid=17774, effective(99, 99), real(99, 0), class=auth] ../source3/auth/auth check ntlm password: winbind had nothing to sav [2017/05/03 13:58:05.858204, 2, pid=17774, effective(99, 99), real(99, 0), class=auth] ../source3/auth/auth check ntlm password: Authentication for user [USER] -> [USER] FAILED with error NT STATUS NO SUCH USER [2017/05/03 ]3:58:05.858217. 5. pid=17774. effective(99. 99). real(99. 0). class=rpc\_srv] ../source3/rpc\_se netr LogonSamLogonWithFlags: check password returned status NT STATUS NO SUCH USER [2017/05/03 13:58:05.858229, 4, pid=17774, effective(99, 99), real(99, 0)] .../source3/smbd/sec ctx.c:217(pu Reg redhat.

### NAME RESOLUTION ORDER



### NAME RESOLUTION ORDER

- CLDAP ping is an important operation but actual name resolution is done by querying Global Catalog
- If Global Catalog available, Windows will try to use that
- If search in Global Catalog does return no results, Windows will fall back to RPC calls
- FreeIPA does not provide a Global Catalog service
- As result, Windows does not even try to fall back to RPC calls



### **FREEIPA CHALLENGES**



### SASL GSS-SPNEGO

Samba implements own SASL code, FreeIPA components rely on Cyrus-SASL

- Cyrus-SASL GSS-SPNEGO implementation is compatible with itself, not Windows
- GSS-SPNEGO negotiates SSF based on GSSAPI flags, not separately
- This was fixed by Simo Sorce in February 2017: https://github.com/cyrusimap/cyrus-sasl/commit/ 67ca66685e11acc0f69d5ff8013107d4b172e67f
- No Cyrus-SASL release with the fix yet but Fedora 26 has it backported



### **KERBEROS TARGET PRINCIPALS**

#### Windows TGS-REQ requests use three-component principal names:

- ldap/host.example.com/example.com@EXAMPLE.COM
- Real service name is ldap/host.example.com@EXAMPLE.COM
- FreeIPA 4.4+ added support for Kerberos principal aliases: ipa service-add-principal ldap/host.example.com@EXAMPLE.COM ldap/host.example.com/example.com@EXAMPLE.COM



### LDAP SASL BIND MAPPING

Windows always uses SASL GSS-SPNEGO for LDAP bind authentication

- Successful authentication means LDAP server needs to map authenticated identity to existing LDAP object
- There are no users or machines accounts from a trusted Active Directory in FreeIPA LDAP store
- Luckily, Global Catalog access for out-of-domain accounts is read-only
- We can map all authenticated but unknown identities to a single LDAP object with read-only rights



## LDAP SCHEMA

- FreeIPA has its own LDAP schema and LDAP tree strucutre
- Active Directory LDAP schema is not compatible with FreeIPA LDAP schema
- Attributes and objects can be re-mapped but direct access is useless



### **389-DS LIMITATIONS**

- 389-ds LDAP server only allows to listen on a single port per protocol
- TCP/389 for LDAP, TCP/636 for LDAPS
- Global Catalog is always TCP/3268 for LDAP access



### FREEIPA GLOBAL CATALOG SERVICE



### **GLOBAL CATALOG SERVICE**

- Runs as a separate 389-ds instance to serve port TCP/3268
- Transforms user and group data from primary FreeIPA LDAP instance to AD schema
- Access is read-only with SASL GSS-SPNEGO authentication



### DATA TRANSFORMATION

- LDAP SYNCREPL is used to pick up changes from the primary FreeIPA LDAP instance running on the same IPA master
- Schema Compatibility plugin code is used to transform the changes to AD-compatible schema and DIT
- https://pagure.io/slapi-nis/



### **CURRENT STATE**

- Design documents are available at https://www.freeipa.org/page/V4/Global\_Catalog\_Support
- SYNCREPL plugin almost ready
- Schema Compatibility plugin refactoring has started
- We plan to have working prototype ready for Redmond IOLab in June 2017





### THANK YOU

https://samba.org/ https://freeipa.org/