Your Server Will Be With You Shortly: Samba and Chrome OS

Jeremy Allison
Samba Team/Google Open Source Programs Office

jra@samba.org
jra@google.com

What is Chrome OS ?

Chrome OS

- Chrome OS is a **managed**, **single-user** desktop environment created by Google.
- All the hard parts of integrating a Linux desktop are not done on the Chrome OS box itself.
  - All set-up is remotely managed.
  - Normally devices are joined to AD before being given to users.
- **Single** user means no winbind needed – no real users.
  - No real user data held locally, everything accessed via cloud.
  - Remote SMB share access available, but use case is the device can be re-imaged at any time.

Chrome OS and Samba

Chrome OS

- Chrome OS uses Samba for two important features.
- 1). Active Directory integration.
  - This is complex.
  - Samba has a long history of (mostly) doing this right.
  - Kerberos only. No NTLM fallbacks allowed here.
- 2). Remote SMB fileshare access for local networks.
  - Samba has a long history of doing this right.

The Chrome OS / Active Directory Logon Process

- net ads workgroup
  - Get the workgroup info for the realm.
- net ads info
  - Get the KDC ip address and time.
- net ads lookup
  - CLDAP request to get the KDC name.
- kinit
  - Get the TGT.

The Chrome OS / Active Directory Logon Process (continued)

- net ads search “(sAMAccountName=user)”
  - Get the user affiliation.
- net ads gpo list
  - Get group policy
  - Parse output to feed into..
- smbclient
  - Download group policy files and apply locally.

Sandboxing can make preserving caches difficult.

Chrome OS quirks

- For security purposes, Chrome OS uses a “allow list” of system calls that can be configured per-binary (seccomp).
- “System” services like Samba are invoked via inter-process communication – DBUS requests.
- Run under “minijail” as a separate user-id.
  - minijail restricts file system access.
  - Custom config files have to be created and passed to invoked binaries.
  - Means many Samba “normal OS” assumptions (can store name → IP address mapping in caches etc.) no longer hold true.

The Start of the problem

- A **large** customer complained that on one remote site, no Active Directory users could log in.
  - All other sites worked fine.
- On entering login credentials, the box spun its wheels for 4 minutes and then went back to the logon screen.
- What is different about this site ?
  - No local DNS server.
- Probably DNS lookup issues.
  - What information can we get from the customer box ?

This should be easy

- Incredibly helpful and knowledgeable customer IT staff.
  - Able to get wireshark traces between Chrome OS and servers.
  - No interactive debugging allowed, but..
- Chrome OS can return Samba tool debug level 10 logs.
  - Available via a simple terminal command.
  - Creates a zip file containing all system logs.

This should be easy (continued)

- If it’s a DNS latency issue, should be easily solvable via caching in Open Source dnsmasq caching DNS resolver code.
- For an earlier reported problem, I added SRV record (widely used to find AD-DC’s) caching to dnsmasq for v2.81.
  - Oh. Turns out dnsmasq wasn’t added to Chrome OS due to concerns about using it as a system-wide solution.

The nightmare unfolds

- Initial logs show DNS SRV record lookup for name “_kerberos._tcp.<CUSTOMER.NAME>” returns over 200+ names.
  - Returned names do not have associated IP addresses returned in the DNS SRV record reply.
  - This means we have to now do DNS name → IP address queries.
- We do this **sequentially** using getaddrinfo().
  - For A (IPv4) records.
  - And AAAA (IPv6 records).
- We don’t do anything until all names are resolved :-(.
  - But but but.. we only need **ONE** working server.

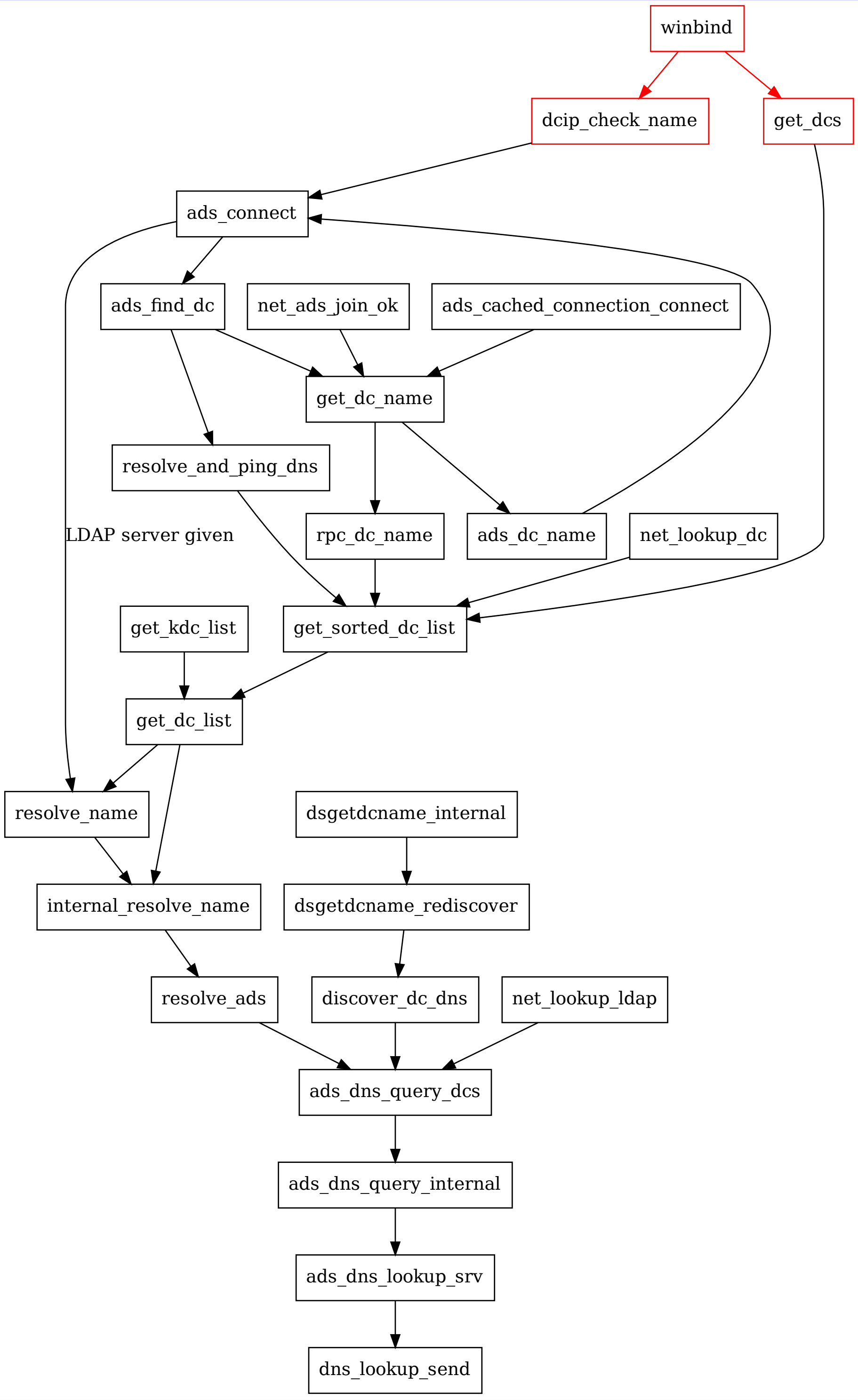
Simple problem statement

- Make DNS name lookup in Samba fast, resilient and scalable to hundreds of DNS servers and thousands of simultaneous names for both IPv4 and IPv6 look-ups.
- This *SHOULD* be a job for the underlying operating system.
  - POSIX sucks, yet again :-(.
  - getaddrinfo() is not quite fit for purpose.
  - Neither is getaddrinfo_a() (wraps getaddrinfo() in a threadpool inside glibc).
- What should a DNS name lookup API look like ?
  - See the end of this talk for my ideas :-).

When you’re in a hole.. start digging into the code !

- Sernet Samba Team member Volker Lendeke already anticipated this problem – built on top of earlier work by Kai Blin.
    - Our DNS record lookup code (dns_lookup_send / recv) is modern, asynchronous, and can contact multiple DNS servers in parallel.
    - It’s also not being used in the ‘net ads’ code in the version in Chrome OS :-(.
- Maybe I can plumb this modern code into the Samba code paths used by Chrome OS ?

Old, Over-designed code (thanks to Sernet Samba Team member Ralph Böhme for the image)



Frantic coding (3 weeks)

- Now is the chance to fix some really old code dealing with name look-ups.
  - First, fix the caching code to move everything to talloc(). Hide this under the guise of the bugfix :-).
- Re-use the existing async DNS lookup code and plumb into name resolution code inside namequery.c
  - This was much easier than expected, the async DNS code APIs inside Samba are really nice.
- Default 10-second timeout added.
  - Collect all the AD-DC addresses you can within that time.
  - Remember we only need one working one.

Overreach

- New function dns_lookup_list_async() can be used to map any array of names to IPv4 or IPv6 addresses.
  - Queries all known DNS servers with all requested names simultaneously.
  - Configurable timeout means we can limit how long we’ll wait for answers.
- This could replace *ALL* name resolution in Samba.
  - Or not :-(. I came close, but could never get a full ‘make test’ to pass.
  - Culprit was resolv_wrapper that “mocks” DNS lookups by interposing at the glibc layer.
  - Hacking our python DNS server nearly made it work.

Why doesn’t it work ?

- Lots of local testing. Test framework added. Delivered to customer.
- Customer cannot login :-(.
- What did we miss ?
  - Logs saved us (again).
  - New code uses readv() system call when falling back from UDP → TCP DNS look-ups (large replies).
  - Minijail had read() in the allow list, but not readv().
    - Well that should be an easy fix.
- Customer still cannot login :-(.

Now why doesn’t it work ?

- Chrome OS issues this time
  - Not everything is Samba’s fault, thank goodness.
- User on problematic site is attempting to login to trusted domain.
  - Configuration code setting up Samba database files for a joined domain member needs a Domain SID for the named domain.
  - Chrome OS framing code wasn’t setting this up for the trusted domain.
    - Note this domain SID isn’t used at all in Chrome OS, but the Samba code expected it to be there
- Customer still cannot login :-(:-(.

Drop, drop, DROP !

https://www.youtube.com/watch?v=WsrVw9Jwtio

Work, damn you, work, Work, WORK !

- All Samba code seems to be working.
- kinit command is taking forever.
  - Wireshark traces are the key.
- MIT krb5 library code is **ALSO** doing SRV lookups..
  - For _kerberos._udp.<CUSTOMER.REALM>
  - Then _kerberos._tcp.<CUSTOMER_REALM>
- And then looking up every name returned via getaddrinfo for IPv4 (A) then IPv6 (AAAA).
- It’s doing this three times :-(.

Red Hat to the rescue

- In 2007 Red Hat Samba Team member Guenther Deschner wrote an MIT krb5 “KDC Locator plugin” for Samba.
    - Purpose was to ask winbind for the closest KDC.
    - Now winbind uses async DNS to locate KDC’s this would fix the problem.
- But Chrome OS doesn’t have winbind.
    - I hacked Guenther’s code to create an async DNS KDC locator that directly calls internal Samba function get_kdc_list().
- Customer can logon :-). Only 2 months later :-).

Lessons learned the hard way

- 1). Logging, logging, logging.
  - Without comprehensive logs this bug could not have been fixed.
- 2). Source code needed.
  - If MIT krb5 had been a proprietary library, this bug could not have been fixed.
- 3). Good customer network debugging.
  - Without full wireshark traces, this bug could not have been fixed.
- 4). Hire Open Source engineers :-).
  - Without a Samba Team member at Google, this bug could not have been fixed.

Getting out of the DNS client business

- POSIX DNS interfaces **suck**.
- What should they look like ?
  - Systemd is the key - resolvectl.c may already have what we need.
- Asynchronous inter-process communication (IPC) to a system daemon that can hide all the ugly hard code.
  - DNS over TLS, DNSSEC etc.
- File-descriptor based allows epoll/poll/kqueue to notify the caller to pick up results.
  - Re-use getaddrinfo structures for easy adoption.

Questions and Comments ?

Email: jra@samba.org
jra@google.com

Slides available at: