

# Samba and Chrome OS

the Start of a beautiful Friendship

Lutz Justen

[ljusten@google.com](mailto:ljusten@google.com)

sambaXP, Göttingen  
June 06, 2018

# Topics

Chrome OS and Chromebooks

Active Directory Integration

How it works, management, Android apps, certificates, file shares

Under the Hood

D-Bus interface, Samba usage, Kerberos integration, Sandboxing

Summary and Future Plans

# Chrome OS

Chrome OS is a **Linux-based OS** built around **Chrome**

Designed based on the 3S: **Simplicity, Security** and **Speed**

# Chromebooks - History

First Chromebooks shipped in 2011

Today more than 50 models, different form factors

Popular in US schools with ~60% market share\*

\* Source: <https://www.zdnet.com/article/windows-pcs-gain-share-in-k-12-in-the-u-s-but-chromebooks-still-dominate/>

# Chromebooks - Evolution

## First Chromebooks

“Laptops running Chrome”

## Today's Chromebooks

Run Android apps ([ARC++](#))

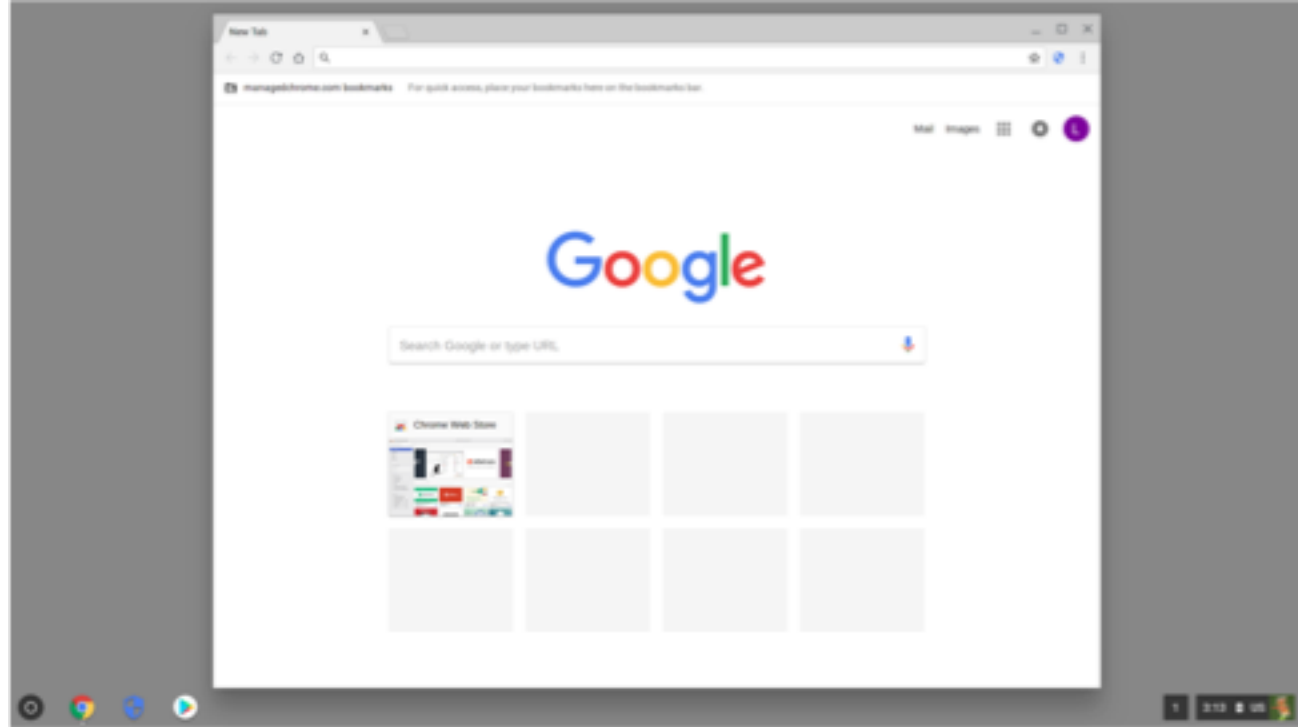
Run Linux apps ([Crostini](#))

Run Windows apps ([CrossOver](#))

# Chrome OS and Chromebooks



**Google Pixelbook**



# Chrome OS and Chromebooks

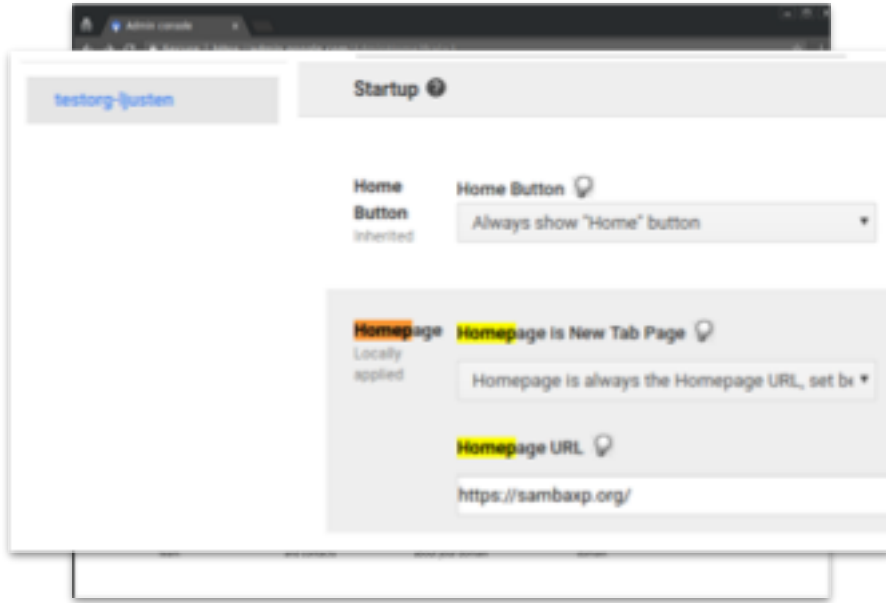
Requires a **Google account**

Can be unmanaged (private) or managed (enterprise, schools)

Management via **cloud-based console**

# Cloud-Based Management

## Management Console



## Chromebook





# Chromebooks in Enterprises

*I'm an Active Directory admin and  
I want to try out Chromebooks in my company*

Requires a Google account

- Not tied to enterprise identity
- Could use [sync tool](#) to create Google accounts for employees and sign in with SAML
- Admins might not want to/be able to share employee data with Google

Separate management (Active Directory GPO + Google Cloud)

Companies might not want all/any Google services

**Large up-front investment!**

# Active Directory Integration

***Goal: Make it easy to use Chromebooks in Active Directory environments***

No Google account necessary

Management via Active Directory Group Policy

Launched Aug 2017 as part of [Chrome Enterprise](#)

Under the hood: **Samba** in sandboxed system daemon

# AD Integration - How it works

Step 1: [Register domain](#) with Google

One-pager

Mainly for license counting, config

All steps on [Help Center](#)



The screenshot shows the 'Chrome Microsoft® Active Directory® Integration' sign-up page. At the top right is an image of a computer monitor. The page title is 'Chrome Microsoft® Active Directory® Integration'. Below the title is the text: 'Sign-up to integrate your Chrome devices with Microsoft® Active Directory®'. The form contains the following fields: 'First name' and 'Last name' (two separate input boxes), 'Email' (with a hint 'e.g. john@mydomain.com'), 'Phone' (with a hint '+16025555555'), 'Business name' and 'Country' (a dropdown menu showing 'Germany'), 'Preferred account name' (with a hint '.deviceadmin.goog'), 'Username' (with the text 'admin' entered) and '@ deviceadmin.goog' (a dropdown menu). At the bottom, there is a checkbox and the text: 'By checking this checkbox, you are indicating that you have read and agreed to the Google Subdomain License Agreement and meet the eligibility requirements.' Below this is a button labeled 'ACCEPT AND CREATE YOUR ACCOUNT'.

# AD Integration - How it works

Step 2: On fresh Chromebook

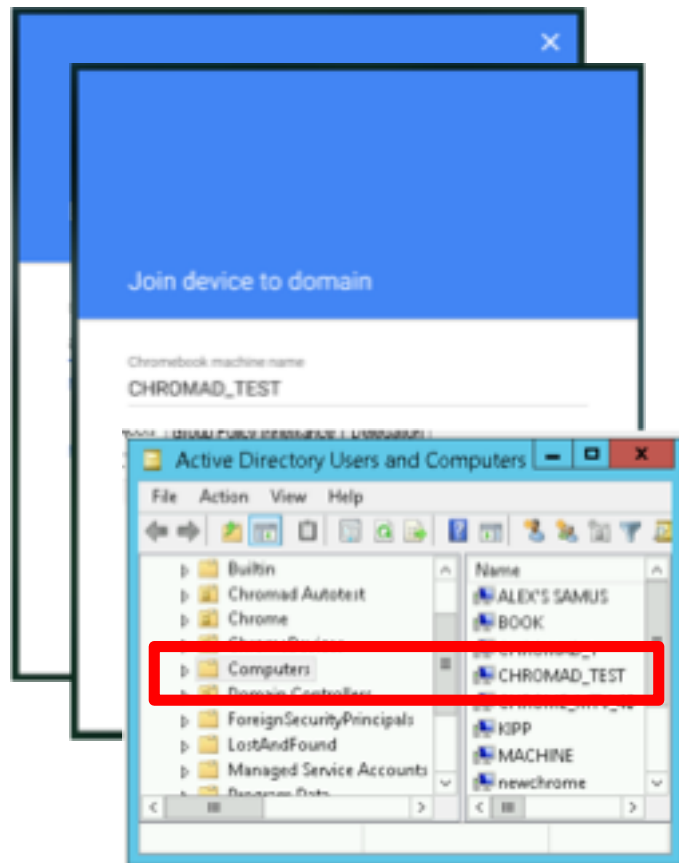
Press CTRL+ALT+E to enroll

Enter Google domain credentials

Enter Active Directory credentials +  
computer name

Computer shows up in  
Active Directory

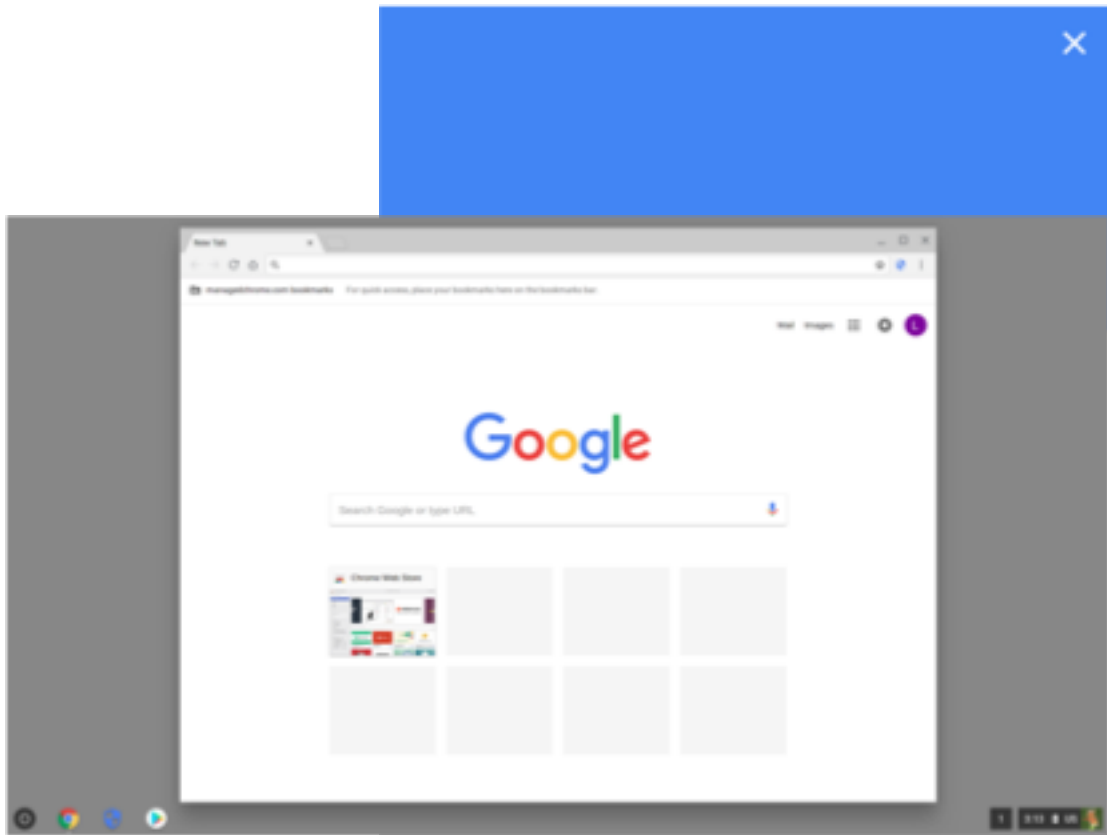
All steps on [Help Center](#)



# AD Integration - How it works

Step 3: Log in with Active  
Directory  
credentials

All steps on [Help Center](#)  
Google



# AD Integration - Features

Identity tied to Active Directory

Handles password change

Management via Group Policy

Kerberos SSO

Android apps

Support for certificates, file shares, printing

# AD Integration - Management

Managed by Group Policy

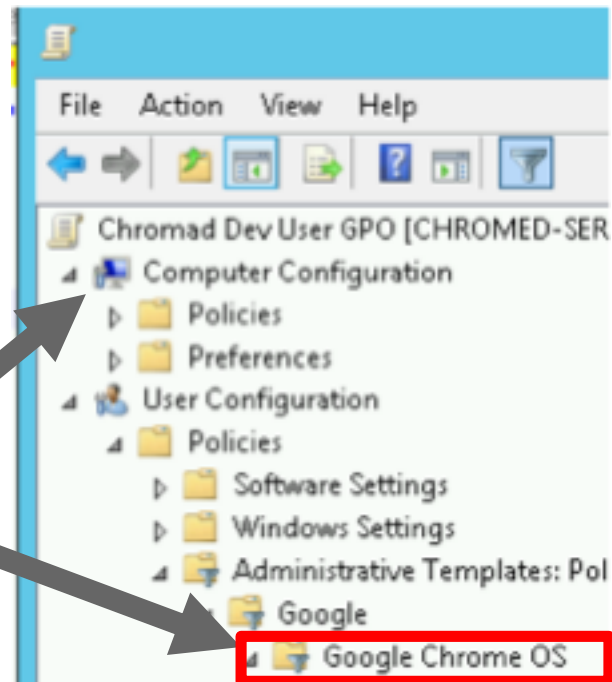
[Download](#) and install Chrome OS ADMX templates

Edit policies in Active Directory

Group Policy Object (GPO) editor

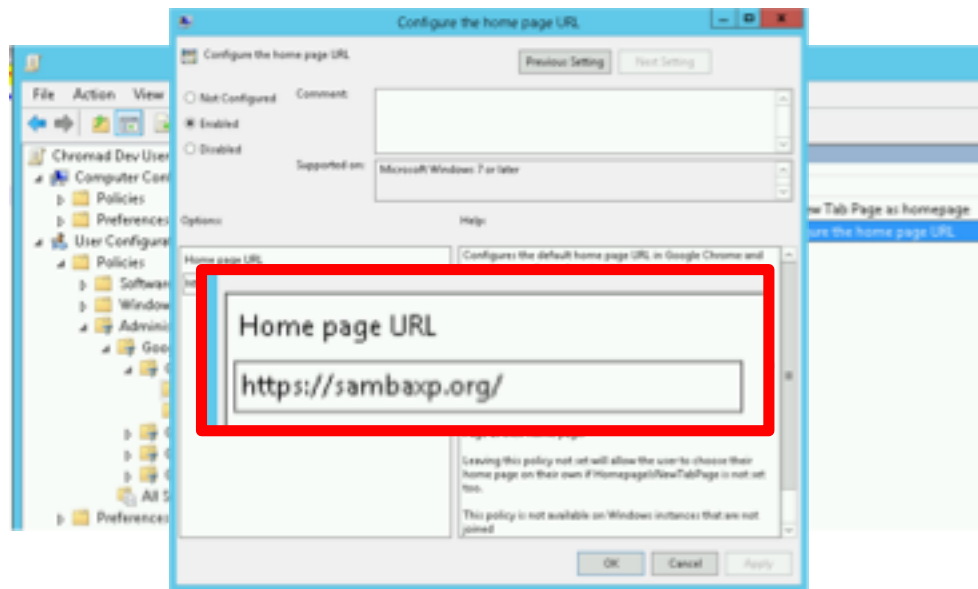
applies to device account (Chrome OS device policy)

applies to user accounts (Chrome OS user policy)

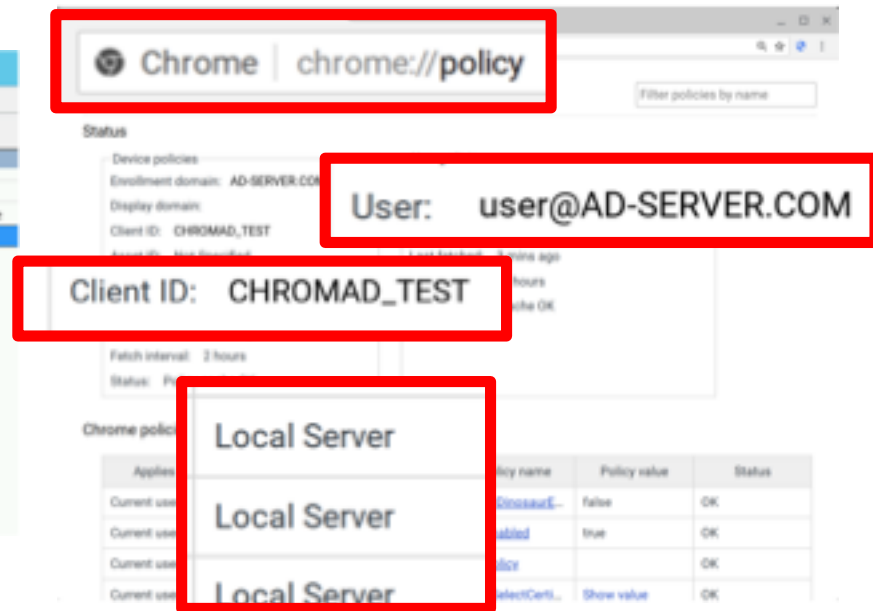


# AD Integration - Management

## GPO Editor



## Chromebook





# AD Integration - Management

JSON for complex policies, e.g.  
Default printer selection rules

## Default printer selection rules

```
{ "kind": "cloud",  
  "idPattern": ".*public",  
  "namePattern": ".*Color" }
```

Default printer selection rules

Previous Setting Next Setting

☐ Not Configured Comment:

☒ Enabled

☐ Disabled Supported on: Microsoft Windows 7 or later

Options:

Default printer selection rules (The single-line field is deprecated and will be removed in the future. Please start using the multi-line textbox below.)

Default printer selection rules

```
{ "kind": "cloud",  
  "idPattern": ".*public",  
  "namePattern": ".*Color" }
```

Help:

Overrides Google Chrome default printer selection rules.

This policy determines the rules for selecting the default printer in Google Chrome which happens the first time the print function is used with a profile.

When this policy is set, Google Chrome will attempt to find a printer matching all of the specified attributes, and select it as default printer. The first printer found matching the policy is selected, in case of non-unique match any matching printer can be selected, depending on the order printers are discovered.

If this policy is not set or matching printer is not found within the timeout, the printer defaults to built-in PDF printer or no printer selected, when PDF printer is not available.

The value is parsed as JSON object, conforming to the following schema:

```
{  
  "type": "object",  
  "properties": {
```

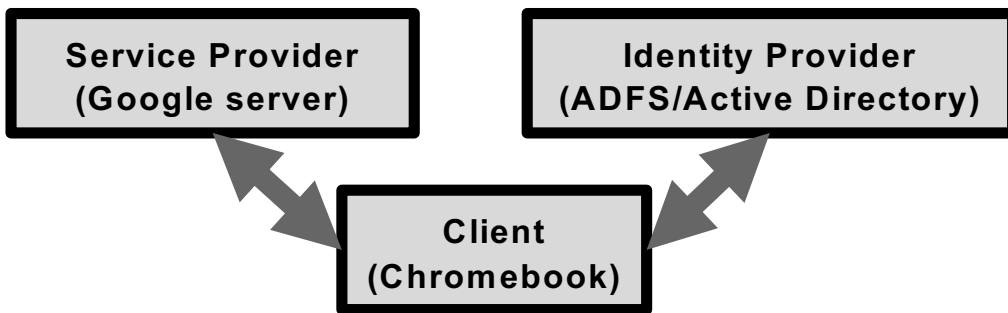
OK Cancel Apply

# AD Integration - Android Apps

Android apps are per user

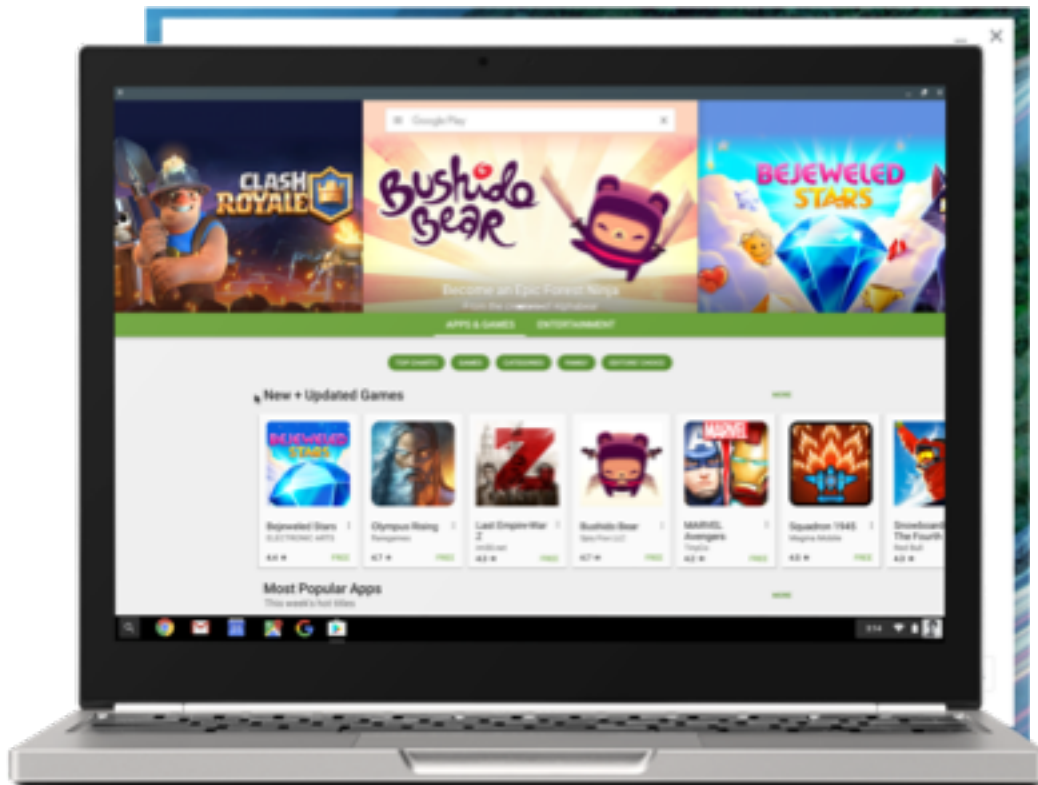
Requires [SAML setup](#) to prove user identity to Google

Google creates a shadow account with scope limited to Android apps



# AD Integration - Android Apps

SAML sign-in page  
appears on first login  
(unless Kerberos SSO is set up)



# AD Integration - Android Apps

Admin can

- Pick apps that users can install  
→ [PlayStore admin console](#)
- Force install or preload apps  
→ [ArcPolicy](#) policy
- Pin apps to launcher  
→ [PinnedLauncherApps](#) policy

# AD Integration - Certificates

## Server and Authority Certificates

→ OpenNetworkConfiguration policy (spec)

```
{
  "Type": "UnencryptedConfiguration",
  "Certificates": [
    { "GUID": "my_cert",
      "TrustBits": [ "Web" ],
      "Type": "Authority",
      "X509": "<base-64 encoded X.509 file>"
    }
  ]
}
```

# AD Integration - Certificates

## Client Certificates

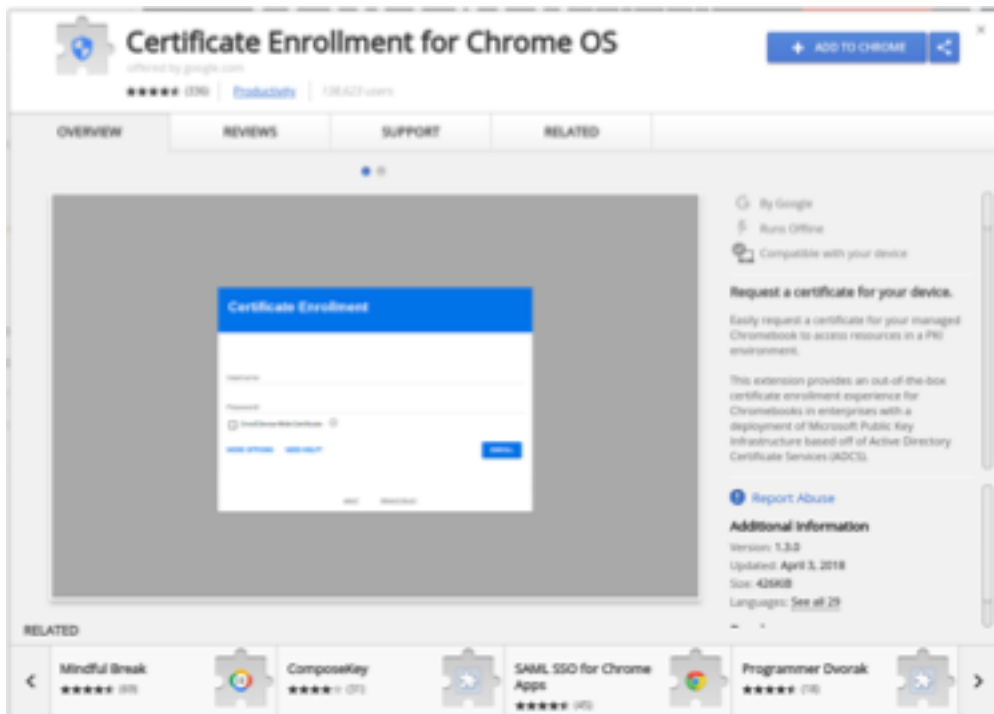
→ [Chrome OS Certificates Enrollment Chrome Extension](#)

Configured in GPO editor  
(needs ADMX templates\*)

Requests certs from ADCS

Keys are hardware-backed

\* Currently not publicly available,  
but we're working on it. Just ask for now!



# AD Integration - File Shares

Currently (being deprecated)

→ [Network File Share for Chrome OS](#)  
[Chrome extension](#)

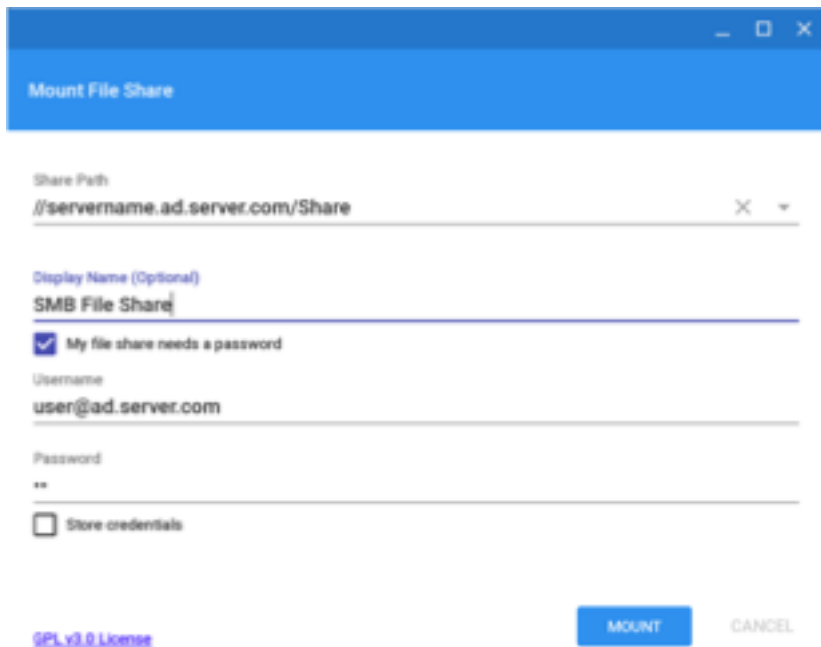
SMB file shares only

Uses Samba as well!



# AD Integration - File Shares

## Configuration in Chrome OS



The 'Mount File Share' dialog in Chrome OS is shown. It has a blue header with the title 'Mount File Share'. Below the header, there are several input fields and checkboxes. The 'Share Path' field contains '//servername.ad.server.com/Share'. The 'Display Name (Optional)' field contains 'SMB File Share'. The checkbox 'My file share needs a password' is checked. The 'Username' field contains 'user@ad.server.com'. The 'Password' field is empty and masked with two asterisks. There is a 'Store credentials' checkbox which is unchecked. At the bottom, there is a 'MOUNT' button and a 'CANCEL' button. A 'GPL v3.0 License' link is visible at the bottom left.

Share Path  
//servername.ad.server.com/Share

Display Name (Optional)  
SMB File Share

☒ My file share needs a password

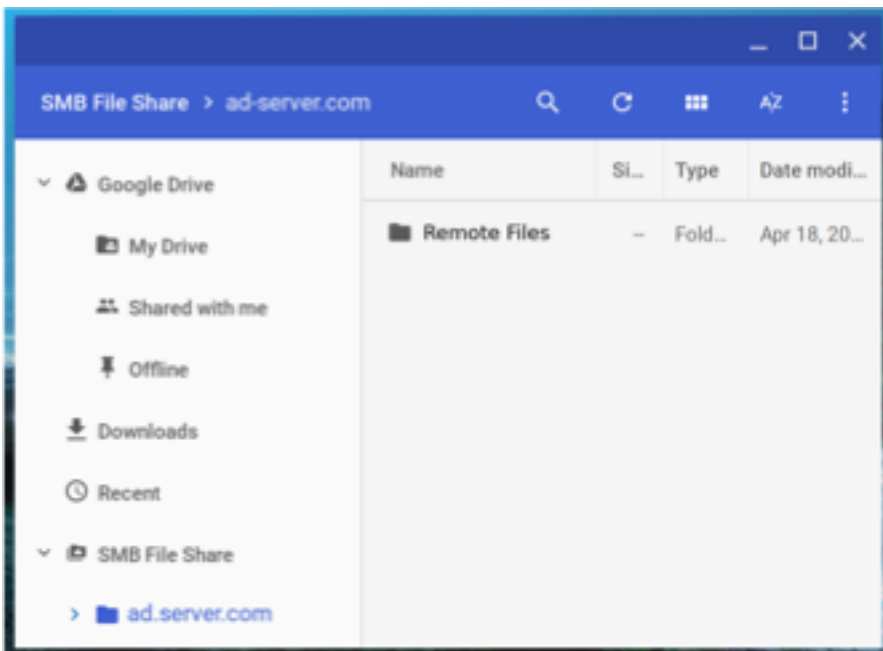
Username  
user@ad.server.com

Password  
..

☐ Store credentials

[GPL v3.0 License](#) **MOUNT** CANCEL

## Files App with SMB Share





# AD Integration - File Shares

File shares extension is being deprecated

- Slow
- Requires reauthentication every time

Under development: Native integration

- Chrome OS system daemon
- Fast
- Kerberos SSO
- Expected on beta channel in Q3

# AD Integration - Under The Hood

Chrome OS system daemon with **D-Bus interface**

Calls Samba binaries **net**, **smbclient** and **kinit**, **klist**, **kpasswd**

Manages **Kerberos ticket**

**Sandboxed** with Minijail

# AD Integration - D-Bus Interface

JoinADDomain

Joins machine to Active Directory domain

AuthenticateUser

Gets Kerberos ticket

GetUserStatus

Returns Kerberos ticket status, password status and user info

GetUserKerberosFiles

Returns Kerberos ticket and Kerberos configuration (krb5.conf)

# AD Integration - D-Bus Interface

## RefreshUserPolicy

Retrieves user policy from Active Directory and stores it securely

## RefreshDevicePolicy

Retrieves device policy from Active Directory and stores it securely

## SetDefaultLogLevel

Turns on debug logs, used by “crash” command authpolicy\_debug

# AD Integration - Samba Usage

`net ads join`

Joins machine to Active Directory domain

`net ads info`

Looks up key distribution center (KDC) IP and server time

`net ads lookup`

Looks up domain controller (DC) name

`net ads workgroup`

Looks up workgroup

# AD Integration - Samba Usage

`net ads search`

Looks up user information (first name, last name, sAMAccountName, ...)

`net ads gpo list`

Gets list of GPOs that apply to user/device account

`smbclient`

Downloads GPOs from sysvol

# AD Integration - MIT-KRB5 Usage

kinit

Gets Kerberos ticket

klist

Checks validity and lifetime of Kerberos ticket

kpasswd

Rotates machine password (every 30 days by default)

# AD Integration - Native Kerberos integration

Daemon gets Kerberos ticket during sign-in

Handles ticket renewal

Provides ticket to Chrome

- **Kerberos SSO**

Automatically signs in to pages requiring Integrated Windows Authentication through GSSAPI

- Controlled by [policies for HTTP authentication](#)



# AD Integration - Sandboxing

As every large project, [Samba has security flaws](#)

Minimize impact of security issues by reducing attack surface

In case process is hijacked, hackers have less options

**Limit what the process can do** using [Minijail](#)

# Pillars of Sandboxing I: Don't run as root

Root can do anything!

Run as non-root user and group

```
minijail0 -u user -g group /path/to/mydaemon
```

# Pillars of Sandboxing II: Only keep capabilities you need

Linux has over 30 capability flags to do root-y stuff

- CAP\_SETUID, CAP\_SETGID to set user/group
- CAP\_CHOWN to change ownership of a file

Minijail lets you keep a subset of capabilities, e.g.

**minijail0 -u user -g group -c c0 /path/to/mydaemon**

  
CAP\_SETUID | CAP\_SETGID = Bits 6, 7 = 0xc0

# Pillars of Sandboxing III: Filtering system calls

Linux has [over 300 system calls](#)

- **read, write** for file manipulation
- **connect, sendto** for networking

Can specify a whitelist (seccomp filter) for syscalls

```
minijail0 -S whitelist_file /path/to/mydaemon
```

*mydaemon crashes if another syscall is executed*



# Pillars of Sandboxing III: Filtering system calls

## **complex\_whitelist\_file**

ioctl: arg1 == TCGETS || arg1 == TCSETS

mmap: arg2 in 0xffffffffb || arg2 in 0xffffffffd

mprotect: arg2 in 0xffffffffb || arg2 in 0xffffffffd

Can only pass TCGETS and TCSETS as second argument to ioctl

Memory can't be both writeable (PROT\_WRITE, bit 1) and executable (PROT\_EXEC, bit 2).

# Pillars of Sandboxing III: Filtering system calls

## Generating a policy file

- 1) `strace -f <cmd> 2>strace.log`
- 2) [`generate\_seccomp\_policy.py`](#) `strace.log > whitelist_file`

## Seccomp filtering caveats

- Syscalls are platform dependent! Need separate policy files.
- Did your strace hit all code paths? Might miss some syscalls.

# Pillars of Sandboxing IV: Namespacing

Process ID namespace

- Hides other processes

Mount namespace

- Hides parts of the file system

- Makes parts read-only

Other namespaces: IPC, cgroup, network, user, UTS

# Pillars of Sandboxing IV: Namespacing

Example: Process ID namespace

```
# minijail0 -p /bin/ps -A
```

PID	TTY	TIME	CMD
-----	-----	------	-----

1	?	00:00:00	minijail-init
---	---	----------	---------------

2	?	00:00:00	ps
---	---	----------	----



# Pillars of Sandboxing IV: Namespacing

Example: Mount namespace

```
# minijail0 -v -P /tmp/my_root_folder \  
-b /bin,/bin /bin/ls /
```

bin

-v	Enter mount namespace
-P	Enters a pivot root (“unmounts everything”)
-b /bin,/bin	Bind-mounts /bin only

# Actual Daemon Startup

```
minijail0 -i -l -r -t -n -c 180 -p -v -P /tmp/authpolicyd_chroot  
-b /,/ -b /dev,/dev -b /sys,/sys -b /run,/run -b /var,/var  
-b /run/authpolicyd,/run/authpolicyd,1  
-b /var/lib/authpolicyd,/var/lib/authpolicyd,1  
-b /var/lib/metrics,/var/lib/metrics,1  
-u authpolicyd -g authpolicyd -G  
/usr/sbin/authpolicyd
```

Custom **seccomp filters** applied to net, smbclient etc. directly

# Summary

Chromebooks can be joined to Active Directory domains

Easy for enterprises to try out Chromebooks

No Google user account necessary

Managed via GPO

Support for Android apps, certificates, file shares, printing

Sandboxed Samba

# Future Plans

Shadow account with full capabilities (Docs, Drive, Chrome Sync etc.)

Native SMB file shares

Simplified policy management

Streamlined domain join

Kiosk and Public Sessions

Reporting

Questions?

Thank you!!!