

Security Engineering

Modern operating systems security 1. Trusted code base. Use of MAC in SELinux and phones. Android, iOS and Windows security mechanisms, both in theory and practice.

Definitions

- Trusted Code Base: Components (hw/sw/human) whose correct function sufficient to ensure the security policy is enforced / whose failure can breach security policy.
- Reference monitors: Mediate access control & be small enough for complete analysis
- Safety integrity levels: a more dependable system must not rely on a less dependable one!

Discretionary Access Control

Discretionary Access Control: ACLs

- Access Control Lists: store permissions with file. May need different permissions for different programs, so actually a (user, file, program) triple
- ACLs scale badly without RBAC.
- Finding all the files a user has access to is a massive pain.

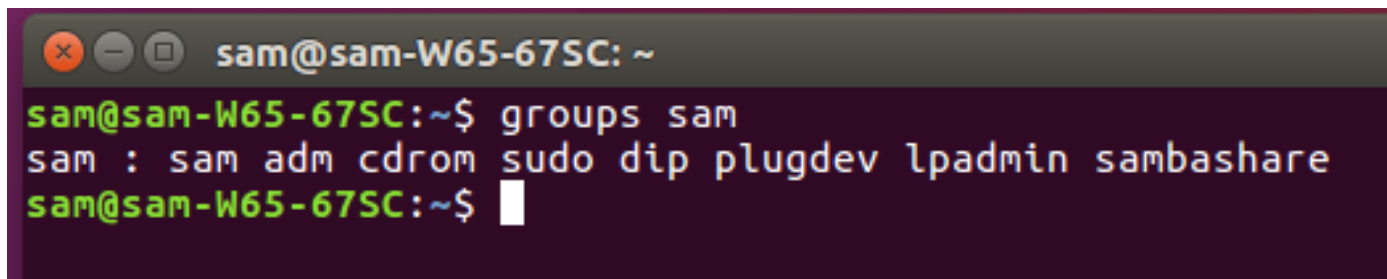
Unix Access Control Lists

- In Unix (and thus Linux, Android, iOS...): rwx attributes, with owner,group,world, per file. Can have richer Posix extended ACL extension.
- Sysadmin can do anything!

```
sam@sam-W65-67SC: ~/.ssh
sam@sam-W65-67SC:~$ cd .ssh
sam@sam-W65-67SC:~/.ssh$ ls -la
total 40
drwx-----  2 sam sam 4096 Mar 26 16:42 .
drwxr-xr-x  76 sam sam 4096 Aug 11 18:38 ..
-rw-----  1 sam sam 1671 May 14 2020 id_dsa
-rw-----  1 sam sam 1675 May 14 2020 id_rsa
-rw-r--r--  1 sam sam  398 May 14 2020 id_rsa.pub
-rw-----  1 sam sam 5546 Mar 31 15:55 known_hosts
```

Unix Access Control Lists

- User part of multiple groups.

A terminal window with a dark purple background and a grey title bar. The title bar contains window control icons and the text 'sam@sam-W65-67SC: ~'. The terminal shows the command 'groups sam' being executed, resulting in the output 'sam : sam adm cdrom sudo dip plugdev lpadmin sambashare'. The prompt 'sam@sam-W65-67SC:~\$' is visible on both lines.

```

sam@sam-W65-67SC: ~
sam@sam-W65-67SC:~$ groups sam
sam : sam adm cdrom sudo dip plugdev lpadmin sambashare
sam@sam-W65-67SC:~$
```

Unix Access Control Lists

- User part of multiple groups.

```
sam@sam-W65-67SC: ~  
sam@sam-W65-67SC:~$ cat /etc/group  
root:x:0:  
daemon:x:1:  
bin:x:2:  
sys:x:3:  
adm:x:4:syslog,sam  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mail:x:8:  
news:x:9:  
uucp:x:10:  
man:x:12:  
proxy:x:13:  
kmem:x:15:  
dialout:x:20:  
fax:x:21:  
voice:x:22:  
cdrom:x:24:sam  
floppy:x:25:  
tape:x:26:  
sudo:x:27:sam
```

Unix Access Control Lists

- Not directly a (user,program,file) triple: gets simplified down to setuid and setgid bits, and (user,file).
- The triple (user,program,file) is only implemented indirectly by setuid. Fiddly and error-prone. Leads to suid root...

Discretionary Access Control: Capabilities

- Capabilities: store per user, not per file.
- Finding all the users who have access to a file is a pain.
- Hard to revoke access to a particular file, or produce evidence of who could have broken said file.
- Easily transferred
- Public key certificates are really capabilities.

```
sam@sam-W65-67SC: ~/.ssh
sam@sam-W65-67SC:~$ cd .ssh
sam@sam-W65-67SC:~/.ssh$ ls -la
total 40
drwx----- 2 sam sam 4096 Mar 26 16:42 .
drwxr-xr-x 76 sam sam 4096 Aug 11 18:38 ..
-rw----- 1 sam sam 1671 May 14 2020 id_dsa
-rw----- 1 sam sam 1675 May 14 2020 id_rsa
-rw-r--r-- 1 sam sam 398 May 14 2020 id_rsa.pub
-rw----- 1 sam sam 5546 Mar 31 15:55 known_hosts
```

Mandatory Access Control

DAC vs MAC

- DAC: Start in supervisor mode, and as admin, can make less privileged accounts available for less trusted tasks.
- MAC: sysadmin no longer the boss: ultimate control rests with the security policy (possibly set by remote govt authority in defence setting).
- Alternate view (Android): DAC requires permission of the user. MAC requires consent of user, developer AND platform -- 3-party consent.

Mandatory Access Control (MAC)

- Occasionally synonymous with MLS - Multi-Level Security (Unclassified, Confidential, Secret, Top Secret)
- Enforced by system policy, not by user discretion!
- Traditionally for military systems, e.g. Bell LaPadula

Bell LaPadula

- Simple Rule (No Read Up): A subject at a given security level may not read an object at a higher security level.
- * Property: (No Write Down): A subject at a given security level may not write to any object at a lower security level.

Mandatory Access Control: BIBA

- BIBA: Uses the opposite duality of confidentiality and integrity and thus reverses BLP -- low water mark: integrity of an object is the lowest level of all the objects that contributed to its creation
- Used in Windows (partially): see later.

BIBA

- Simple Rule (No **Write** Up): A subject at a given security level may not write to an object at a higher security level.
- * Property: (No **Read** Down): A subject at a given security level may not read from any object at a lower security level.

MAC Today

- The military mostly gave up on BLP systems, because they were riddled with Covert Channels
- In military systems, tend to now use system-high: every system just at one level, with airgaps implemented by firewall. Multiple *Independent* Levels of Security (MILS).
- Linux/Windows: Mostly discretionary access control schemes. But, some ideas inspired by MAC: especially in SELinux, used in Android (see later).

Other Forms of MAC

- Bring-Your-Own-Device Management: Not just Samsung Knox: also Office 365 and permissions changes on Android. This gives 4-party consent (user, developer, platform, company).
- DRM also a form of mandatory access control: stopping a subscriber (Top Secret) sharing with a non-subscriber (Unclassified).
- Trusted Boot: See next time

Case Study 1: Android

Android Discretionary Access Control

- Based on Linux
- App Isolation: Treat Apps by different companies as different users, using SETUID.
- Permissions also effectively capabilities, implemented by adding GIDs to the list of groups of the SETUID. “Permissions manifests” basically compile down to this.
- Early versions: all granted at install time. So flashlight apps started demanding your address book at install time so they could sell it.
- Since Android 6, Google moved to Apple model of TOFU, but earlier apps still demand on installation.

Android Mandatory Access Control: SELinux

- Consent of user, developer AND platform: 3-party consent.
- Protect core system functions, even from some parts of the kernel.
- Can't solve all kernel attacks but provides some isolation.

Example: GingerBreak / GingerMaster

- Bug in *vold*, the external storage (SD card) manager
- Since vold runs as root, this allowed running a “root shell” and thus “rooting” of the whole device.
- SELinux could have blocked in several places, despite vold being root:
 1. Apps blocked from reading process ID of vold
 2. Apps blocked from sending messages to vold
 3. Vold blocked from executing non-system binaries
 4. Root shell still only allowed same *security id* privileges as vold itself

Android Mandatory Access Control: SELinux

- Also provides some stronger defences than discretion in userland
- Assumed that users will be tricked into installing malicious apps

CVE-ID	
CVE-2011-1717	Learn more at National Vulnerability Database (NVD) <ul style="list-style-type: none">• CVSS Severity Rating• Fix Information• Vulnerable Software Versions• SCAP Mappings• CPE Information
Description	
Skype for Android stores sensitive user data without encryption in sqlite3 databases that have weak permissions, which allows local applications to read user IDs, contacts, phone numbers, date of birth, instant message logs, and other private information.	

SELinux

- Implemented since Linux 2.6.
- Builds RBAC on top of Type Enforcement.
- Users *-map>* Roles at login, Roles *-auth>* Domains, Domains *-permission>* types.
- Can handle integrity as well as confidentiality: allows roles to be revised when programs invoked, e.g. can lose system-writing privilege when running internet-downloaded software.
- Implements a general constraints engine that can express RBAC, TE and MLS.
- E.g. can separate your DNS server from your web server.

Android Permissions: Issues

- API has poor documentation, and the permissions system is often the enemy of the developer, who ends up requesting more permission than they really need.
- Android still has malware! e.g. Pegasus via zero day, but costs \$1 million. Alternative markets out of Google's control
- And lots of unpatched devices. The OS-update ecosystem is a disaster...
- Getting access control right intersects with lots of awkward edge cases, e.g. factory reset

Case Study 2: iOS

iOS

- Also a Unix derivative, via FreeBSD and Mach kernel.
- Domain and Type Enforcement for tamper-proof system components. App permissions are capabilities, granted on first use on consent.
- Signed ecosystem from the market, just as Android has its default supported Google Play. Allows screening and also revenue taking.
- On the App Store, Apple signs the binaries. On Google Play, the developer does.

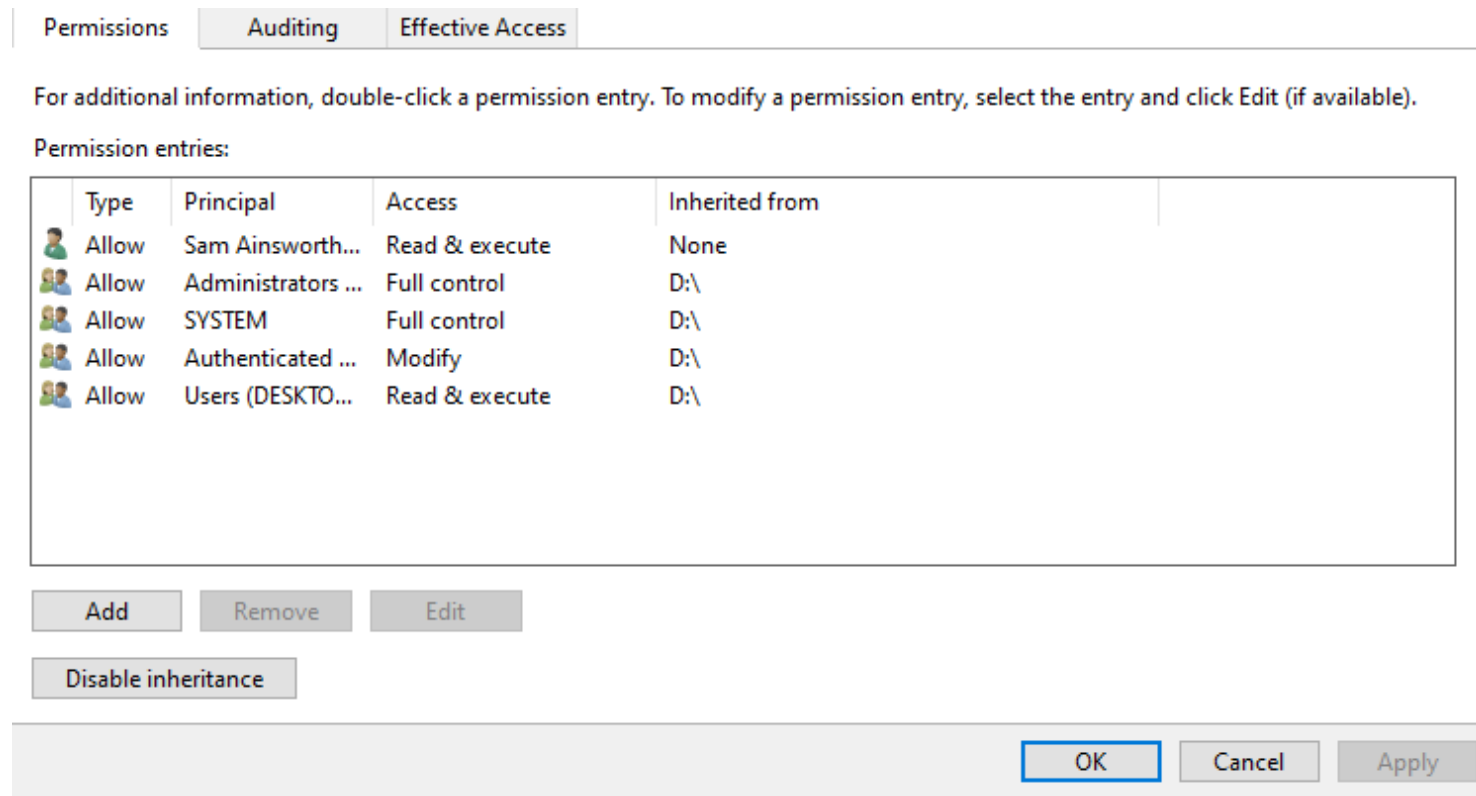
iOS

- Biometrics stored via encryption by the secure enclave (SE). Neither iOS nor TrustZone are trusted with this data!
- Passcode 10 tries: file keys derived only then.
- Vertically integrated, closed ecosystem

Case Study 3: Windows

Windows Access Control Lists

- Very complex Access Control, from Windows NT onwards. RWX, but also Take Ownership, Change Permissions and Delete.



The screenshot shows the 'Effective Access' tab of a Windows dialog box. It contains a table of permission entries and several control buttons.

Permissions Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

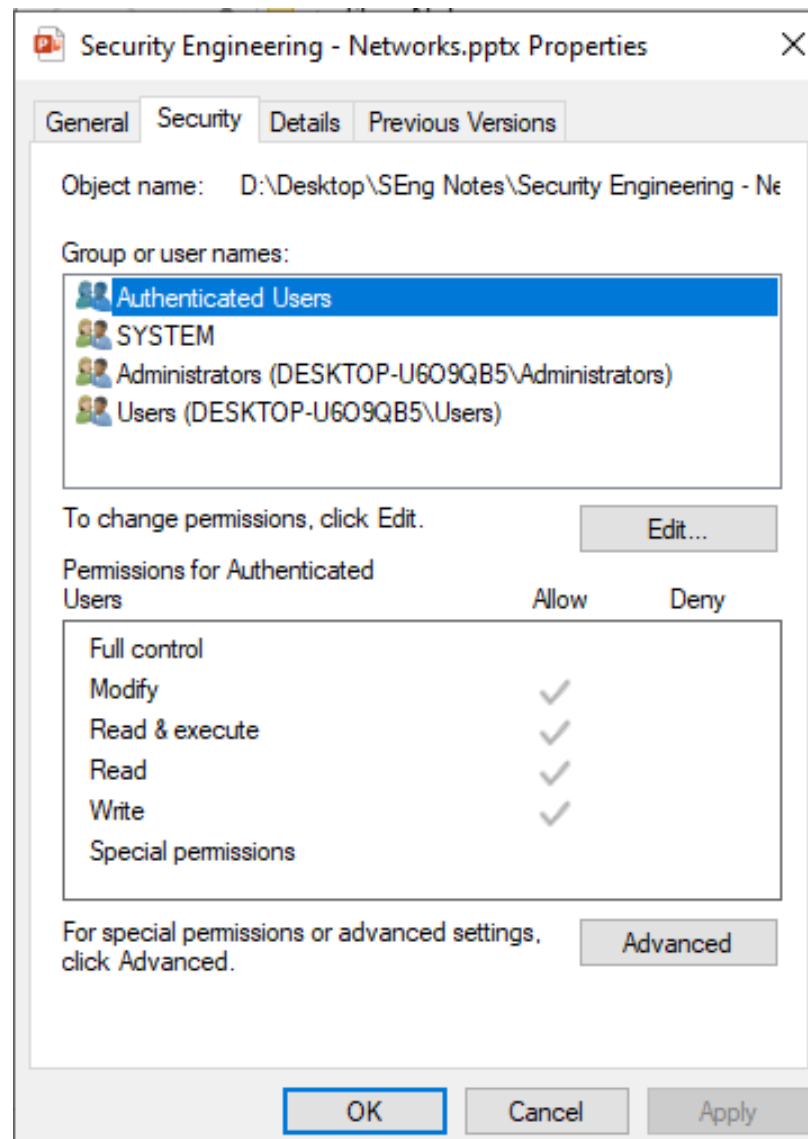
Type	Principal	Access	Inherited from
Allow	Sam Ainsworth...	Read & execute	None
Allow	Administrators ...	Full control	D:\
Allow	SYSTEM	Full control	D:\
Allow	Authenticated ...	Modify	D:\
Allow	Users (DESKTO...	Read & execute	D:\

Add Remove Edit

Disable inheritance

OK Cancel Apply

Windows Access Control Lists



Windows ACLs

- Very complex Access Control, from Windows NT onwards. RWX, but also Take Ownership, Change Permissions and Delete.
- AccessDenied, AccessAllowed, SystemAudit: AD overrides AA if set multiple times.

Permission Entry for OS Security 1.pptx

Principal: Sam Ainsworth [Select a principal](#)

Type: Allow

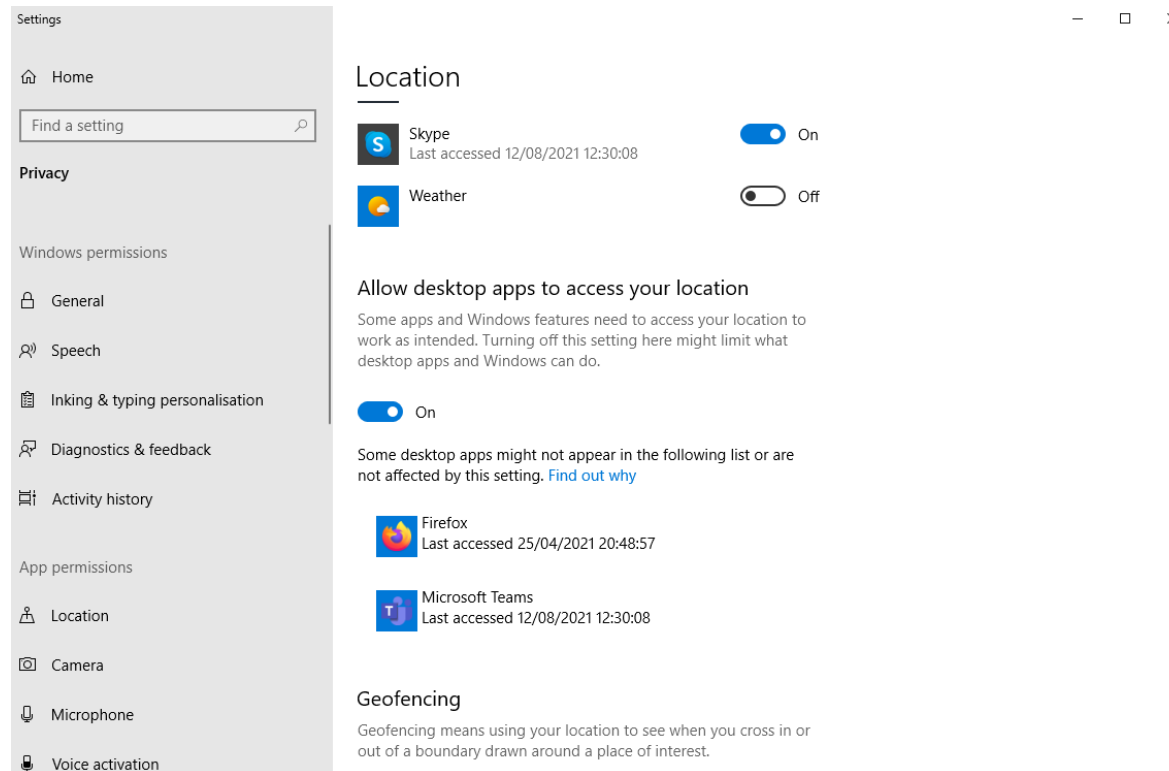
Advanced permissions: [Show basic permissions](#)

<input type="checkbox"/> Full control	<input type="checkbox"/> Write attributes
<input checked="" type="checkbox"/> Traverse folder / execute file	<input type="checkbox"/> Write extended attributes
<input checked="" type="checkbox"/> List folder / read data	<input type="checkbox"/> Delete
<input checked="" type="checkbox"/> Read attributes	<input checked="" type="checkbox"/> Read permissions
<input checked="" type="checkbox"/> Read extended attributes	<input type="checkbox"/> Change permissions
<input type="checkbox"/> Create files / write data	<input type="checkbox"/> Take ownership
<input type="checkbox"/> Create folders / append data	

Clear all

Windows App Permissions

- Not really integrated as a (user,program,file) triple – or even setuid.
- ACL just does (user,file), and a separate system handles a limited set of permissions, mostly for “apps”



Windows Vista (2007)

- Remove most drivers from the kernel.
- UAC replaced default admin privilege with user-mode default. In XP (2001) many routine tasks needed admin privilege.
- Application Information Service to launch applications that require elevated privilege, virtualizes them to give them an imaginary registry to alter.
- Elevation prompts for admin privilege. ("ambient authority" frowned upon where possible to avoid -- access should be temporary -- sudo not root)
- Mandatory Integrity Control (see later)

Windows 8 (2012)

- Dynamic access control to give contextual control: work vs home pc vs phone, in Active Directory/Kerberos.
- 8.1: Security Identifiers (SIDs) given on login. Encouraged to sign in via Microsoft account, authenticated remotely, and where credentials stored locally, protected by virtualisation.
- Secure Boot to verify the boot sequence and software all matches that from the OEM
- Pin Login

Windows 10 (2016)

- Windows XP: Ctrl-Alt-Del for login -- gone with Windows 10 because nobody understood it.
- Multi-factor authentication support e.g. FIDO
- Device Encryption AKA Bitlocker Device Encryption (not Bitlocker, that's different) – encrypt files and recover the key via Microsoft account.

Windows: Mandatory Integrity Control: BIBA?

- Adds an integrity level (Low, Medium, High, System)
- Standard users Medium, elevated users High, browsers Low
- When a file executed, object starts with the minimum integrity level of (User, File)
- Files downloaded from the internet are therefore low, assuming the browser is.

Windows: Mandatory Integrity Control: not BIBA?

- NoWriteUp (simple rule) but no NoReadDown (* property).
- Things downloaded in IE can read most files, but not write to them, to limit malware damage.
- Not really “mandatory”: have user confirmation instead to upgrade downloaded content.
- Contrast with Android, which isolates each app to its own domain.

Why is Windows so complicated?

- Corporate customers need complicated access controls. MS made half its revenue from firms >25000 seats.
- Decades of backwards compatibility means testing at scale. And introducing features slowly, and complex compatibility layers e.g. Application Information Service

Further Reading

- Security Engineering Chapters 6, 9, 22, 27.
- Google SRS Chapter 5: Design for Least Privilege
- <http://www.cs.columbia.edu/~lierranli/coms6998-7Spring2014/papers/SEAndroid-NDSS2013.pdf>
- <https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-device-encryption-overview-windows-10>
- <https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-overview>