Storage Systems

NPTEL Course
Jan 2012
(Lecture 29)

K. Gopinath
Indian Institute of Science

Why Secure Storage?

If storage has to be a bedrock, have to ensure that it is

- Highly available
  - Resilient to failures
  - Resilient to DoS/DDoS Attacks
- Protected from intruders
  - Prevent malicious tampering
  - Controlled access; avoid leakage of information
  - Prevent replay of stale information
- To the extent feasible, use formally verified ("correct") storage protocols
  - No trapdoors either from a systems, protocol or cryptographic perspective: NFS root; X11 auth, ...

May need tamper-proof archival storage (legal&c reasons)

Security in Storage

- Security at FS, block, device levels
  - Also at std network security issues if storage is networked
- Standard security issues
  - Integrity
  - Secrecy
  - Availability (DoS attacks)
- New security issues:
  - Flash wear (DoS)
- Viruses often spread thru storage devices (floppy, USB, ...)
- Security for Metadata (small amounts) vs Data (large amounts)
  - Public Key encryption OK for metadata but not for data
  - Stream ciphers with symmetric encryption for data
- Aggregation attacks
  - When lots of data, new patterns or secrets can be deduced

Systems security

- Systems with basic access control since timesharing systems began ('60)
    - Multics, (Unix) *rwxrwxrwx*! at file level
    - MAC vs DAC
        - SELinux model
- Cryptography used widely but...
    - ``If you think cryptography is the solution to your problem, you don't know what your problem is,'' Roger Needham
    - Key mgmt critical
- Complex world-wide information systems, netw/storage subsystems, etc require much more sophisticated models
    - anonymous users/services, delegation, trust mgmt, scalability
    - need to have an integrated model of all authentication/ authorization models: rwx, setuid, PAM, SELinux, cryptofs, X11 auth, NFS, ssh, httpd, IPSec, firewalls, iSCSI, ...
    - highly available access control: eg: clusters, SANs
- Info Flow Models
    - Need proof that info flow respects some security policies

CD/DVD/Blu Ray

- CD: no protection or ad hoc
- DVD: CSS (content scrambling system)
  - Every DVD player equipped with a small set of player keys (per DVD player manufacturer)
  - Every disk has a disk key data block organized as:
    - 5 bytes hash of decrypted disk key (H)
    - disk key encrypted with player key 1 (dk1), player key 2 (dk2)... player key 409 (dk409)
  - When presented with a new disc, a player will attempt to decrypt contents with set of keys it possesses
    - Suppose a player has a valid key for slot 100, it will calculate
      - Kd? = decrypt(dk100, Kp100)
    - To verify that Kd is correct, check following; otherwise, next player key
      - H == hash(Kd?)
  - Problem! By trying all 2^40 possible Kd, disk key can be deduced without knowing any valid player key.
  - To decrypt contents, an additional key tk (title key) decrypted with valid Kd (Kt)
  - Each sector of data files optionally encrypted by a key derived from Kt by XOR of specified bytes from the unencrypted first 128 bytes of the 2048 bytes sector
    - Uses a stream cipher (LFSR).
  - However, due to flaws, 2^40 checks reduced to 2^16=> 450MHz Pentium needs <1 min

AACS (Advanced Access Control System)

- Blu-Ray
- Fixed some of the problems of CCS but broken here also due to another attack
  - Inspite of many layers of encryption, keys needed to obtain unencrypted content stream that is available somewhere in memory for playback
  - Write a simple device driver to scan kernel memory for keys and check!
- Called “Trusted client” problem
- Need “trusted computing platform” that only lets validated sw to run (not, the dd above!)
- But PC is not such a platform
- With “Trusted Boot” PC. May be possible.
- However: “Against the average user, anything works. Against the skilled attacker, nothing works.” B. Schneier

Access Control Models

DAC model

- each subject decides how its objects interact with others
- security mgr keeps access control matrix
- checking safety problem: HRU undecidable
  - However, many decidable models exist: eg: Take grant model
    - Num of subjects and objects fixed but can have some dynamicity such as conditional auth (based on state)

MAC model

- security server decides how any object interacts

RBAC model

- introduces roles

ABAC (attr based access control)

- rights based on attributes

SELinux

- concepts and capabilities
  - mandatory access controls
  - mandatory integrity controls
  - role-based access control (RBAC)
  - type enforcement architecture
- For every every current user or process, SELinux assigns a 3 string context (role, user name, domain)
  - domain and type equiv
- Policy rules give explicit perms: eg. which domains user must possess to perform certain actions with given target (R/X/W)
- A policy consists of a mapping (labeling) file, a rule file, and an interface file that defines the domain transition
  - Domain transition on fork, execv with setuid programs
- Can confine a deamon to safe actions
- Very detailed; hence easy to get wrong
  - First try in permissive mode and tighten it but may make it too restrictive/break
- Not possible or difficult across different systems

Conclusions

- Security is a tough problem
- Many attack vectors; each requires careful analysis and mitigation
  - Nowadays, good crypto techniques widely avlbl
  - Hence, attackers do not try to break crypto!
- System wide analysis needed