Building Chinese Walls in BSD UNIX[1]

Simon Foley,
Department of Computer Science,
University College, Cork, Ireland.
s.foley@cs.ucc.ie

I. Introduction

A Chinese Wall security policy [1] can be given as $(ORG, \_ \wr \_)$, where ORG is a set of organizations, and given $a, b \in ORG$, then $a \wr b$ means that a is in competition with b. A system enforces a Chinese Wall policy if it ensures that it is not possible for a consultant (user) to access data about $a, b \in ORG$, with $a \wr b$ (conflict of interest). We outline how such a requirement can be supported by standard UNIX. The approach is not appropriate for security critical applications: it serves to illustrate that it can be done in a rather simple way, and may be useful for less critical applications.

II. SUID Building Blocks

In our initial scheme every organization a has a unique phantom user-id $u(a)$, and a unique group-id $g(a)$ (functions u and g are 1:1). A phantom user-id has no associated user (for example, `news` and `ingres`). A file (dataset) f containing organization a data has owner $u(a)$, effective group $g(a)$, and protection bits `RW` set for group access only. If $mbrs(grp)$ gives the users who are members of group grp, then $mbrs(g(a))$ represents the users (consultants) who may access files of organization a. Thus, `/etc/group` implements $mbrs$, and we assume that `/etc/conflict` implements $\_ \wr \_$.

Initially, $mbrs(g(a))$ is empty for every organization a. A consultant c invokes `consult(a)`, if she wishes to access organization a files. Program `consult` is owned by root and is SUID. This operation succeeds if

$$\forall b : ORG \bullet c \in mbrs(g(b)) \Rightarrow \neg a \wr b$$

In this case, $mbrs(g(a))$ is updated to include c. Assuming that organization groups may grow only according to operation `consult`, it follows that it is not possible for a consultant to gain *access* to files owned by competing organizations.

III. Controlling Insider Trading

A problem with this scheme is that insider trading is still possible. The mechanism works at the granularity of access-rights: a Trojan Horse embedded in an application can copy data around the Wall to a conspirator. This attack can be limited if security is also enforced within the application; this should be certified according to the Clark-Wilson criteria [2].

Organization files are Constrained Data Items (CDIs), and application operations are trusted Transform Procedures (TPs). Consultants may access CDIs only via TPs. If G represents the set of organizations that consultant c can currently access, then for every TP tp, the system must support access triples of the form (c, tp, G'), where $G' \subseteq G$. Initially, $G = \{\}$, and this 'grows' as the consultant is granted the right to access organization CDIs.

We adapt the strategy described in [3] to implement these access triples in UNIX. Initially, assume there is just one consultant c, and introduce a further phantom user-id $tu(c)$ and group-id $tg(c)$. Each tp is owned by $tu(c)$, has effective group-id $tg(c)$, is SUID, and has protection bits `RX` set for group access, with $mbrs(tg(c)) = \{c\}$. Initially, $mbrs(g(a))$ is empty for every organization a (CDI). Request `consult(a)` operates as before, except that, instead of user-id c, it adds the TP user-id $tu(c)$ to $mbrs(g(a))$. Thus, consultant c may access the CDI only by invoking a TP, which runs as $tu(c)$. This scheme can be generalized to many users by either having unique copies of TPs for each consultant, or using TP wrappers as proposed in [3].

As with [3], this scheme is effective only if proper care is taken in developing the SUID programs, all organization datasets are modeled as CDIs, and every TP is certified to enforce the Chinese Wall policy.

IV. Dynamic Segregation of Duties

Dynamic segregation of duties may be viewed as an integrity dual of aggregation policies (such as Chinese Walls). For example, a clerk may *either* generate a purchase order *or* process an invoice, but not both. Representing this as a conflict of interest ($\texttt{inv} \wr \texttt{po}$) allows it to be supported by our mechanism. Given initial access-triple $(\texttt{clerk}, tp, \{\})$, the clerk may access invoices (`consult(inv)`), or purchase orders (`consult(po)`), but not both.

V. References

[1] D.Brewer and M.Nash. The Chinese Wall Security Policy. In *Proceedings 1989 IEEE Symposium on Security and Privacy.*

[2] D.D.Clark and D.R.Wilson. A Comparison of Commercial and Military Computer Security Models. In *Proceedings 1987 IEEE Symposium on Security and Privacy.*

[3] W.Polk. Approximating Clark-Wilson Access Triples With Basic UNIX Controls. In *Unix Security Symposium IV.* 1993.

[1] Registered Trademark.