MULTILEVEL ARCHITECTURES FOR ELECTRONIC DOCUMENT RETRIEVAL[1]

James A. Rome (jar@ornl.gov) and Johnny S. Tolliver (jxt@ornl.gov)
Oak Ridge National Laboratory; P.O. Box 2008; Oak Ridge, TN 37831

Abstract

Traditionally, most classified computer systems run at the highest level of any of the data on the system, and all users must be cleared to this security level. This architecture precludes the use of low-level clearance personnel for such tasks as data entry, and makes sharing data with other entities difficult. This paper presents three architectures for an MLS electronic document retrieval system using COTS products. The CMW version has been implemented but not deployed. Although we believe that the resulting systems represent a real advance in usability, scaleability, and scope, the disconnect between existing security rules and regulations and the rapidly-changing state of technology will make accreditation of such systems a challenge.

Introduction

There exists a need for the electronic multilevel storage and retrieval of classified documents. In most existing systems, all documents are stored on a system that runs at syshi (the highest classification level of any document in the system), and all users must also be cleared to this high level. The U.S. government is under great pressure to reduce costs by reducing the number of high-level security clearances. As a result, there is a recognized need to move to a multilevel secure (MLS) computer architecture that would allow users with different security clearances to access the same system, but that would also protect all data against unauthorized disclosure.

One way to achieve MLS systems is to use compartmented mode workstations (CMW) to enforce security. Such systems have been evaluated at a *B1* level of trust in the "Orange Book"[1], but also support parts of higher levels of trust. Where they are deficient is in the areas of covert channels and code assurance and testing. Moreover, according to the regulations, only two adjacent classifications and two compartments are allowed on a multilevel system; such a requirement makes it impossible to actualize a realistic security environment where many more need-to-know categories are required, and where four classification levels are desirable (Unclassified, Confidential, Secret, Top Secret).

The government is following the lead of other industries in the decision to use commercial off-the-shelf (COTS) software products where ever possible. The use of COTS software greatly reduces the cost of long-term maintenance associated with custom software products, and shifts the burden of user support to the software manufacturer. In most cases, this results in a better, more up-to-date product, greater user acceptance, and lower costs. However, the use of COTS in a MLS system poses numerous challenges.

The entire program may be a "black box," which probably needs to raise several system privileges in order to run. In such cases, these privileges need to remain in effect for the entire time that the program is running. How can we decide whether or not the program abuses this trust? License demons (processes running to check on the number of users running a program) are especially troublesome in this respect because they might require read/write access to license files at any security level. Allowing this action requires an override of mandatory access control (MAC), the primary mechanism that exists to enforce security policies. Usually such programs (e.g., word processors) can only be run at a single level without greatly compromising security.

Other programs come as a combination of proprietary code in black boxes, together with developer's hooks, and perhaps source code to allow customization of the high-level interfaces. In these cases, it may be possible to turn the program into a *trusted program*. This process invokes the security mechanisms of the trusted platform to raise privileges only when necessary (privilege bracketing). However, the granularity of the hooks into the black box code might not be fine enough to assure that privileges are not raised for some operations that take place within the black box code. In addition, such code modifications must be maintained whenever the system or the COTS product are updated, thus obviating some of the advantages of using COTS.

Finally, the regulations and procedures for evaluating the security of COTS or modified COTS are murky, and outdated [2].

---

[1]This work was supported by the U. S. Department of Energy Office of Safeguards and Security under DOE Project No. BR GD605030 with Lockheed Martin Energy Systems, Inc.

Functional Description

The Department of Energy Office of Safeguards and Security (OSS) Information Management Center (IMC) currently has about 700 linear feet of documents in storage, with an additional 25% in long-term storage. These documents range from Unclassified to Top Secret, and are subject to handling caveats and need-to-know restrictions on access. The Department is legally required to preserve these documents for varying periods of time.

There is a great need to be able to search these documents in an efficient and timely manner which implies that they need to be converted into an electronic format. Legally, this format must preserve the "look" of the original document. In addition to searches over the document text, it is also necessary to search over keywords such as author, subject, date, security label, etc.

Initially the Electronic Document Center (EDC) will be connected to a classified LAN on which all users have appropriate clearances, but not necessarily all need-to-know categories. Later, users cleared to a lower level will be able to access the LAN. Eventually, it would also be desirable to allow access to the unclassified portions of this database from the unclassified LAN.

Therefore, the challenge is to provide a secure but user-friendly method of accessing, searching, and retrieving documents from the EDC. For implementing these requirements, the architecture chosen employs the ubiquitous and familiar World Wide Web technology in an "intranet" environment coupled with a CMW to enforce the multilevel security. COTS software is used whenever possible to reduce the amount of custom programming required.

DOE LAN Environment

The environment for the EDC is rather special, and requires explanation. In this incarnation, the EDC will be attached to a classified LAN on which all users are cleared to the highest clearance (Top Secret), but may not have "need-to-know" requirements for all categories (or compartments in the CMW parlance). At a later stage, not all users will be cleared to Top Secret. In addition, the users are all connected to the LAN by PCs running Windows 3.1. The PCs are accredited to run at syshi (Top Secret plus all categories), but because Windows has no knowledge of the labeled packets used by CMW systems, the PCs are all regarded as "untrusted" by the CMW and come into the CMW with a single, fixed security label.

Because users share offices and PCs with removable hard disks, and because the PCs are assigned dynamic *IP* addresses, it would be infeasible to assign a different label to each PC. Therefore, it makes sense to label all PCs either syshi or syslo. Because all the PCs are in fact at syshi, one could argue that they should all be assigned to syshi. However, in that case, if users were able to access a shell on the CMW through some security flaw, they would dominate any mandatory access control (MAC) label and have free run of the system. Accordingly, we assign all PCs a syslo label recognizing that it is rather arbitrary.

Thus, our system is truly multilevel, but its security assurances need only protect against access to unallowed compartments. However, in the future it is felt that such systems can be made strong enough so that they could provide access to users not cleared to the highest level. Future enhancements to these systems might include Fortezza cards which would provide strong encryption and authentication [3].

By fielding this system now, all of the data will be properly labeled so that it can be the basis of a later (improved) system. We have been careful to create a design that does not preclude the possibility of interacting with users on multilevel systems, or even eventually connecting the system (carefully!) to an Unclassified LAN. The architecture for such a system will be discussed later in this paper.

Single CMW Architecture

We decided to use a CMW platform because it is the only multilevel accredited platform for which a large variety of COTS products are available. The major components of the system were determined by comparing the various storage media, formats and COTS products that were available. The major components and design decisions of the system are:

- Documents are converted to Adobe Portable Document Format (PDF) to satisfy the legal requirement of "looking like the original," and to enable full text searches.
- Documents are stored on a read/write magneto-optical disk jukebox using a security-labeled file system.
- Indexes are stored on hard disks.
- The Excalibur Technologies RetrievalWare (RW) text retrieval database is used for search and retrieval.

- The Apache Web server would interface with the Web front end of RW.
- Adobe Capture is used to convert scanned TIFF images to PDF files. Capture's optical character recognition capabilities (OCR) provide searchable text. Because it is very expensive to review the accuracy of the OCR, the PDF files may also con - tain the original TIFF images in order to ensure that users can check any uncertain text.

The challenge is to determine how to utilize these COTS solutions in a CMW venue to produce a solution that satisfies user needs while enforcing security. The overall architecture of the system is shown in Fig. 1.

The heart of The EDC is a Hewlett-Packard (HP) J-200 workstation running HP-UX 10.16, a SecureWare-developed version of the CMW operating system (currently in evaluation).

Security issues

Client level

Although two CMWs can exchange labeled packets across an Ethernet connection, the clients in our case are all PCs running Windows 3.1 and the Netscape Web browser. They can only communicate using unla - beled packets, and hence must be enrolled into the CMW's network security database (*M6RHDB* file) at a single, fixed security level. But which level should be used? It only makes sense to choose either the highest (syshi) or lowest (syslo) security level available in the system. Because running at syshi would bypass most of

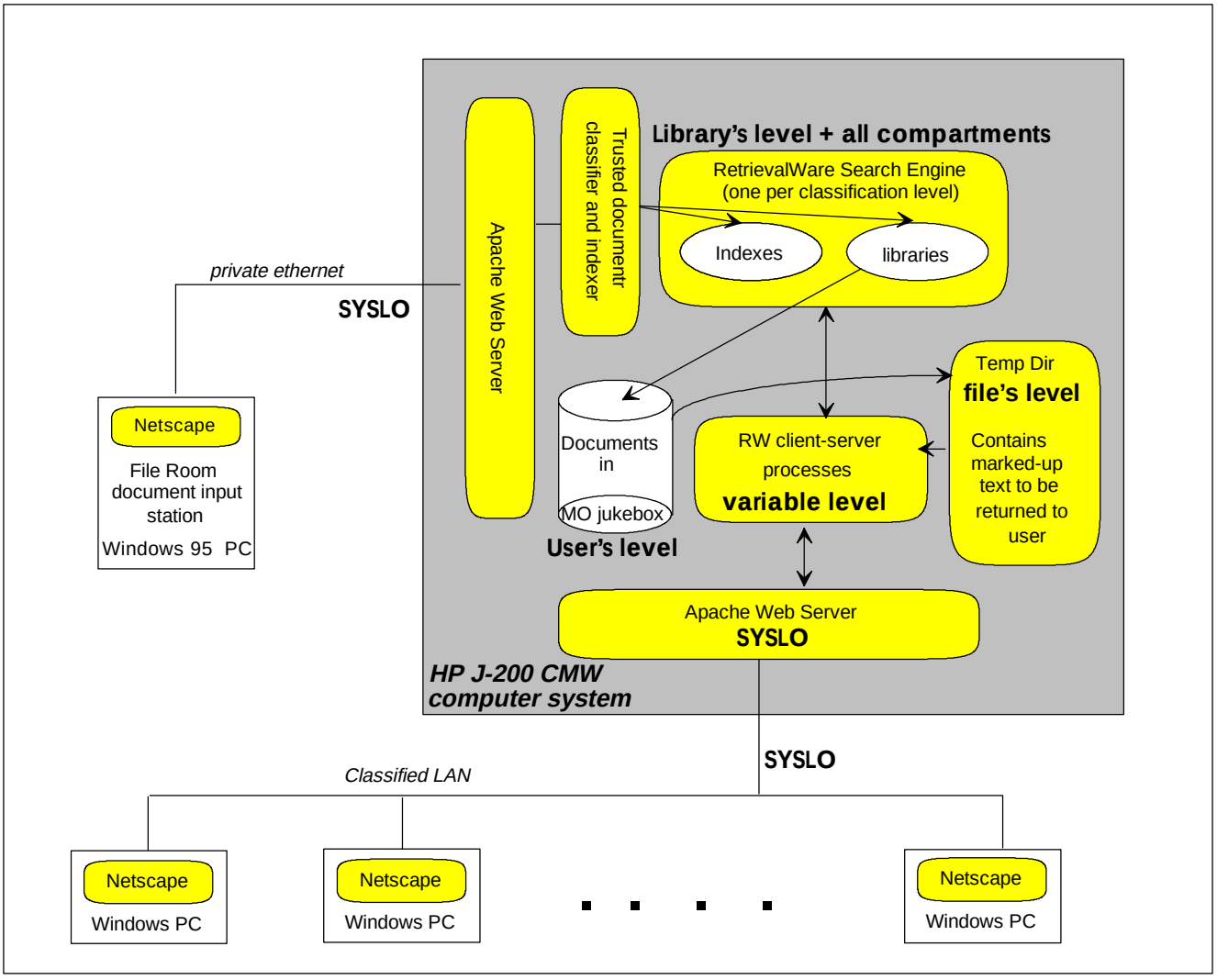

Fig. 1. Overall architecture of the CMW-based Electronic Document Center

the MAC security features of the system, the PCs are all assumed to run at syslo *even though this is not the case*.

Port access

It is possible to raise and lower the security levels of the network ports of the CMW in accord with the clearance of the user, or to override the MAC of the port. We felt that it provided more resistance to attack from the network to keep the port MAC intact. Because of the stateless nature of the Web interactions, the multiple Web server daemons that can be launched, and the complicated client-server database engine, it proved difficult to ensure that each Web request was served at the correct (user's) level so we decided to leave the ports at syslo at all times. To avoid the server problems, the user is reauthenticated for each Web request, and a new server is launched by the *inet* daemon to service the request.

Setting all the ports and Windows clients to syslo solves the above problems but raises another. All documents are stored in labeled files with their correct security label (classification + compartments) , and if they are to be served to the users at syslo, somewhere in the system, they will have to be "declassified" in order to send them through the syslo port. To perform this function, we utilized the fact that the CMW's trusted computing base (TCB) only checks MAC privileges when a file is opened. Therefore, we can obtain a file handle at one level and use it at another level. The file itself is not declassified. As is indicated in Fig. 1, the RetrievalWare client-server processes float their security levels up and down to be able to access the documents at their correct level and then to deliver them to the Apache server at syslo. Because the transaction only contains information dominated by the user's clearance, this level change does not cause a security risk provided that we make sure that the correct information is sent to the correct user. This bookkeeping is performed by the RW engine using a session key for each user..

Web server

The classified data are organized into four separate libraries, one for each classification level. When the user logs into Apache, we use his user id (*uid*) and *IP* address to identify the user and to determine his security level. The Apache server's authentication routines were modified to use the CMW's TCB for authentication (rather than the *.htpasswd* file), and also to reauthenticate the user for each Web request. Secure socket layer (SSL) encryption could be used to prevent sniffing attacks, but the stronger encryption provided by the next version of the Fortezza Card is preferable.

RetrievalWare security flow

Knowing the user's clearance, we present an html page that only allows the user to select from those libraries dominated by his clearance. This could be overridden by a clever user, so we check the user's selections against his clearance in the RetrievalWare CGI driving routine. Even if this step is bypassed, the user will be unable to retrieve any document from a library not dominated by his clearance because MAC will prevent it when the document is opened with the user's clearance. The RW Web front end keeps track of the user, using the *uid*, *IP* address and the time of day to form a hashed session key. Because the *uid* is used to identify the user to the CMW, and hence determine his privileges, it is vital that at every stage of the query process, we keep track of the *uid*. Unfortunately, the early versions of RW did not transmit the *uid* to the back-end client/server processes. To overcome this deficiency, we tack the *uid* onto the end of the query string, and then recover it in the back-end query engine.

Because there are only four separate libraries, they must contain documents with all of the different need-to-know categories. Therefore, the indexes for each library must have a security tag that is the classification level plus all of the compartments available at that level. To search the mounted indexes, the RW query process must be raised to the user's classification level plus all of the compartments. Note that this requirement implies that although compartments can have a lower bound, they cannot have an upper bound. If they did, a label could not be formed that would dominate the compartment at a classification higher than the compartment's highest level.

The user is not allowed to see all of the documents in the mounted libraries unless he is a member of all the compartments. To prevent the user from seeing hits on any documents in unallowed compartments, the trusted code performs a Boolean hidden query over the compartments of the documents (stored as a searchable field). If the compartments of the document are not all contained in the user's compartment list, no hit is returned. Thus, the RW search engine is relied upon to enforce the inadvertent disclosure of the title of a document in an unallowed compartment.

However, MAC is used to ensure that the contents of a document in an unallowed compartment are not disclosed. When the user selects a document to be returned, the retrieval engine's security level is set equal to the user's level and the document is opened. This process fails unless the user's clearance dominates the document's clearance. Once the document handle is

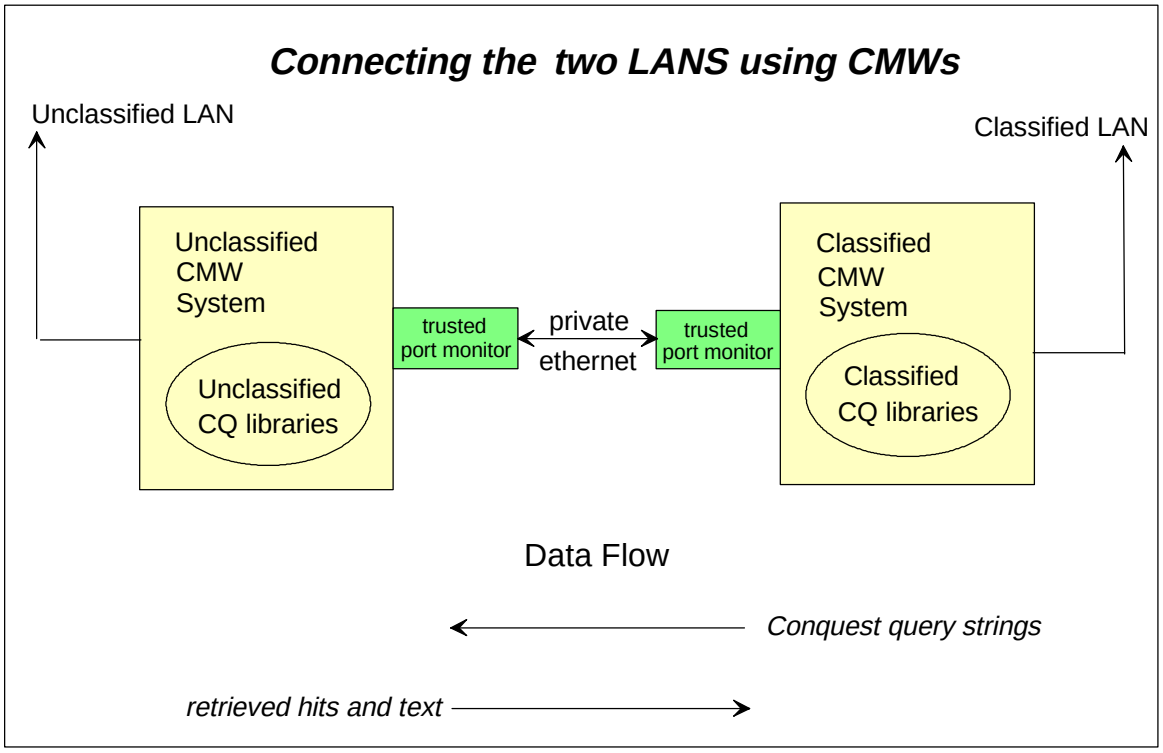

Fig. 2. Connecting the Classified and Unclassified LANs using an information monitor.

obtained, the process level is lowered to syslo, and the information can be sent out the syslo port to the user on the Classified LAN.

Connecting to an Unclassified LAN

Figure 2 shows how the scheme of Fig. 1 can be repli - cated to allow users on the Classified (C) LAN to query over documents contained on the Unclassified (U) LAN. Because RW supports remote libraries, the Unclassified Library can be moved to the U LAN. The only connection between the two LANs is via a private Ethernet link. The two trusted port monitors make sure that only RW queries can flow from the C LAN to the U LAN, and that only hit lists and retrieved documents can flow in the opposite direction. If the port monitors do their job properly (the main task for this future pro - ject), the only chance for release of information to the U LAN is if a user on the C LAN makes a "classified query." The bandwidth for this is quite low (the query string is limited in length) , and in any event, the RW engine on the U LAN will dispose of the query string when the query is completed.

Multiple Computer Architecture

To make the scheme of Fig. 1 work, both the Web server and the RW processes must be converted into trusted processes. The Web server authentica tion rou- tine is localized and can be swapped in easily if the server is upgraded. However, more extensive modifica - tions have to be performed on the RW code. Three of the RW servers undergo significant code changes. Many of these changes broke when the version of RW was upgraded from 5.0 to 5.2. These are just the prob - lems that the use of COTS was supposed to avoid. Accordingly, we have devised a scheme that requires much less extensive code modification, but at the expense of using multiple computers. The system, shown in Fig. 3, is fronted by a CMW-based system for several reasons. In the first place, the CMW offers enhanced resistance to attack and functions as a very secure firewall. The outside world has access to the web server on one Ethernet port , and the inside (the database computers) is attached to a second Ethernet port via a smart hub (not shown). Second, the software in the CMW system is modified to allow enforcement of the user's clearance level and his need-to-know cate - gories. The user's classification level is used to deter - mine which database servers are mounted to fulfill his search request. This adjudication is conceptually shown by the gray arc in the data flow diagram. For example, if the user is cleared to SECRET, the TOP SECRET server is not mounted.

Need-to-know categories are enforced by the database system as in the previous scheme. The code on the CMW system is modified to create a hidden query over the category keyword field that goes along with each of the user's queries. The hidden query prevents a hit on any document that is labeled with categories not possessed by the user.

In this architecture, all back-end systems are single-level hosts running at syslo. The database and Web server running on the CMW also run at syslo. Therefore, as compared to the pure CMW architecture, the security level of all processes running on the front-end machine are never changed (*principle of tranquillity*). MAC of classification is provided by mounting the correct databases, and MAC over categories is provided by the database engine via the hidden query. However, the extra step of assuming the user's security level to actually open the document is skipped. The RW code running on the back-end servers is unmodified COTS. The only RW modification needed on the CMW is the code to mount the correct databases and to perform the hidden query. Both of these changes are made in the RW Web front-end routine using standard developer's hooks which are not subject to upgrade changes.

If the external network has multilevel users, strong encryption can be used to protect the returned documents from the CMW to the user's PC. The encryption would probably be done by a future version of the Fortezza card.

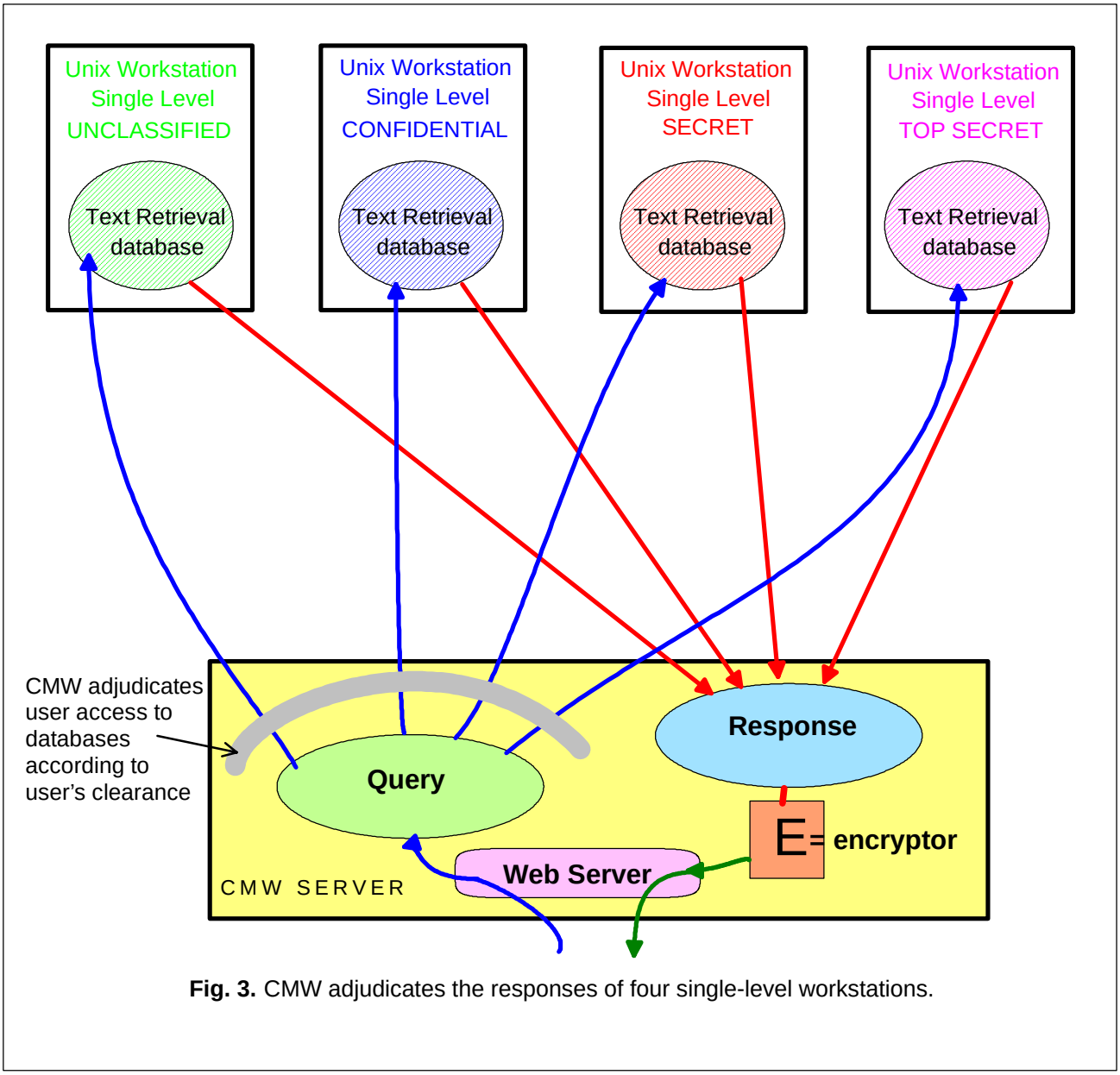

Fig. 3. CMW adjudicates the responses of four single-level workstations.

Non-CMW Architecture

The single-CMW architecture that we implemented had several practical problems that require consideration.

- Maintenance of the trusted CMW code is a difficult and ongoing burden. Every time that a new version of RetrievalWare was released, the code had to be reworked. These updates seem to occur every few months.
- The granularity of the developer hooks in RW is not very fine, so that it is difficult to do privilege bracketing.
- In order to communicate with the unlabeled Windows clients on the LAN, many of the CMW security features must be (temporarily) overridden. It will be hard to convince accreditors that the CMW security assurances are really all in effect.

These problems are partially solved in the multiple computer architecture of Fig. 3, but at the expense of four extra computers. In addition, the use of the CMW front end may not add much extra security because information must still be sent out to the Windows PCs at syslo.

As a response to these concerns, we also examined a multilevel architecture on a single non-CMW Unix workstation as shown in Fig. 4. Because all access to information is via Web access, there are no user accounts on the machine hosting the text retrieval engine and Web server. Thus, because there are no ordinary user accounts on the system, DAC can be used to segregate data without fear of an authorized user giving a file away to an unauthorized user. Unneeded services such as *ftp* and *telnet* will be disabled on the host. The use of a CMW would add resistance to attack, but our system is going to be placed on a classified LAN and is not subject to such attacks. Authorized users on the classified LAN are implicitly trusted by the EDC system. As an example, consider the fact that once an authorized user downloads a classified file to his PC, there is nothing that this system can do to prevent disclosure. However, audit trails that list all retrieved documents will be maintained.

Classification level protection

We propose to use DAC to keep the classification levels secure, and to rely upon the RW engine and some custom code to protect category and need-to-know access. Consider only the problem of creating a system that isolates and protects the four levels of classification Unclassified (U), Confidential (C), Secret (S) and Top Secret (TS). Create four (dummy) users named after the four levels, i.e., U, C, S, TS. Then create four groups with the following membership:

Group	Members
Unclassified	U, C, S, TS
Confidential	C, S, TS
Secret	S, TS
TopSecret	TS

Thus, the user TS has access to all groups, while U can only access Unclassified.

For user access, four Web servers are used, one running as each of the 4 users. Each server would be interfaced with its own RetrievalWare engine, also running as the appropriate user. For example, a user with a Secret classification would connect to the Secret server and the Secret database engine. This server would mount the libraries at or below Secret, i.e., Unclassified, Confidential, and Secret.

The owner of each file in a library would be the user that corresponds to its classification, and the group of each file would be the corresponding group. Therefore, the Secret server would have read access to a Confidential file if the DAC label was 640 (owner=*rw;* group=*r,* world=*nothing*) because the user S (running the Web server) is a member of the Confidential group.

Similarly, a user on the Unclassified server would not have access to the file both because it is not mounted to be searched by the Unclassified Web Server, and because U is not a member of the Confidential group.

The above scheme requires NO changes to the Web servers or the database engine and provides strong protection against inadvertent disclosure.

Access control lists and categories

The security model for categories is that a user must be a member of at least all the categories that belong to a document in order to be able to access it. This model can perhaps be instantiated by means of access control lists (ACLs). For example, on some versions of Unix such as with IBM's AIX, the ACL directive

```
allow John, CAT1 CAT2
```

will only allow access to the file by the user John if John is a member of both groups CAT1 and CAT2.

Under other versions of Unix (e.g., HP-UX 10.x), the access control list mechanisms only permit the association of a single group with a given user, so it would be impossible to enforce the correct security policy with ACLs. The use of ACLs to enforce access control poses

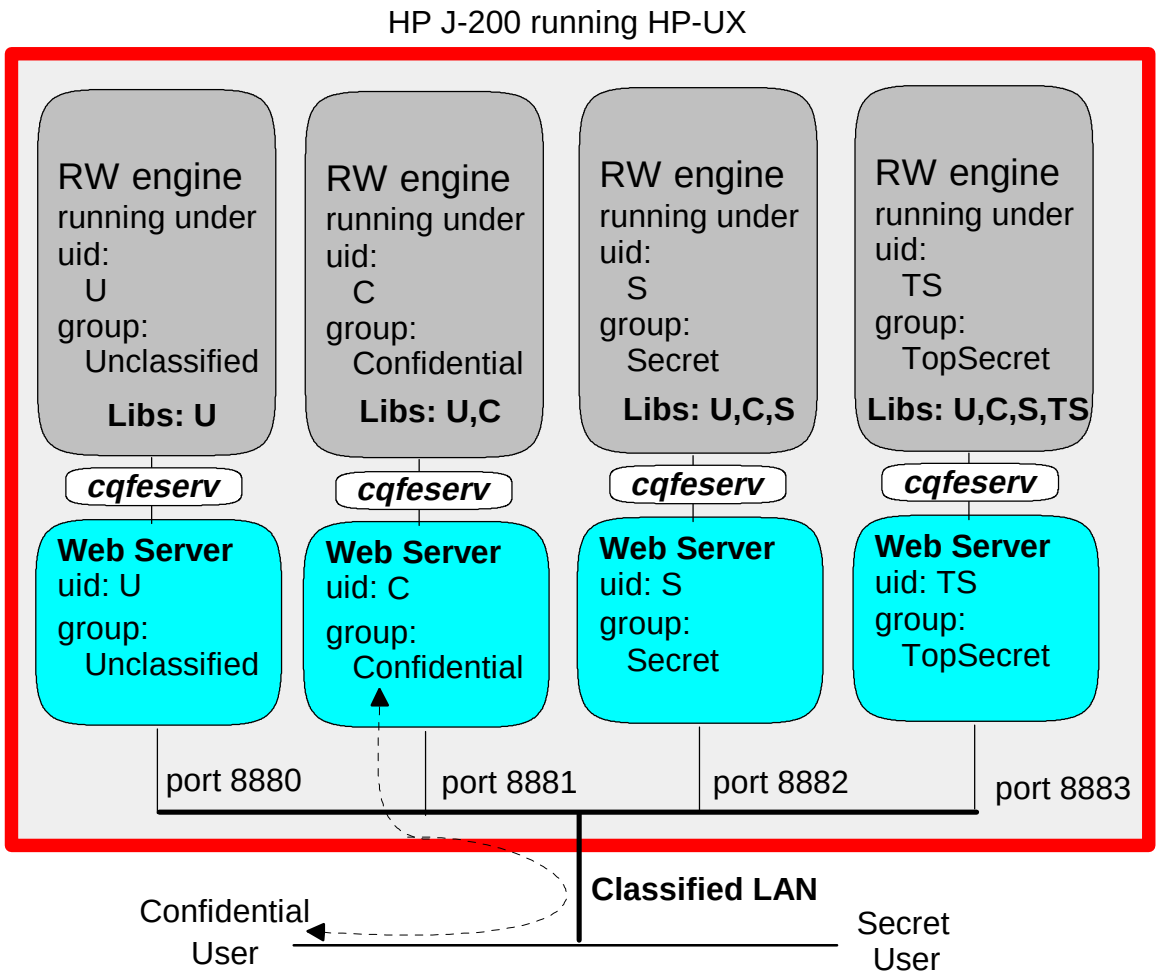

Fig. 4. Non-CMW EFR architecture. The dashed line is the information flow path for a user cleared to Confidential. Each Web server checks the user's classification level before access is allowed.

another severe difficulty. To make use of them, it would be necessary to have the process (the text retrieval search engine) either assume the identity of the user, or else change its identity by becoming a member of only those groups (categories) that are available to the user. Unfortunately, this requires mak - ing changes deep within the code of the text retrieval engine. Such changes entail several difficulties which is the main thing that this scheme was supposed to avoid. Therefore, we have decided that it is highly desirable for there to be NO code changes within the text retrieval modules. As a result of this decision, ACLs will not be used to determine access to a file by either category or need-to-know.

Intervention in *cqfeserv*

The client-server model for the RetrievalWare (RW) text retrieval engine is that there is a Web-based user interface that communicates to a set of "back end" processes by means of an intermediary called *cqfeserv*. *cqfeserv* takes the user's URL request and translates it into a form that the back end databases can respond to. It intercepts all communications going between the Web server and the RW back end as is shown in Fig. 4. We propose to change *cqfeserv* so that categories and need-to-know (NTK) access are enforced.

Managing access control lists is a very time-consuming and tedious process, so we believe that it is only practi - cal to support need-to-know access according to some predefined groups, which we have labeled projects. In this manner, there can be a much smaller database of projects, each one associated with a project manager who will be responsible for maintaining a list of group members for his project. Because File Room personnel have no easy way of determining changes in personnel,

missions, etc., off loading this burden onto the information owner seems appropriate.

Each document will have a keyword field corresponding to its categories and another corresponding to a project. Documents without any special need-to-know controls will have a project "all" meaning that anyone possessing the correct categories and clearance can have access.

To enforce access by category and NTK, we will supplement each query made by the user by a hidden, Boolean query over the category and project fields. If the user does not possess all the required categories and the projects, he will not see any hits on documents outside of his clearance, categories, and NTK. Thus, the user cannot retrieve any forbidden documents. The hidden query concept was deployed on the CMW version of the system and worked successfully. Therefore, the RW text engine will be responsible for making sure that a user does not see hits on any unallowed document categories; DAC, together with separate (access-controlled) servers and proper mounting of libraries will maintain access by classification level.

Just to be sure that the user does not create a URL for a forbidden object manually, when a request for a document retrieval is made, we will check the user's clearance, categories and projects against those of the document and allow or deny access accordingly. All of these operations can be performed in *cqfeserv*. Because all document retrieval requests go through *cqfeserv*, this is also where entries are created for the audit trail.

File labeling

One disadvantage of the non-CMW approach is that the file system itself is unlabeled, and labels are a requirement of B1-level security. There are several methods to overcome this difficulty.

- We propose to insert a PDF classification cover sheet as the first page of every document so that the actual data file will properly contain the document's label. This page will be the first one seen by a user when the document is retrieved.
- The files could be placed in a directory structure that corresponds to the file's clearance and categories. For example, a file labeled *Secret RD NWI* could be in the directory */Secret/RD.NWI/*.
- The correct labels appear as keywords in the data fields of the PDF file

Other advantages of a non-CMW approach

- The CMW releases of operating systems lag behind the non-CMW versions. For example, the 64-bit HP-UX is available in the non-CMW environment and, together with a processor upgrade, would more than double the performance of our existing J-200 workstation
- The normal versions of HP-UX cost about half as much as the trusted version.
- More hardware is supported, for example, multiple Ethernet cards.
- Much less training of the computer operations personnel is required.
- All software changes are in a single location (*cqfeserv*) that is more easily checked in the accreditation process.

Summary and Conclusions

We have presented the architecture of an actual CMW-based multilevel document retrieval system to show the tradeoffs that must be made and the problems that must be solved. Unfortunately, the implementation requires a significant amount (hundreds of lines) of code customization which is quite dependent upon the details of the version of the database engine. The document retrieval part of the CMW architecture was implemented and successfully tested.

To avoid these code modifications, we have proposed two other systems that allow the use of unaltered COTS on the back-end data storage systems.

The political and procedural hurdles that must be overcome to get these systems approved is the subject of future work.

Acknowledgments

The authors wish to thank Patricia Payne and David Dillow for their helpful suggestions.

References

[1]. "Department of Defense Trusted Computer System Evaluation Criteria," DoD 5200.28-STD (Dec. 1985).

[2]. "Introduction to Certification and Accreditation," National Computer Security Center Report NCSC-TG-029 (Jan. 1994).

[3]. See http://www.armadillo.huntsville.al.us/