INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 643-9153 • FAX (510) 643-7684

A Security Mechanism for the Resource Management in a Web Operating System

Herwig Unger*

TR-98-032

September 1998

Abstract

Resource security is maybe one of the most important features for any distributed computating on the Web. In this article a new adaptive approach shall be presented realizing the authorization and identification of a remote user using fingerprints built from a set of typical system data related to the respective user. The suggested approach avoids the use of secured, trusted third machines and adapts access rights using a fine-grained set of confidence levels for a possibly changing group of users.

*Department of Computer Science, University of Rostock, D-18051 Rostock, Germany; Phone: +49 381 498 3403, Fax: +49 381 498 3366, E-mail: hunger@informatik.uni-rostock.de

1 Remarks on the Web Security Situation

With the increase of the use of distributed computing in networks of workstations or even in the Internet security features become more and more important.

Therefore systems like SDSI [4], Kerboros [3] or extended systems like CRISIS [1] were developed to certify the security while working on a distributed wide area computer architecture. All these systems need some kind of a centralized, secure and therefore trusted third machine. Mostly the third machine has to confirm the authorization or to give a certificate or a part of a symmetric code to requesting machines. It might be possible as in CRISIS to have a decentralized secure instance even with a redundancy. Nevertheless, it is required to have anywhere a (maybe distributed) catalog of all approved users what seems to be difficult in a global dimension with the respective steady changes. A great step forward seems to be the JANUS-system [2] 'sandboxing' locally running applications that are not fully trusted without having complex login informations. Therefore it will be evaluated, if an application uses dangerous system calls (what also may cause an interruption of non-dangerous programs).

2 Basics

From our point of view, it does not seem to be realistic to use a kind of login procedure in a web-wide system (e.g. the WOS) because fixed user catalogues cannot be established [5] In addition, it make no sense to know who is the user rattening my machine, if he is in a country where I cannot get hold of him. Finally, we argue that each authorization using a login procedure with a password or code can easily be affected. Even if the data are very complex they had to be fixed for some times.

Therefore, the main problem does not consist only in the authentication of a user but in the question 'Whom I can trust?'.

From the working mechanisms of the WOS-warehouses described in [5] it can be expected that a limited, but - maybe slowly - changing group of people will frequently use a considered machine because they find there most services with the required quality for their work. In such a way, local self-adapting catalogues for the authorization and confidence checks can be reasonably established.

That is why we need a kind of an intelligent mechanism to learn, who has the right to do something on a machine.

There are some basic remarks about such mechanisms.

1. Confidence is a mutual thing and need exchange of information in both directions.

2. There must be different levels of confidence with maybe flowing transitions. From the point of view of the WOS-demon this could be:

   - *Level 1:* Allows only the execution of so called elementary jobs which do not need any input data from the remote machine and only generate output data allowing a secure transmission to the requesting machine. In this case no system damage can be occur if the WOS-demon works as desired.

- *Level 2:* Special data files may be transmitted to the executing machine. Check mechanisms may be active reducing the speed.
- *Level 3:* In addition to 1 the transmission of input data is allowed by giving URL's and an transmission initiated by the service executing machine.
- ...
- *Level n-1:* Interactive Jobs are allowed.
- *Level n:* All transactions are allowed with the same access rights like for an user with a regular login in a UNIX-environment.

The more fine-grained levels exist the better the work of the system can be organized.

3. Confidence levels may also influence the access to different warehouse data.
4. There are 'rules' describing under which circumstances a user has a special confidential level and when he will be transfered to another one depending on the 'character' of the respective administrator. Fuzzy rules and fuzzy decision systems will be a good possibility for doing so.

For a better solution of the described problem, it shall be shown in this contribution that each user has data allowing its proper authorization with level-1-services and authenticating himself. Hereby each machine will use its own set of level-1-services for doing so. Therefore the amount of data which must be affected for an attack is so high that their simulation might be very complicated or impossible. So a set of typical system parameters become something like a fingerprint which is unique for any user. Useful data building this fingerprint could be for instance:

- the IP-address of the respective machine giving a more or less location of the user
- the time of the last collaborative work (random parameter)
- the configuration of the remote machine, e.g. the number of devices
- the size and structure of the file system of the remote user
- the data in special files, e.g. the 'last'-records
- a set of typical running processes, e.g. from the prefered interface system
- checksums or the size of some files (especially of the level-1-service files)
- transmission and average processing times
- ....

A selection of a set of these parameters might be flexible but their values and the changes in a considered development might give an impression if the user is really authentic and can also give some clues to determine his confidence level. Note that some of the above parameters can be considered to be constant for a longer period while others will be changed all day by the work of an (active) user.

Depending on these data the authentication of a user should be doubted if for instance

- the size of typical files like the .cshrc or the allocated space on the harddisk were rapidly decreased (for a normal user these parameters mostly grow up)
- significant changes in the home directory-root are made
- fixed or almost fixed values are changed, e.g. the number of peripherical devices, the user ID or checksums
- the dynamic in the development of any parameter is changed
- the remote user was not seen for a long time
- ...

In such a manner another important argument should be how often two machines have used each other for the execution of a service request and how often therefore information were exchanged. The level of confidence on the other machine might be evaluated as well as how often a machine has given a 'hello'-request to be kept in memory. Furthermore, the higher access possibilities in higher confidence levels may also be used to refine the remote user identification process using other than level-1-services.

3 Implementation

3.1 Structure of the System

Figure 1 shows the structure of the system realizing the above described security mechanism in the context of the WOS^{TM}-System, although an application in other systems will be easy, too.

The shown system works as follow. If a service request requires the execution of programs with a confidence level of 2 and higher, the system activates an user check by the 'Access Control Unit'. This unit makes the 'fingerprint'-data via a call of selected level-1-services (which cannot harm the machine as discussed above) by the 'Answer Unit' of the remote machine as well as the data eventually collected about the remote user in the local security warehouse available. These data allow the 'Decision Manager' to accept or refuse the respective service execution. Making this decision also requires the use of a rule basis containing some imaginations of the administrator, which data shall be used with which weights for that process. Last but not least each decision made may influence the confidence level which is kept in the local security

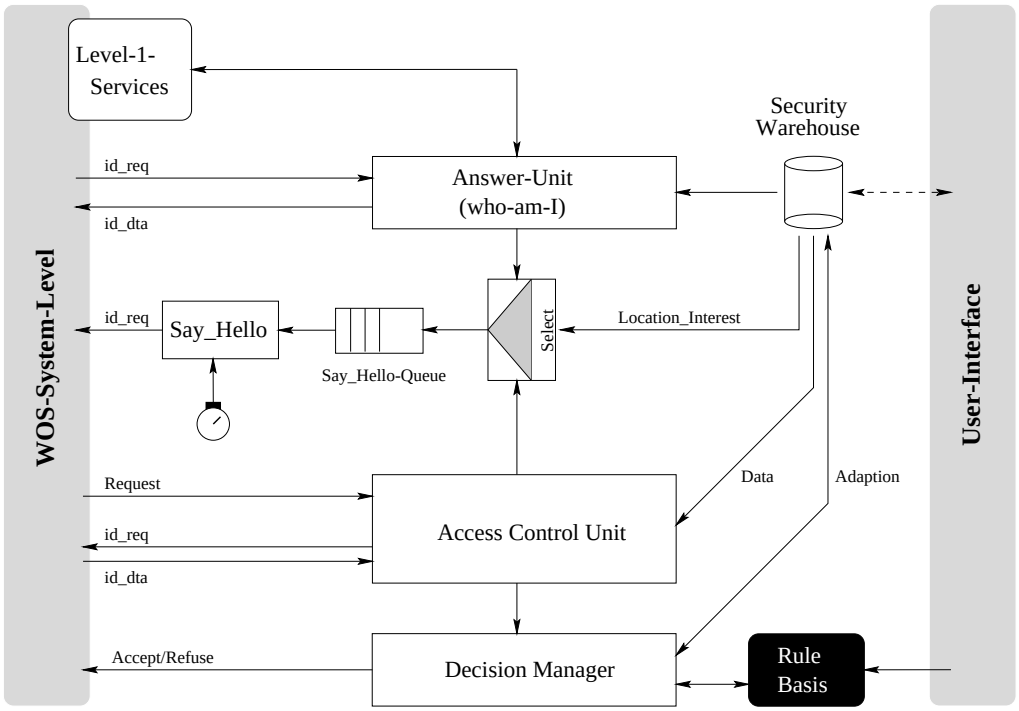

Figure 1: Structure of the WOS^{TM} Security Component

warehouse for the remote machine. This adaption will also be done by the 'Decision Manager'.

Both- the 'Access Control Unit' on the server side as well as the 'Answer Unit' on the client side are able to put remote machine data into the *Say-Hello-Queue* to provide a later data exchange for increasing the confidence level for the work with that machine. The data of the local security (and service-) warehouse will be used to limit the number of machine to the really interesting ones. Any user is also able to specify such machines directly via its interface in its local warehouse.

3.2 Experiences

While final implementation in the WOS-system is still going on the above described system values were recorded and processed for a set of $m = 40$ people by an independent UNIX-subsystem. This program was written when the first successful experiences were obtained from a manual consideration of data collected by a UNIX-script. The main results are that

1. three parameters were already sufficient to identify each of the 40 users every time when it was requested (10 parameters were recorded at all);

2. one group of parameters could be confirmed to be fixed (uid, gid) or to be moved within a very small changing range only (e.g. the size of the .cshrc file);

3. using an average value v with exponential oblivion $exp^{-\lambda}$ recursively built from the measured data x_i ($v_{new} = v_{old} * exp^{-\lambda} + x_i/N$) and its first derivation a

prediction of changes for the group of changing values (e.g. total size of the home-filesystem of a user) can be made with an sufficient exactness for the identification process.

In detail we can outline the following results and experiences from our practical measurements.
Different parameters g_i were taken for all users $N_1, ..., N_m$ at n times in a period of about 15 days. At first, the values $g_i(N_j)$ were used to determine their character (constant, slowly or fast changing). For changing parameters a prediction was established in the above described manner, selecting an oblivion depending on the speed of changes of the respective parameter. If the difference of the predicted and measured average value was larger than 3 times the variance σ the respective measurement was wasted and not used in the identification process.[1]

Otherwise the data of the users were processed. Therefore the possible deviation $\epsilon \in (0, 1)$ was introduced to determine whether two measurement values $g_i(N_k)$ and $g_i(N_l)$ are equal or not. Two different users can be distinguished for a given ϵ using the parameter g_i , if and only if

$$|g_i(N_k) - g_i(N_l)| \geq \epsilon \frac{(g_i(N_k) + g_i(N_l))}{2}$$

For each measurement t the *instantaneous significance* (the probability that any two of the m users can be properly distinguished with the data from this measurement only) $P_{g_i,t}$ for a given parameter g_i can be obtained by

$$P_{g_i,t} = \frac{|\{(k,l) : |g_i(N_k) - g_i(N_l)| \geq \epsilon \frac{(g_i(N_k)+g_i(N_l))}{2}\}|}{m(m-1)/2}$$

The average value of n instantaneous significances P_{g_i} of the same parameter g_i is called the *significance* of that parameter. For the size of the home partition of the users (P_{home}), the number of files in the root of each home directory ($P_{number-of-file}$) and the size of the .cshrc file ($P_{.cshrc}$) the respective values are shown in Table 1.

Possible Deviation ϵ [percent]	1.0	0.1	exact, 0.0
P_{home} [percent]	98.1	100.0	100.0
$P_{number-of-file}$ [percent]	93.8	99.0	99.0
$P_{.cshrc}$ [percent]	89.5	96.7	99.0

Table 1: Probabilities that different Parameters identify a user

Because all used machine values are independent, the *global significance* P_G that all users can be identified by the used set G of parameters g_i is given by

$$P_G = 1 - \prod_{i=1}^{|G|}(1 - P_{g_i})$$

[1] Because the 3σ interval is very slowly 'learnt' with the necessary exactness by the system, suitable (start-) data shall be given as it was done for the experiments.

where $|G|$ determines the number of considered parameters and P_{g_i} is the significance of the parameter g_i. In such a manner, the more data with a high significance P_i are used the better P_G converges to 1.

4 Conclusion and Outlook

It is shown that users from a remote machine can be identified (authorized and authenticated) using a set of data obtained from their own hard- and software. These data can easily be determined with so called level-1-services, which cannot cause security problems. In contrast to standard password procedures these parameters are difficult to be affected, once more because of the selection and evaluation of the set of parameters may be influenced by the owner of any server taking part in the collaboration. Furthermore a method for the determination of confidence levels and therefore remote access rights was suggested basing on these parameters and the behavior (activities) of the remote user.

The work will be continued including the described approach in the Web Operating System. This allows its verification in a large frame and shall give a real possibility to increase the security of such a wide area distributed system.

Acknowledgment

The author would like to thank the International Computer Science Institute (ICSI) in Berkeley for the support of his research from July until September 1998.

References

[1] E. Belani et al., *The CRISIS Wide Area Security Architecture*, Technical Report, University of California at Berkeley, http://now.cs.berkeley.edu/WebOS, (1998)

[2] I. Goldberg, D. Wagner, R. Thomas, E. Brewer, *A Secure Environment for Untrusted Helper Application*, In Proceedings of the Sixth Usenix Security Symposium, (1996)

[3] J.G. Steiner, B.C. Neumann, J.I. Schiller, *Kerboros: an authentication service for open network system*, In Usenix Conference Proceedings, Dallas, Texas, 1998

[4] R.L. Rivest, B. Lampson, *SDSI-A Simple Distributed Security Infrastructure*, http://theory.lcs.mit.edu/ cis/sdsi.html, (1998)

[5] H. Unger, P. Kropf, G. Babin and T. Böhme, *Simulation of search and distribution methods for jobs in a Web Operating System (WOS*, In A. Tentner (ed.) 1998 Advanced Simulation Technologies Conference (ASTC 1998), Boston, (1998)