An Efficient Access Mechanism for Database Systems

Arun.S[1], Mahalakshmi.R.M[2], Bhoopathi Siva.K[3]

[1]Department of Computer Science, R.V.S Faculty of Engineering
Coimbatore, India

[2] Department of Computer Science, R.V.S Faculty of Engineering
Coimbatore, India

[3] Department of Computer Science, R.V.S Faculty of Engineering
Coimbatore, India

[1]arunshanmugam1211@gmail.com
[2]sachuyuva007@gmail.com
[3]bhoopathi001@gmail.com

***Abstract*— Break-glass within computing is a term used to describe the act of checking out a system account password for use by a human. It is generally used for highest level system accounts such as root for UNIX or SYS/SA for database. These accounts are highly privileged and not in themselves individualized to a specific human, so instead break-glass limits them by the password time duration, with the aim of controlling and reducing the account's usage to that which is necessary. Break-glass has been examined in a number of publications applied to medical systems. What is currently missing is an accurate translation of original break-glass concepts, especially applied to high security environments such as banking. This paper will provide a description of how break-glass is evolving into a broader method of time-based access control mechanism. Finally how time-based access control and break-glass can be varied adaptively based on threat level is proposed.**

***Keywords*— Access control, Break-glass, Database, Administration, Security.**

I. Introduction

The origin of the term Break-Glass is from publicly accessible fire-alarms. It will be useful to use this context to aid in the definition of the term, and compare the evolution of Break-Glass within fire alarms to the evolution of break-glass to modern computer systems, as we may be able to predict the future from previous historic trends.

The first fire alarm networks were installed in Berlin by Siemens in 1951[1] [2] [3], closely followed by Boston in 1852[4]. The Boston system was based on a telegraph-ic network of 45 boxes which enabled a local person to electronically signal to the central fire department that there was a fire so they should come and put it out quick-ly [5]. The ability to set off the publicly situated Boston alarms was through a key bestowed to a small number of local responsible individuals (police officers etc). This was to avoid false alarms by accident or malice. The unfortunate counter effect of this was that in the 1872 Great Fire of Boston, the fire department were delayed by 20 minutes due to a lack of a key-holders to raise the alarm[6] resulting in the deaths of 30 people.

Subsequent to this event, new public fire alarm systems were installed in Glasgow 1878, and in London 1880. The use of a purposefully designed, Break-Glass mechanism, with contemporary documentation is the 1880 system introduced by Charles Bright. The main advantage of this system being that the ability to "Raise Fire Alarm" privilege could be granted to the public without increasing false alarms to the point of making the system unusable. In other words a publicly available fire alarm system would be less abused due to the break-glass protection.

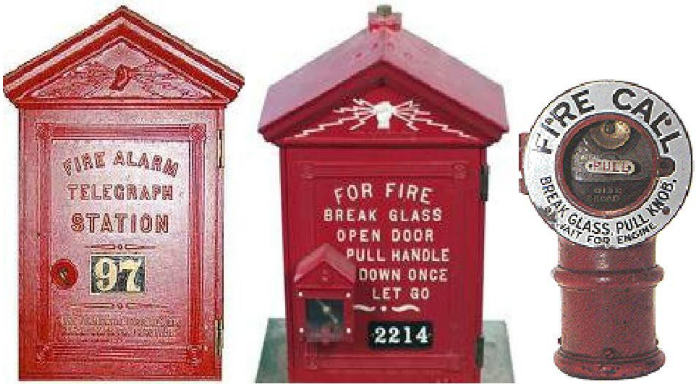

Figure 1 Gamewell Fire Alarm, break glass adaption and Bright's built in break glass.

After success in London and in Glasgow [10] [11] with built-in break glass the US followed with break-glass add-ons in about 1900[5]. Because the US fire alarms had already been installed prior to the use of break-glass a small box had to be added onto the front of the already existing Fire Alarm Boxes.

From this we learn that Break-Glass has had the following properties.

- Shares a single privilege between many users (the "raise fire alarm privilege").

- Identified initiating individual by creating loud noise thus calling attention to the initiator and increasing the risk they will be identified and caught in the case of a deliberate false alarm.
- Time limit for the use of the privilege, as the break glass would be reset after the fire.

The above is important so that we can derive the essence of what break-glass actually means. So for instance it can be said that Break-glass is not a method of escalating privilege – it is a way to reduce abuse of a shared privilege, which is not in itself individualized.

To be of use the break-glass request should be easy to do quickly, not denied, with the emphasis on preserving safety rather than preventative security. Safety in the case of computer systems is analogous to preserving availability i.e. the health of a system. Security questions and ramifications are examined after the break-glass event and punitive measures taken if necessary. This has been described as optimistic security mechanism[12], though this should not be taken as being a naive approach – in fact it is pragmatic and in many cases the only way of realistically decreasing risk for sys-tem accounts in a large distributed network, where historically the shared system ac-counts have been allowed to be passed from colleague to colleague and not reset regularly, resulting in administrators no longer employed by the organization, still know-ing the password.

II. ACADEMIC REVIEW RELEVANT TO BREAK-GLASS AND TIME-BASED ACCESS CONTROL (STARTING FROM 2000 THROUGH TO 2012 IN CHRONOLOGICAL SEQUENCE)AGE LAYOUT

The earliest dated document describing a functioning break-glass design is Charles Bright's London network of publicly accessible fire alarms [7]. In the field of com-puter science the first reference that posits break-glass as a potential idea for computer systems is Povey – 2000[12]. If one digs into commercial practice it is possible to see that the first commercial company to sell a break-glass solution publicly was CyberArk[13] with their EPV product launched in 2003. We should be aware that prior to CyberArk's publicly available commercial software there were inhouse solutions within the banking industry that carried out the same functionality. This is to be expected as break-glass is the most practical method of gaining control of a shared credential for a privileged account. If an organization cannot limit the account to a single user i.e. it's geography, then the other dimension for reducing risk would be limiting the time for which that credential will be effective – hence break-glass and time-based access control.

More accurately Break-Glass provides a method to associate human identity to a system account e.g. root, and limits the potential insecurity of allowing usage of uncontrollable privilege by limiting how much time it has been used for, and by automatically changing the password at the end of the break-glass period, and by warning the user that their actions will be monitored, and punished if abused.

The key feature of a break-glass system is that it automatically resets the password of the account in a pre-defined time frame. Generally a logged out root password would be reset within 24 hours. Having read the small number of papers on break-glass, none of them precisely and fully define either the concept as per the original break-glass concept, or the practice as experienced within financial services - hence the requirement to write this paper. However it is useful to list the papers that intersect this topic which are medical industry based.

The automated reset of an account password based on time is discussed in [14] and called as timely revocation of trust. Although related to break-glass this is not same because break-glass does not care if the revocation of the privilege is at an appropriate time, it just does it in 24 hours – whether the user needs the privilege any more or not. If the user gets locked out when doing their work, they have to break the glass again. Yet another related concept is authorization based on time of day e.g. Fred is allowed access only during the day time and not at the night time. There are more number of these authorization context papers [15][16][17] which are worthy but different from break-glass in that they are dealing with a fixed portion of the day which is a repeated authorized window, rather than a break-glass session which is – you have 1 hour and then you are kicked out. Yet another related "temporal concept" is to limit the synchronous privileges. For example a user can only have one role at a time or a role can only have 5 users at a time[18]. This is related in that one of the benefits of break-glass as a time-based access control mechanism is that for that break-glass period there would normally only be one person on that machine at that time. However if a team effort were need-ed break-glass systems do not prevent the recipient of the password for that time period delegating the password to their colleagues – but at the end of the break-glass time period the password will forcibly automatically change. This concept is named the 'Emergency Lifetime' of a privilege in Georgakakis 2011 and is a useful term, applicable to the traditional meaning of break-glass. What is interesting is to see how the use of "Lifetime" of an account credential and access is being transferred from the emergency only scenario to the Business As Usual access (BAU). The idea being that all human access either individualized or through a system account should be on time limited basis by default. Instead of accounts that by default last forever, which is the position of current systems. Accidental non-removal of logged out users is one of the greatest sources of risk in financial services systems.

One of the most recent context-based break-glass papers is Marinovic 2011 which essentially lists rules upon which to either permit or deny break-glass access. Interest-ingly the scenarios suggested in the paper do not include the scenario of denial.

The Marinovic-2011 paper proposes a mechanism for denying break-glass access defeats the object of the mechanism, however, I am not a Medical professional, so there may be ontological differences between the two subject areas that explain this difference, and it would be interesting to develop this conversation in the future as time allows.

On this note, the final paper reviewed was a paper specifically detailing the use of OWL to define Temporal Access Control constraints within ontologies such as in medical and financials[19]. OWL is nothing but a Web Ontology Language that provides mark up language to define semantic ontological means for different subject domains. For a concise and usefully abbreviated form of OWL please see the Manchester Syntax [20].

III. OBSERVATIONS ON THE ACADEMIC BODY OF KNOWLEDGE RELATED TO COMMERCIAL BANKING PRACTICE

A Generally, system access using break-glass has two main trends.

The first is for more categories of access to be time limited so that access to OS, DB and even MS Office software is becoming managed using time-based access control mechanisms[22]. The conception of all accounts being time-limited is positive for reducing risk. One of the most common security issues in a system are the open accounts from employees that have left the organization for reasons, such as churn, redundancy or death. The lifecycle of human accounts over time is moving away from the default –this account lasts forever stance, to this account should be recertified once in every year stance. Various considerations for handling digital identity in the case of human death are discussed in this innovative paper published at the same NSPW conference as original Optimistic Security, which can be referenced at http://www.nspw.org/proceedings/2011.

Secondly, the categorization of privileges which in the past may have been recognized as BAU, but are now being moved under break-glass shows a trend towards removal of human intervention in terms of ongoing system administration and maintenance. This consolidation is the expected result of autonomic computing [24].

Given that this trend towards consolidation and automation is increasing with future software such as the Cloud based 12c database being released by Oracle in 2013, it can be seen that the removal of human intervention will be an increasingly interesting topic, especially from a security point of view. The word Sabotage originates from the introduction of automatic weaving equipment which was deliberately damaged by the employees paid to work the machines. They used their sabots (clogs) to wreck the new looms, that were about to relieve them of their jobs. Business, vendors and security professionals will be mindful to avoid alienating workforces during this consolidation of human resource, and also very keen to be able to apply real control over privileged admin accounts – hence this paper.

IV. BREAK-GLASS SECURITY APPLIED TO BUSINESS SCENARIO

The plot is a large estate of 10,000 databases in a financial services organization and the vendor platform is Oracle Database on Red Hat Linux.

The key account that would be subject to break-glass on Oracle Database is the SYS account as it is a non-individualised system account with very high privileges. Additionally the SYS account in Oracle is immune to all security controls managed within the database system i.e. there is absence of password complexity verification, history, account locking or failed logon delay (aka connection throttling). The reason for this is that Oracle are very availability focused and wish to avoid the scenario where the administrator is accidentally or purposefully locked out from the server. Unfortunately this also means that the password for SYS could be weak and an attacker may be able to get in. Counter intuitively all the other accounts in Oracle do have proper security controls. The reason is that these are less important to remain unlocked. So we have a situation where the most security sensitive account has the least security controls. This is where are a centralized break-glass server has a role to play, as the break-glass passwords are set externally from the database they can be set to be long random values and verified to be secure. Additionally they can be changed on a regular basis automatically. This means that the main weakness of security in Oracle is fixed by the use of a centralized break-glass server such as OPAM or CyberArk EPV.

There are some technical red flags to this.

First one is that the communication between break-glass server and the databases needs to be encrypted to protect the automated password changes.

Second one is that the break-glass server should to be secured, as it contains all the passwords.

Thirdly OS access to the DB server needs to be secured to prevent DB access locally. These are all achievable though not fully realized at this stage, though there are extra security considerations for controlling advanced administrative access.

Most importantly, a chance for greater efficiency exists, because break-glass time-based access control lends itself to being adaptively varied depending on security level.

V. ADAPTIVE BREAK-GLASS

The main drawback of break-glass systems, expressed by humans, is the action of having to break-glass takes too long and slows down emergency response and general day to day administration of system. This drawback is liable to gain traction when an estate has never had a security issue in it's history. The psychology of the human teams involved tends towards laissez-faire security i.e. just enough[25]. Because database estates have low frequency of security events the tendency is to drop the guard. Problem is that if a security event does happen it could well be catastrophic i.e. end of business. Therefore the risk profile is broadly similar to that of a nuclear power station.

If we analogize DB security with personal human security, it would be strange for a human to walk round with their guard up wearing a crash helmet all day when walking the street, having a coffee or sat at their desk. This security posture is inappropriate for today's life style. So why do the database estates have a single security posture that is set at high every time. It would be more sensible for sure to adapt security level dependent on the threat level which varies over time. A framework for measuring, controlling and responding to threat level has already been discussed in Mutually Adaptive database paper at http://www.journalofdatabasesecurity,com/[26].

All that needs to be done is for the break-glass mechanism should be integrated into that adaptation mechanism, so that the break-glass session length is varied depending on the measured threat level. For an instance, a break-glass ticket could last 1 day normally, but if there were a lot of failed log in attempts detected; ticket could automatically shorten to one hour, thus increasing security whenever it is needed. This would be of great business benefit to the adopter as they could have more efficient systems when it was safe to do so.

The other big objection to break-glass systems is that they are usually separate servers managed by separate teams often on different platforms (e.g. MS Windows/AD) and therefore not fully trusted by the Unix/Oracle team both in political and reliability terms. The future of break-glass technology is to build in the break-glass authorization mechanism to the entire database, as in Bights Fire posts in London [9].

VI. CONCLUSION

This paper has put together the definition of break-glass access control using the historical meaning from fire-alarms extended to contemporary banking security and drawn comparisons to how each technology evolves to having break-glass built-in.

Then this paper proposed break-glass as control that could be responsive to threat level. this paper will contribute to understanding future directions for access control.

ACKNOWLEDGMENT

This work was supported by Computer Science Department in R.V.S Faculty of Engineering of Coimbatore.

REFERENCES

[1] J. Kastl and L. Moore, "Wily welfare capitalist: Werner von Siemens and the pension plan," Cliometrica, vol. 4, no. 3, pp. 321–348, Dec. 2009.

[2] B. Klaedtke, M. Thissen, A. Damaschke, and P. Korte, "Feuerwehrchronik," vol. 30, no. Teil 1, 2011.

[3] "Ernst Werner von Siemens - QFINANCE." [Online]. Available: http://www.qfinance.com/operations-management-thinkers/ernst-werner-von-siemens. [Accessed: 11-Jan-2013].

[4] "Boston stands by its fire alarm boxes - The Boston Globe." [Online]. Available: http://www.boston.com/news/local/articles/2008/01/06/no_cause_for_alarm/?page=full. [Accessed: 10-Jan-2013].

[5] "April 1988 Fire Alarm and Police Call Boxes." [Online]. Available: http://www.cjow.com/archive/article.php?month=4&a=04Fire+Alarm+and+Police+Call+Boxes.htm&year=1988. [Accessed: 10-Jan-2013].

[6] "Great Boston Fire of 1872 - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/Great_Boston_Fire_of_1872. [Accessed: 10-Jan-2013].

[7] W. Smith, "A system of electric fire-alarms," Telegraph-Engineers and Electricians, Journal of the Society of, vol. 13 ,, no. Issue: 51, pp. 51 – 64, 1884.

[8] C. E. Spagnoletti and E. Ayrton, W.E. ; Bright, Charles ; von Treuenfeld, R.F. ; Spratt, G.C. ; Smith, Willoughby ; Bright, "Remarks on A system of electric fire-alarms," Telegraph-Engineers and Electricians, Journal of the Society of, vol. XIII, no. 51, pp. 65–73, 1884.

[9] "London Fire Brigade - Fire alarm posts. Left - telephone alarm direct to fire station, right - break glass alarm." [Online]. Available: http://www.london-fire.gov.uk/equipment_fireAlarmPosts.asp. [Accessed: 14-Jan-2013].

[10] "THE TARDIS MATERIALISED IN GLASGOW!," On The Run - Strathclyde Fire and Rescue Retired Employees Association, no. 3, 2007.

[11] R. W. Stewart, "The Police Signal Box : A 100 Year History," no. June 1993, pp. 1–17, 1994.

[12] D. Povey, "Optimistic Security : A New Access Control Paradigm," pp. 40– 45, 2000.

[13] "Cyber-Ark - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/Cyber-Ark. [Accessed: 10-Jan-2013].

[14] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," ACM Transactions on Information and System Security, vol. 4, no. 3, pp. 224–274, Aug. 2001.

[15] S. K. Tzelepi, D. K. Koukopoulos, and G. Pangalos, "A flexible content and context-based access control model for multimedia medical image database systems," Proceedings of the 2001 workshop on Multimedia and security new challenges - MM&Sec '01, p. 52, 2001.

[16] C. Bertolissi and M. Fernández, "Time and location based services with access control," New Technologies, Mobility and Security, 2008.

[17] A. D. Brucker and H. Petritsch, "Extending access control models with break-glass," Proceedings of the 14th ACM symposium on Access control models and technologies - SACMAT '09, p. 197, 2009.

[18] Z. Liu, Y. Jia, and T. Sun, "Study on Role-Based Access Control with Time Constraint," 2011 Seventh International Conference on Computational Intelligence and Security, pp. 1046–1048, Dec. 2011.

[19] M. Kim, J. B. D. Joshi, and M. B. Spring, "An Architecture for Specification and Enforcement of Temporal Access Control Constraints using OWL Categories and Subject Descriptors," pp. 21–28.

[20] "OWL 2 Web Ontology Language Manchester Syntax (Second Edition)." [Online]. Available: http://www.w3.org/TR/owl2-manchester-syntax/. [Accessed: 11-Jan-2013].

[21] C. a. Ardagna, S. De Capitani di Vimercati, S. Foresti, T. W. Grandison, S. Jajodia, and P. Samarati, "Access control for smarter healthcare using policy spaces," Computers & Security, vol. 29, no. 8, pp. 848–858, Nov. 2010.

[22] "Application access control system - Patent # 8104076 - PatentGenius." [Online]. Available: http://www.patentgenius.com/patent/8104076.html. [Accessed: 11-Jan-2013].

[23] M. E. Locasto, M. Massimi, and P. J. DePasquale, "Security and privacy considerations in digital death," Proceedings of the 2011 workshop on New security paradigms workshop - NSPW '11, p. 1, 2011.

[24] "IBM Research | Autonomic Computing." [Online]. Available: http://www.research.ibm.com/autonomic/. [Accessed: 11-Jan-2013].

[25] "Schneier on Security: Laissez-Faire Access Control." [Online].Available:http://www.schneier.com/blog/archives/2009/11/laissez-faire_a.html. [Accessed: 11-Jan-2013].

[26] "Generate a random string for a password in Oracle | Complicated things possible." [Online]. Available: http://hugepang.wordpress.com/2011/03/25/generate-a-random-string-for-a-password-in-oracle/. [Accessed: 15-Jan-2013].

[27] S. Oxley, "Mutually Adaptive Distributed Databases : Using responsive control loops on distributed database audit trails to form herd-like security protection against brute-force attacks B . Implementation – Oracle DB estate within financial services," The Journal of Database Security, pp. 1–11, 2012.