13th ICCRTS

"C2 for Complex Endeavors"

Multi Level Security, $3\frac{1}{2}$ decades later

Topics: (2) Networks and Networking; (7) Network-Centric Experimentation and Analysis; (8) C2 Architectures

Capt Erik Muller [STUDENT]*†, Tim Grant*, Erik Poll†

POC: Erik Muller*†

* Netherlands Defence Academy
P.O. Box 90.002, 4800 PA Breda, The Netherlands
Tel: +31 76 527 3278 / Fax: +31 76 527 3259
E.Muller.01@nlda.nl / E.Muller@cs.ru.nl

† Radboud University of Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands

Abstract

This paper will review the state-of-the-art in research into MLS, almost four decades since their original introduction, based partly on the first authors recent experience in FOB Ripley (Afghanistan). Examples will also include C2 models and systems being developed by the Royal Netherlands Army. Finally, the paper will draw conclusions and recommend further research in order to meet todays, but even more important, tomorrows complex challenges in military operations.

Introduction

The use of digital or digitised information in the military battlefield has increased tremendously over the past few decades. Complex missions require more information sharing, not just between military units from different services and nations, but also between governmental organisations (GOs) such as Ministries of Foreign Affairs and of Overseas Development, non-governmental organisations (NGOs), the media, repair organisations, and suppliers. In the twenty-first century, C2 decision time needs to speed up to provide military units with the agility to respond more quickly to dynamic situations, whilst adversaries, - symmetric or asymmetric - become increasingly keen on intercepting classified information. The increasing complexity and speed, as well as growth in caveats, International Defence Organisation (IDO) markings and classification levels etc. underline the need for methodologies and concepts on how to handle differently marked information in a Network Enabled Capabilities (NEC) setting.

Since the 70's, research has been ongoing on the subject of Multi Level Security (MLS) and has resulted in several theoretical models describing MLS systems and some implementations of MLS-like systems. Most of these MLS-like systems and models are merely based on Nato Tempest standards when it comes to Emission Security (EMSEC), resulting in physically separated networks or on encryption techniques, often used to securely tunnel one security level over another.

Current implementations are still largely based on notions developed in the 70's or even earlier. It is not clear that this basis is still adequate, given the increased complexity, need for security and the agility to quickly adapt to the changing environments of today's and tomorrow's worlds. We have to be prepared to re-evaluate this basis and be prepared to consider alternatives.

MLS, Bell-LaPadula, and Biba

Strictly speaking, MLS (Multi-Level Security) just means that information is classified into multiple levels. The standard example of such a hierarchy

of multiple levels involves the levels unclass-confidential-secret-top secret, but richer hierachies are possible. More generally, arbitrary lattices can be considered as hierarchies of security levels [8].

In practice however, the term MLS is used for the combination of of such multiple levels with mandatory access control (MAC) enforcing the restrictions of the Bell-LaPadula model [2], i.e. 'no read up' and 'no write down' (aka the *-property). We will call this "Classic MLS" to avoid any confusion. Such a policy ensures confidentiality even in the presence of malicious users and the presence of Trojan Horses, i.e. if users are tempted into using malicious applications.

Confidentiality means that information at level X can only depend on information at the same level X and below. It can never depend on information at some higher level Y, which is obviously more confidential. In other words, no information can leak from a level X to any lower level. This can be formally proven.

The big problem with Classic MLS is its complexity and restrictivity, both when it comes to implementing and using it. In fact, Classic MLS is notorious as a white elephant in the security community, and considered to be an unworkable approach by many.

Biba [5] is the dual to Bell-Lapadula, to ensure integrity rather than confidentiality. It also involves some lattice of security properties, but the MAC policy now enforce 'no write up' and 'no read down'.

Integrity means that information at level X can only depend on information at level X or higher. It can therefore never depend on information at any lower level Y, which is less trusted. In other words, no information can leak from a less trusted level X to a more trusted level Y. This can also be formally proven.

Declassification of Information

Bell-LaPadula and Biba are both notoriously restrictive and inflexible, meaning that systems implementing them are difficult – and frustating – to use.

To overcome this problem, one can add some mechanism of declassification. Adding a declassification mechanism to Bell-Lapadula means that a 'write down' can under certain conditions be regarded as acceptable. It is for example possible to introduce a 'swivel-chair interface'. That means, a trusted person who is in charge of transferring information from a higher security level to a lower, i.e. copying data from one system to another by using a removable storage device. This information would of course first have to be cleared for declassification by it's author or creator or any other authorised person.

With the possibility of declassification, the original confidentiality property can no longer be guaranteed: the declassification constitutes a delib-

erate loophole, and one can only hope - or trust someone - that it is not abused.

Dually, one can add declassification to Biba by allowing some 'write up'-transaction, subject of course to restrictions. (Maybe this should be called reclassfication rather than decalssification, as the "de" in declassification suggests a classification downwards in the hierarchy.)

Ensuring both Confidentiality and Integrity

If we want to ensure integrity *and* confidentiality, we can combine the restrictions of Biba and Bell-LaPadula. This ensures there are no flow downwards (breaking confidentiality) or upwards (breaking integrity). But this means no communication between different security levels is possible at all: every process can only read and write information on a single level. The combination Of Bell-LaPadula and Biba does provide us a "near-perfect" multi-level secure system with only one remark: there will be no information-flow in any direction.

The whole idea of the lattice is then gone: each level is completely unrelated to any other level. A system implementing this combination with n security levels is equivalent to n completely independent and unconnected systems, one for each security level. This corresponds to the idea of MILS (Multiple Independent Levels of Security).

System High and High Watermark

Bell-LaPadula restrictions are not only frustating to use, but the complexity also makes them difficult to implement in a system: in particular, for every process in the system we have to keep track of its current security level.

One can simplify the implementation by migrating all the processes in a system to the same level, namely the highest level one of the processes has. Having such a global level for the whole system is less complicated to implement, and possibly also easier for the user to understand, but note that it enforces even stronger restrictions that strictly needed for Bell-Lapadula.

Write downs effectively become impossible in such a system: after information of level X has been read, the whole system can only ever be in a global level Z which is at least that high, and all the files written will then immediately get level Z.

High Watermark is a Bell-LaPadula variant whereby the security level of an object changes to the user's highest security level currently open. This means that an unclassified news-item will be upgraded to confidential, if a user A accessing it is also editing a confidential document. If a user B is however editing a secret intelligence report and at the same time accesses an unclassified news-item, this news-item will become secret. It is clear that undesired side-effects show up when implementing High Watermark.

System High is a security mode where each user with direct or indirect access to the information system, has a valid clearance for all information contained within this system. An information system in System High mode usually has a fixed classification-level. Information leaving the system therefore has to be either formally declassified or treated according to the rules belonging to the system's security level. An example of a system implementing System High mode is Titaan, the Netherlands Army and Air-force operational network. Titaan is certified to contain information up to secret level. Access control is being regulated by rules implementing DAC. Titaan is currently being used in southern Afghanistan by Netherlands and Australian troops.

Combining Systems

In a typical defence working environment, several securtity-levels exist. When combining them, various approaches are possible. Although full integration would be the most desirable option, intermediate solutions are also possible. If systems are not physically connected, they can coexist in a system called MILS.

MILS is an approach where multiple computer networks are deployed, each dedicated to a specific security level. These networks are physically separated according to Tempest (Emission Security or Emanations Security - EMSEC) rules. Data can only be copied from one network onto a higher classified network using some sort of removable storage device. Although this might seem as a workable solution, other new risks are introduced by placing classified data on removable media, such as loss or theft of USB-sticks.

A step closer towards MLS is an approach where several security levels share physical means, but are logically separated by using encryption and tunnelling.

MSL (Multiple Single-Levels) is a method where several security levels form a MLS system. Different levels of security are separated by using separate computers or virtual machines for each level of security. It's main advantage is that it gives some of the benefits of Multilevel Security, without adapting the OS or individual applications. MSL is however more expensive in terms of extra hardware needed than true MLS, but less expensive compared to MILS.

What Went Wrong

[1] describes technical as well as political and economical issues which have led tot the current state of MLS systems. The large effort spend on developing and building classic MLS systems has not only led to several failed systems, it has clarified many consequences of information flow controls.

The changed political en economical situation over the past two decades has led to tremendous budget-cuts whilst the total of all classification actions reported has increased by more than 60% [1]. As a result governments have been drawn towards low-security solutions in its most critical and most targeted components.

According to Bell himself [4, 3], the marketplace has never produced high security products spontaneously and government versions of MLS products have never been viable over the long term. To overcome this, he calls for a government-induced nurturing environment consisting of security standards, evaluation of secure products against those standards, policy mandating their use, and evidence that the policy will be enforced.

Conclusions

MLS (Multi-Level Security) is typically taken to comprise a system with multiple levels of security with Mandatory Access Control enforcing the Bell-LaPaluda rules. However, most MLS systems used in the defence environment, e.g. in Afghanistan or Iraq, implement simpler, but more restictive forms of this, namely System High or MILS.

Moreover, such systems are typically used used side by side, with some (often manual) form of communication between them, which effectively allows some form of de-, or more generally re-classification. Considering the combination of several of these systems, it is less restrictive than classic MLS, as it implements MLS with the possibility of some form of re-classification.

A further complication in the search for true MLS are covert channels [7], induced by meta- and history information that can be included in formats of data, e.g. Microsoft Word documents including information about edits (incl. deletion) made in the past and about the persons who performed this [6].

Although a lot of knowledge about security policies and MLS systems is available and the call for high-security MLS-like systems has increased, we have not managed to implement true MLS in $3\frac{1}{2}$ decades. The question therefore rises whether the direction of the research into MLS has to be amended. One thing however is clear: MLS products are unavoidable, now and in future.

List of Abbreviations

IDO	=	International Defence Organisation
MILS	=	Multiple Independent Levels of Security
MLS	=	Multi-Level Security
MSL	=	Multiple Single-Levels
NEC	=	Network Enabled Capabilities
Titaan	=	Theatre Independent Tactical Army an Air-force Network

References

[1] Ross J. Anderson, Frank Stajano, and Jong-Hyeon Lee. Security policies. *Advances in Computers*, 55:186–237, 2001. Available from World Wide Web: `http://www.cl.cam.ac.uk/~rja14/Papers/security-policies.pdf`.

[2] D. Bell and L. LaPadula. Secure computer system: Unified exposition and multics interpretation. Technical Report MTR-2997, MITRE Corp., Bedford, MA, July 1976. Available from World Wide Web: `http://csrc.nist.gov/publications/history/bell76.pdf`.

[3] David Elliott Bell. Looking back: Addendum, 2005. Available from World Wide Web: `http://www.selfless-security.org/`.

[4] David Elliott Bell. Looking back at the bell-la padula model. In *ACSAC*, pages 337–351. IEEE Computer Society, 2005. Available from World Wide Web: `http://doi.ieeecomputersociety.org/10.1109/CSAC.2005.37`.

[5] K. Biba. Integrity considerations for secure computer systems. Technical Report TR-3153, Mitre, Bedford, MA, April 1977.

[6] Simon Byers. Information leakage caused by hidden data in published documents. *IEEE Security & Privacy*, 2(2):23–27, 2004. Available from World Wide Web: `http://doi.ieeecomputersociety.org/10.1109/MSECP.2004.1281241`.

[7] B. W. Lampson. A note on the confinement problem. *ACM*, 16(10):613–615, October 1973. Available from World Wide Web: `http://portal.acm.org/citation.cfm?id=362389`.

[8] Ravi S. Sandhu. Lattice-based access control models. *Computer*, 26(11):9–19, November 1993. Available from World Wide Web: `http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/mags/co/&toc=comp/mags/co/1993/11/rytoc.xml&DOI=10.1109/2.241422`.