MASTER per l'impresa e la Pubblica Amministrazione in:
Sicurezza dei Sistemi e delle Reti Informatiche
Gestione della Sicurezza Informatica

UNIVERSITÀ DEGLI STUDI DI ROMA
"LA SAPIENZA"

Rapporto Tecnico n. 6/2006

Atti del Primo workshop italiano su PRIvacy e SEcurity

– PRISE 2006 –

Mercoledì 21 giugno 2006

Sheraton Roma Hotel
Viale Del Pattinaggio, 100
Roma 00144 Italy

Dipartimento di Informatica
Via Salaria, 113 – 00198 Roma, Italy

Clusit
Associazione Italiana per la Sicurezza Informatica

Prefazione

La sicurezza dei dati e delle reti in funzione del suo impatto sul sistema Paese, con particolare riferimento all'economia e alla sicurezza dei cittadini, è oramai diventata un tema centrale nel contesto della moderna Società dell'Informazione e della Comunicazione. In questo quadro si sono moltiplicate in tutto il mondo le iniziative mirate a stimolare attività di ricerca, sviluppo e innovazione nel campo della sicurezza informatica. Gli attori coinvolti in queste iniziative non sono solo le Accademie e gli istituti di ricerca ma anche soggetti privati e pubbliche amministrazioni interessate alla realizzazione di dispositivi e applicazioni che oltre ad innovare i processi produttivi tengano conto dei necessari requisiti di sicurezza.

Anche nel nostro Paese, nel corso degli ultimi anni abbiamo assistito al moltiplicarsi di iniziative, tra le più disparate, nel settore. Diversi gruppi di ricerca hanno iniziato ad operare su temi specifici del settore, sono stati avviati Master Universitari sul tema, Corsi di Laurea e numerose realtà aziendali sono impegnate in progetti di ricerca su tematiche centrali o molto contigue a quelle della sicurezza informatica.

PRISE 2006 è il primo Workshop Italiano di Privacy and Security. Aperto a ricercatori, esperti dal mondo della pubblica amministrazione e dell'industria, è patrocinato dal Master in sicurezza dei sistemi e delle reti dell'Università di Roma "La Sapienza" e dal Clusit: Associazione Italiana per la Sicurezza Informatica.

Grazie ai numerosi articoli sottomessi, è stato possibile stilare un programma dei lavori che copre le principali tematiche di ricerca della sicurezza dell'informazione. In particolare, gli argomenti ricoperti includono:

Analisi di codice maligno e nuove forme di attacco - Analisi di protocolli crittografici - Autenticazione e autorizzazione - Certificazione della sicurezza di sistemi informativi – Certificati digitali - Controllo degli accessi - Informatica forense - Intrusion detection systems - Metodi per il mantenimento dell'anonimato - Privacy-enhancing technology - Sicurezza dei dati e delle reti - Sicurezza in ambienti mobili - Sviluppo di Software Sicuro

Non possiamo che essere contenti della accoglienza positiva di PRISE 2006 da parte della comunità italiana, per questo il nostro ringraziamento più sincero va a tutti coloro che vi hanno dedicato tempo e lavoro prezioso.

Workshop chairs

Prof. Danilo Bruschi - Università di Milano
Prof. Luigi V. Mancini - Università di Roma "La Sapienza"

MASTER per l'impresa e la Pubblica Amministrazione in:
Sicurezza dei Sistemi e delle Reti Informatiche
Gestione della Sicurezza Informatica

Primo workshop italiano su PRIvacy e SEcurity - PRISE

Sheraton Roma Hotel
Viale Del Pattinaggio, 100
Roma 00144 Italy
Mercoledì 21 giugno 2006

1. *"Control-flow graph matching in malware recognition"*
Danilo Bruschi, Lorenzo Martignoni, Mattia Monga - Università degli Studi di Milano

2. "S-VPN Policy: Access List Intra-Device Conflict Automatic Analysis and Resolution"
Simone Ferraresi, Stefano Pesic, Livia Trazza, Andrea Baiocchi – Elsag e Università di Roma "La Sapienza"

3. "Unsupervised Learning Algorithms for Intrusion Detection"
Giuseppe Serazzi, Stefano Zanero - DEI, Politecnico di Milano

4. "Worm Detection Using E-mail Data Mining"
Maurizio Aiello, David Avanzini, Davide Chiarella, Gianluca Papaleo - CNR-IEIIT Genova

5. *"Utilizzo di Advanced Forensic Format nell'Informatica forense"*
Stefano Fratepietro, Cesare Maioli - Cirsfid, Università di Bologna

6. "*Privacy and Data Mining: the GeoPKDD Approach"*
Francesco Bonchi, Fosca Giannotti, Dino Pedreschi, Franco Turini - CNR-ISTI, Pisa

7. *"Preserving k-anonymity in spatio-temporal datasets and location-based services*
Claudio Bettini, Sergio Mascetti - DICO, Università di Milano

8. "*A Mathematical Framework to Assess the Security of an Information Infrastructure"*
F. Baiardi, S. Suin, Claudio Telmon - Dipartimento di Informatica, Università di Pisa

9. *"SockMi: How to migrate SSL sessions"*
Massimo Bernaschi, Luigi V. Mancini, Paolo Tassotti - CNR-IAC e Università La Sapienza di Roma

10. *"Common Criteria Security Certification of Complex, "Network Centric" ICT Sytstems"*
Marco Lisi, Gaetano Tassone - Telespazio SPA, Roma

11. *"StemCerts: customizable X.509 v3 certificates for higher security, flexibility, and convenience"*
Giovanni Chiola, Paolo Gasti - DISI, Università di Genova

infosecurity ITALIA
Clusit Associazione Italiana per la Sicurezza Informatica

extended abstract

Control-flow graph matching in malware recognition

Danilo Bruschi Lorenzo Martignoni
Mattia Monga
Dip. di Informatica e Comunicazione
Università degli Studi di Milano
Via Comelico 39, 20135 Milano
{bruschi,martign,monga}@dico.unimi.it

Popular malware detectors look for their target by pattern matching. Detectors maintain a database of distinctive patterns (*signatures*) of malicious pieces of code and possibly infected systems are searched for them. This approach is fast and, up to now, quite effective when it is used to find known viruses.

However, this kind of defence will probably be circumvented by the next generation malicious code which will intensively make use of metamorphic approaches. Though this type of malware is not yet appeared in the wild, their feasibility and efficacy has been shown by some prototypes [6] (see for example METAPHOR [1], ZMIST [5], EVOL). Moreover, commercial virus scanners can be circumvented by simple mutation techniques [4, 3]. Detecting malware passed through arbitrary human-driven mutations is doomed to failure, since it is a problem as hard as program equivalence, an undecidable problem. However, virus mutations have to be automatic: they normally range ranging from trivial modifications (e.g. useless instructions insertion, and registers swapping) to the complete mutation of the payload by cryptography. One of the most advanced prototype is the ZMIST virus, which uses a metamorphic engine to change the static structure of the virus payload and inserts itself into an executable code by scattering its body among benign instructions. Malicious fragments are then connected together using appropriate control flow transition instructions. Then, the malicious code will be executed when the normal control flow reaches its first instruction: this technique is known as Entry Point Obfuscation [2].

Threats such as those represented by the ZMIST virus, poses three serious challenges to malware detectors:

- the ability to recognize self-mutating code;
- the ability to recognize malware which is randomly spread in the original code;
- the ability to recognize code which does not modify neither the behavior nor the properties of the infected program.

Note also that in order to be effective a malware detector has to be able to solve the above challenges simultaneously. The only viable way for dealing with such a kind of threat is the construction of detectors which are able to recognize malware by analyzing its *dynamic behavior* rather by scanning its text (e.g. by looking for fixed byte sequences or anomalies in the executable header). In order to cope with the above problems we propose we following strategy: given an executable program P we disassemble it in P'. P' is normalized in order to obtain a canonical form P_N in which most of the mutations are undone. We then build the corresponding labelled inter-procedural control flow graph of P_N, i.e. CFG_{P_N}, which will be compared against the control flow graph of a normalized malware CFG_M in order to verify whether CFG_{P_N} contains a subgraph isomorphic to CFG_M. The problem of detecting a malware inside an executable is therefore reduced to the subgraph isomorphism problem. Our strategy proved to be able to defeat most of the mutations techniques of polymorphic malware and code scattering. with encouraging experimental results

References

[1] MetaPHOR. `http://securityresponse.symantec.com/avcenter/venc/data/w32.simile.html`.

[2] Computer Associates. Security advisor center glossary. `http://www3.ca.com/securityadvisor/glossary.aspx`.

[3] Mihai Christodorescu and Somesh Jha. Static analysis of executables to detect malicious patterns. In *Proceedings of USENIX Security Symposium*, August 2003.

[4] Mihai Christodorescu and Somesh Jha. Testing malware detectors. In *Proceedings of the 2004 ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2004)*, pages 34–44, Boston, MA, USA, July 2004. ACM Press.

[5] Peter Ferrie and Peter Ször. Zmist opportunities. *Virus Bullettin*, 2001.

[6] P. Ször and P. Ferrie. Hunting for metamorphic. In *Proceedings of Virus Bulletin Conference*, September 2001.

S-VPN Policy: Access List Intra-Device Conflict Automatic Analysis and Resolution

Simone Ferraresi*°, Stefano Pesic*, Livia Trazza*, Andrea Baiocchi°

*Elsag S.p.A. - Via Guattani, 1 - 00161 Rome, Italy
e-mail: {simone.ferraresi, stefano.pesic, livia.trazza}@elsag.it

°INFOCOM Dept.- University of Rome "La Sapienza" - Via Eudossiana, 18 - 00184 Rome, Italy
e-mail: {andrea.baiocchi, simone.ferraresi}@uniroma1.it

I. INTRODUCTION

A key feature of secure systems, including network ones, is the management of security policies, from those at high level down to the platform specific implementation. Security policy defines constraints, limitations and authorisation on data handling and communications. In a network environment many problems may also arise due to the inconsistency of policies implemented by security gateways and firewalls interconnected over an insecure network. As distributed secure systems increase in complexity, policy configuration and maintenance tasks become increasingly prone to errors. These errors can give rise to holes in security in the entire system, that are difficult to detect before an attack is suffered. For this reason a verification phase should be performed when policy is defined and after any modification or integration.

In a single device environment, the local policy may include intra-device anomalies, where the same packet may match more than one filtering rule. Moreover, in distributed architecture, security gateway might also have inter-device anomalies when individual elements in the same path apply different filters and actions on the same traffic. Therefore, the administrator must give special attention not only to all rule relations in the same device in order to determine the correct rule order, but also to all relations between rules in different device in order to determine the proper rule placement in the proper device. As the number of filtering rules increases, the difficulty of adding a new rule or modifying an existing one significantly increases. This work gives bases for the development of a complete system for the policy conflict detection and resolution starting from the analysis of a single device environment.

We assume that policies are formally stated according to a well defined formal language, so that the access lists of a security gateway can be reduced to an ordered list of predicates of the form C -> A, where C is a condition and A is an action. We refer to predicates implementing security policies as rules. For security gateway the condition of a filtering rule is composed of five selectors:

<protocol><src ip><src port><dst ip><dst port>

The action that could be performed on the packet is allow, deny or process, where process imply that the packet has to be submitted to the IPSec algorithm (ESP, AH). How to process that packet is described in a specific MAP which details how to apply the security mechanism. Conditions are checked on each packet flowing through the device, applying it exclusively the action required by the first matching rule.

In general a conflict occurs when desired effects of a policy are ambiguous or not clear. In this paper we focus our attention on automatic conflict detection and resolution on a policy implemented in a single security gateway.

Policy consistency has been the subject of a lot of attention from the research community. The most significant contributions to this subject are [1][2][3][4][5]. In [5] the authors only attempt to detect if firewall rules are correlated to each other, while in [1][2][3] a set of techniques and algorithms are defined to discover all of the possible policy conflicts, with a particular focus on S-VPN policy described on [4]. However, none of these studies provided a resolution phase. Original contributions of our work are the definition of a general methodology for policy conflict classification and detection and exploitation of this methodology as a basis to define automatic conflict resolution algorithms. The output of such algorithms is a corrected policy file plus possibly feedback to the security manager to fix detected conflicts that cannot be decided automatically.

This work is organized as follows. In Section II we briefly present logical relationships between rules as in [1][4]. In Section III we present our formalization of severity classification for conflicts. In Section IV an algorithm is defined for automatic conflict resolution. In Section V, we give a summary of the software implementation. Finally, in Section VI, we give our conclusions and plans for future work.

II. MODELING OF RULES RELATION

To be able to introduce S-VPN policy conflict analysis and resolution it is useful to define all the relations that may tie a couple of rules. These relations, as defined in [1][4], require a comparison between the network fields of filtering rules independently of the rule action. First we state formally what a rule is in our context.

Security gateway policies as well as firewall policies specify a filtering condition C and an action A. The condition C aims at selecting those IP packets that the corresponding action applies to. Packet filtering is based on the values of five header fields: IP destination and source addresses (dst ip, src ip), destination and source port numbers (dst port, src port) and the IP header protocol type (prot). For each of these selectors we need to specify a value or a range of values. To each predicate listed in the policy file we associate a rule R defined as a five-tuple of selector variables ranges:

$$R = \{S_{prot}, S_{src\_ip}, S_{src\_port}, S_{dst\_ip}, S_{dst\_port}\}$$

where $S_{sel} = ANY$ or $S_{sel} = [sel_{\min}, sel_{\max}]$ for $sel \in \{prot, src\_ip, src\_port, dst\_ip, dst\_port\}$,

ANY being a "don't care" flag; equivalently we could replace ANY by the entire possible range of the corresponding selector. In the following we refer to the five selector with an integer values variable i = 1, · · · , 5; we denote the i-th selector range associated to a given rule R by $R[i]$. We also define a set operator as $\Diamond \in \{\subset, \supset, =\}$; also $A \Diamond B$ means that the set A is not a superset nor a subset nor the same as the set B.

We can now state formally the basic relations among rules for the conflict analysis.

Definition 1: Rules R_x and R_y are Completely Disjoint if every field in R_x is not a subset of, nor a superset of, nor equal to the corresponding field in R_y. Example:

	prot	src.ip	src.port	dst.ip	dst.port
R_x	TCP	10.33.0.0/24	1003	10.1.0.0/24	80
R_y	UDP	10.45.0.0/24	1005	10.2.0.0/24	88

Definition 2: Rules R_x and R_y are Exactly Matching if every field in R_x is equal to the corresponding field in R_y. Example:

	prot	src.ip	src.port	dst.ip	dst.port
R_x	TCP	10.33.0.0/24	1003	10.1.0.0/24	80
R_y	TCP	10.33.0.0/24	1003	10.1.0.0/24	80

Definition 3: Rules R_x and R_y are Inclusively Matching if they do not exactly match and every field in R_x is a subset of or equal to the corresponding field in R_y. R_x is called the subset match, while R_y is called the superset match. Example:

	prot	src.ip	src.port	dst.ip	dst.port
R_x	TCP	10.33.0.0/16	1003	10.1.0.0/16	80
R_y	TCP	10.33.0.0/16	1003	10.1.2.0/24	80

Definition 4: Rules R_x and R_y are Partially Matching if there is at least one field in R_x that is a subset of or a superset of or equal to the corresponding field in R_y and there is at least one field in R_x that is not a subset nor a superset, nor equal to the corresponding field in R_y. Example:

	prot	src.ip	src.port	dst.ip	dst.port
R_x	TCP	10.33.0.0/16	1003	10.1.0.0/16	80
R_y	TCP	10.45.0.0/16	1003	10.1.2.0/24	80

Definition 5: Rules R_x and R_y are Correlated if some fields in R_x are subsets of or equal to the corresponding fields in R_y and the rest of the fields in R_x are supersets of the corresponding fields in R_y. Example:

	prot	src.ip	src.port	dst.ip	dst.port
R_x	TCP	10.33.2.0/24	ANY	10.1.0.0/16	80
R_y	TCP	10.33.0.0/16	ANY	10.1.2.0/24	80

III. CONFLICT ANALYSIS

To deal with the resolution of S-VPN policy conflicts we have to elaborate a new classification of them which is more selective than the one presented in [1][4]. This has been done by introducing the concept of conflict severity.

Definition 6: The severity of a conflict is defined as the rank of correlation between the presence of the conflict in the policy and the erroneous behaviour of the respective device.

A device behaviour is considered erroneous when it does not correspond to the aim of the security manager.

To univocally identify the aim of security manager it has been necessary to formulate a working hypothesis on his behaviour.

Working Hypothesis. The security manager inserts a rule in the policy because he wants to apply it to at least one packet.

S-VPN policies are formalized into rules and rules are listed in an ordered file; for each packet the list is scanned and the first matching rule is found (if any); the corresponding action is applied to the packet. Rules further down in the file are ignored for that packet.

We now introduce the classification of conflicts according to severity levels:

Exact Match: A rule is in exact match with another one when the two rules are equal in all selectors independently of the value assumed in the action field. These two rules match the same traffic, on which, however, only the action of the rule with higher priority is performed.

The Exact Match conflict is the most severe since it is impossible to understand which one of rules Rx and Ry the

security manager wanted to be performed. This is a major problem both in case the actions of the two rules are different and in case they are the same, since the rules in between them might act on packets matched by the condition of rules R_x and R_y, so that it can make a critical difference to eliminate either R_x or R_y.

Shadowing: A rule is shadowed when a previous rule, with different action, matches all the packets that this rule matches so that the shadowed rule will never be activated.

This kind of conflict is very severe because the inactivity of the rule R_y represents a violation of the aim of the security manager, consequently performing an erroneous behavior of the device. Example:

	prot	src.ip	src.port	dst.ip	dst.port	action
R_x	TCP	10.0.0.0/8	ANY	80.0.0.0/8	80	bypass
R_y	TCP	10.2.2.0/24	ANY	80.1.3.4/32	80	process

Post Redundancy: A rule is in post redundancy when a previous rule, with same action, matches all the packets that this rule matches so that the redundant rule will never be activated. Example:

	prot	src.ip	src.port	dst.ip	dst.port	action
R_x	TCP	192.0.0.0/8	ANY	10.1.0.0/16	80	process
R_y	TCP	192.168.2.0/24	ANY	10.1.0.0/16	80	process

Since the actions are the same, an erroneous behaviour of the device does not necessarily occur. Yet, this kind of conflict is severe because the inactivity of the rule R_y represents a violation of the aim of the security manager, according to our working hypothesis.

Correlation: Two rules are correlated if they have different filtering actions, and the first rule matches the same packets that the second rule matches and vice versa. Example:

	prot	src.ip	src.port	dst.ip	dst.port	action
R_x	TCP	10.3.2.0/24	ANY	80.1.0.0/16	80	discard
R_y	TCP	10.0.0.0/8	ANY	80.1.3.4/32	80	bypass

In this situation a violation of the aim of the security manager does not occur because both the rules are active, but is not possible to decide if the behaviour of the device on the traffic matching both rules is erroneous or not, because it depends on the relative order of the two rules.

Pre Redundancy: A rule is in pre redundancy with a previous rule if these have the same actions and if the second rule can match all the packets that the first rule matches. Between the two rules there should not be rules that are in relation with the redundant rule. Example:

	prot	src.ip	src.port	dst.ip	dst.port	action
R_x	TCP	10.4.3.0/24	ANY	80.6.6.0/24	80	bypass
R_y	TCP	10.0.0.0/8	ANY	80.6.6.0/24	80	bypass

This kind of conflict can be seen just as an anomaly, not a severe conflict, because both rules will process some traffic.

Generalization: A rule is a generalization of a previous rule if these have different actions and if the second rule can match all the packets that the first rule matches. Example:

	prot	src.ip	src.port	dst.ip	dst.port	action
R_x	TCP	10.2.3.2/32	ANY	80.4.4.3/32	80	bypass
R_y	TCP	10.2.0.0/8	ANY	80.4.4.0/24	80	process

This kind of conflict is also not severe because both rules will process some traffic performing different actions. This case has to be labelled as a normal and desirable situation.

No Conflict: When two rules do not fall under any of the previous categories there is a state of No conflict.

To perform the comparison with the selectors of the rules necessary for the detection of conflicts the state diagram presented in [1] has been changed. The modified state machine transition diagram is shown in Figure 1, with some simplification for space reasons.

IV. CONFLICT RESOLUTION

S-VPN or FW policy has to be declared incorrect if a progressive rules analysis points out at least one of the severe conflicts as Exact Match, Shadowing and Post Redundancy. Therefore it can be judged correct if the analysis discovers that the policy does not have any severe conflicts.

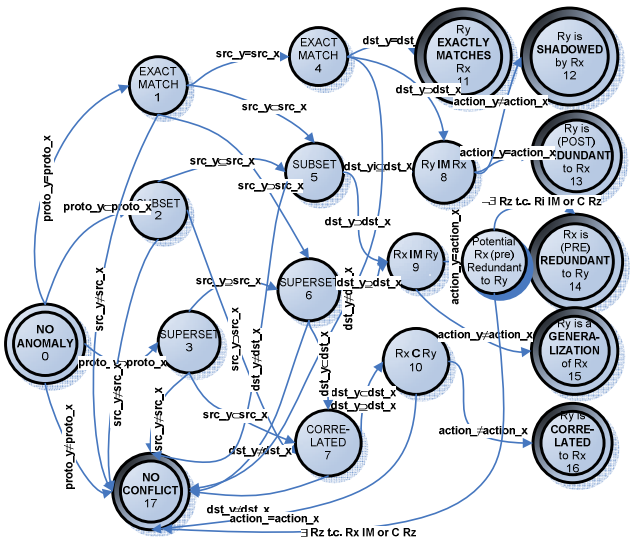

Figure 1

Conflict resolution makes changes in S-VPN policy so that it becomes free from the severe conflicts, without losing any of the security services required by security manager.

Inclusive Match Ordered Algorithm

This Inclusive Match Ordered (IMO) algorithm ensures that, through the elimination of some rules, and modifications of the priority value of the remaining rules, the list of all S-VPN policy converges to a final state without severe conflicts.

Let B be any set consisting of n distinct rules, subset of set A which includes all the rules. B's elements can be assumed distinct provided to have taken care of all the rules in Exact Match relation as described in the previous algorithm.

Any pair of rules arising a severe conflict other than Exact Match, i.e. Shadowing or Post Redundancy, are in $\Re_{IM}$ relation, so that the most specific rule has a higher priority than the other rule. Thus, the research of a severe-conflictfree policy list state can be reduced to a problem of partial ordering of the rules related by $\Re_{IM}$.

Algorithm description: The IMO algorithm is composed of four steps:

1) Delete Exact Match conflicts asking the user on the processing he wants to apply to the examined traffic (or by a conservative philosophy automatically deleting the less restrictive rule).
2) Scan the policy list and solve any found conflict $R_x C_{SH} R_y$ or $R_x C_{PO} R_y$ by moving R_y immediately before R_x which was hiding it.
3) Repeat step 2) until reaching the state of absence of Shadowing and PostRedundancy conflicts.
4) Compare the output of the previous two steps with the initial policy list and notify the security manager with the possible presence of Correlation conflicts so that the respective rule order is modified.

The existence of a conflict-free state and the convergence of the research algorithm on this final state within a finite number of steps, are both formally proved.

Incidentally, the last equality of the proof process poses an upper bound on the computational complexity of IMO algorithm which is,

$$\sum_{i=2}^{n} (i-1) = \frac{(n-1)\cdot n}{2} .$$

V. SOFTWARE IMPLEMENTATION

Analysis and resolution methodologies have been implemented in a software tool developed in C#, the object oriented Microsoft programming language. The tool is named PETRA.

Development phase has been realized in the now freeware environment MS Visual Studio 2005 Express Edition based on the framework .NET 2.0.

VI. CONCLUSIONS AND FUTURE WORKS

One of the most critical aspects of security problems is the impossibility of accurately checking a system real weakness.

In a complex and distributed environment this problem is greatly accentuated. During the process of configuration and implementation of the network security policies errors can occur, resulting in holes in security and, consequently, compromising the entire system functionality. These errors are often very hard to detect by performing a manual or visual inspection. For this reason, automatic management of this phase is required.

We define a framework and algorithms to alleviate this problem for a single S-VPN or Firewall device, by means of automated policy conflict identification and resolution. In this way, the security manager is supplied with an automatic tool that can detect, locate and solve conflicts that may occur within an S-VPN policy file. Based on the previous literature on this subject we have formalized a new conflict classification founded on the severity concept. Moreover we have proposed an automatic resolution algorithm, and implemented it in a software tool. Our future research plans include extending the proposed anomaly resolution techniques to handle distributed firewall policies, and the interaction between S-VPN policy and FW policy. Our approach will base on applying the detection methodology on rules obtained merging the rule set of two access lists in a distributed environment. The conflicts severity classification will be applied to the distributed environment and new detection and resolution algorithms must be defined. The detection methodology will be modified because of the absence of priority relation between rules applied on different devices. Thus the resolution algorithm won't be based on the modifications of the priority value of some rules, but on creation or elimination of rules according to the new classification.

REFERENCES

[1] E. Al Shaer and H. Hamed, "Modeling and Management of Firewall Policies", in *IEEE eTransactions on Network and Service Management*, Volume 1-1, April 2004.

[2] E. Al Shaer, H. Hamed, R. Boutaba, M. Hasan, "Conflict Classification and Analysis of Distributed Firewall Policies", in *IEEE Journal on Selected Areas in Communications*,vol.23, no.10, October 2005.

[3] E. Al Shaer and H. Hamed, "Firewall Policy Advisor for Anomaly Detection and Rule Editing", in *Proceedings of IEEE/IFIP Integrated Management Conference (IM2003),*March 2003.

[4] E. Al Shaer, H. Hamed, W. Marrero "Modeling and Verification of IPSec and VPN Security Policies", *Proceedings of IEEE ICNP'2005*, November 2005.

[5] HB. Hari, S. Suri and G. Parulkar, "Detecting and Resolving Packet Filter Conflicts", *Proceedings of IEEE INFOCOM 2000,* March 2000.

[6] M. Gouda and X. Liu, "Firewall Design: Consistency, Completeness, and Compactness" *Proceedings of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS'04),* March 2004.

[7] S. Ioannidis, A. Keromytis, S. Bellovin and J. Smith, "Implementing a Distributed Firewall" *Proceedings of 7th ACM Conference on Computer and Comminications Security (CCS'00),* November 2000.

[8] W. Cheswick and S. Bellovin, "Firewalls and Internet Security", *Addison-Wesley,* 1995.

Unsupervised Learning Algorithms for Intrusion Detection

Giuseppe Serazzi, Stefano Zanero
{serazzi,zanero}@elet.polimi.it
Dip. di Elettronica e Informazione, Politecnico di Milano

1 Introduction

Our research work focuses on the analysis and development of anomaly based intrusion detection systems based on *unsupervised learning algorithms*.

Unsupervised learning algorithms are natural candidates for the task of detecting anomalous and intrusive behavior in computer systems and networks, for a number of reasons:

Outlier detection: unsupervised learning techniques are capable of identifying "strange" observations in a wide range of phenomena; this is a characteristic we definitely need in an anomaly based IDS.

Generalization: unsupervised learning techniques are also quite robust and therefore can show better resistance to polymorphic attacks.

Unsupervised learning: we wanted to create a model totally orthogonal to the misuse based model, which is dependent on the input of expert knowledge, so we tried to develop an IDS which needed no a priori knowledge inputs.

Adaptation: a learning algorithm can be tuned totally to the specific network or system it operates into, which is also an important feature to reduce the number of false positives and optimize the detection rate.

2 Original research contributions

Our key original contributions in this area of research (which we have published in international conferences [1–4], and which have been the core of a doctoral thesis [5]) can be identified as follows.

Network Intrusion Detection is a particularly difficult field for the application of unsupervised learning algorithms. In particular, the varying size of the payloads of the datagrams, and their heterogeneous nature which defies a compact representation as a single feature, are the hardest problems to solve. Most existing researches on this topic avoid this problem altogether by discarding the payload and retaining only the information in the packet header, or by tracking connection-wide variables instead of analyzing single packets.

This, however, inevitably leads to information loss: most attacks, in fact, are detectable only by analyzing the payload of a packet, not the headers alone.

We proposed instead in [4] a two-tier architecture for a network based anomaly detection system capable of handling also the content of the payload of network packets.

In the first tier of the system, a Self Organizing Map (SOM) operates a basic form of pattern recognition on the payload of the packets, observing one packet payload at a time and "compressing" it into a byte of information (a "payload class" value). This classification is then added to a subset of the information decoded from the packet header and passed on to the second tier algorithm, which is an anomaly detector (Smart Sifter) capable of detecting outliers in multivariate time series.

We evaluated different algorithms for both tiers, and reported on our results [2]. We considered performance issues and proposed improvements and heuristics to increase the throughput of SOMs by almost three times, with marginal misclassification rates, to reach a speed which is suitable for online Intrusion Detection purposes [3].

SmartSifter [6] is an unsupervised algorithm for outlier detection in multivariate time series based on discounting learning. It is designed for online usage, and it uses a "forgetting factor" in order to adapt the model to non-stationary data sources. The output of SmartSifter is a value expressing the statistical distance of the new observation from the former ones. In order to automatically tune the threshold beyond which a data vector is considered an outlier, we modified SmartSifter by introducing a training phase during which the distribution of the anomaly scores is approximated, and an estimated quantile of the distribution is also computed. In this way we can directly set the IDS sensitivity as the percentage of packets we want to consider as outliers.

In order to evaluate how well the proposed system performs, we can compare it against two comparable state-of-the-art systems. The authors of SmartSifter claim a 18% detection rate, with a 0.9% false positive rate. Our algorithm can instead reach a 92% detection rate with a 0.17% false positive rate, thus demonstrating a highly superior performance.

PAYL [7] is a prototype which uses part of the payload of packets: in fact, it is the only instance in literature, besides our own work, where such a concept is applied. The best overall results for PAYL show a detection rate of 58.7%, with a false positive rate that is between 0.1% and 1%. Our architecture can reach the same detection rate with a false positive rate below 0.03%, thus an order of magnitude better than PAYL, or on the other hand it can reach a 88.9% detection rate with no more than a 1% rate of false positives.

3 Future work perspectives

We are now striving to improve the speed of the network based IDS system to make it suitable for Gigabit speed detection, as well as working to reduce the false positive rate as much as possible.

In this direction, we feel that an interesting perspective is the integration of the network based system with a novel host based systems we designed [5], in order to use the results of both to automatically filter out false positives and to improve correlation and alert quality. We are also trying to allow a human expert to refine the training of the system, with a "semi-supervised" approach. Additionally, we need to enhance the amount of information a human operator

can get from the system, and to make it more user friendly and actionable.

Evaluation of an intrusion detection system is a difficult and open research topic [8]. It is very difficult to plan tests for the different performance metrics of an IDS system (such as throughput, detection capabilities, etc.), and it is even more difficult to combine these tests in a meaningful, overall evaluation. Most evaluations therefore consist of a simple run of the algorithms over a dataset containing background activities as well as attacks.

For privacy reasons, it is very difficult to gather the full payload traces of real networks. In addition, IDS researchers need clearly labeled data where attacks are described in full details, something which is usually impossible to achieve with real-world dumps. The only such dataset is the so-called "DARPA IDS Evaluation dataset", collected between 1998 and 1999 in order to evaluate detection rates and false positives rates of IDS.

Other datasets exist, but they are not labeled and do not contain "background traffic". Thus, most existing researches on network based IDSs use the DARPA datasets for evaluation. This is a crucial factor: any bias or error in the DARPA dataset has influenced, and will influence in the future, the very basic research on this topic.

Both the background traffic and the attack traffic are artificially generated specifically for IDS evaluation. In [9] there is a detailed analysis of the shortcomings of the 1999 traffic sample set. In particular, the author notes that no detail is available on the generation methods, that there is no evidence that the traffic is actually realistic, and that spurious packets, so common on the Internet today, are not taken into account. The same can be said for checksum errors, fragmented packets, and similars. The simulated network is flat, and therefore unrealistic. In [10] additional strange characteristics of the synthetic packets are detected, and the authors even propose a simple IDS system based on a single byte of the IP header (the third byte of the IP address, in particular), which achieves a 45% Detection Rate with just a bunch of false positives.

In [5] we also make it evident that the host based traces suffer from similar issues. The first problem is that there are too few execution instances for each software. The number of system calls used is also extremely limited, making execution flows very similars. Additionally, most of these executions are similars, not covering the full range of possible execution paths of the programs (thus causing overfitting of any anomaly model). The arguments show the same lack of variability.

Furthermore, since the last dataset in the IDEVAL series was created in 1999, attacks and programs are hopelessly outdated by now. Since most attacks in the dataset are buffer overflows, we were able to create a detector which finds all the attacks in the host based datasets without any false positive.

Therefore we think it is high time to study and create a more sound methodology for evaluating and testing intrusion detection systems. We are designing a toolset for generating synthetic traffic and superimposing attacks, and we will try to develop a methodology for evaluation which is both scientifically repeatable and sound with respect to real world usage requirements.

References

[1] Stefano Zanero. Behavioral intrusion detection. In Cevdet Aykanat, Tugrul Dayar, and Ibrahim Korpeoglu, editors, *Proceedings of ISCIS 2004*, volume 3280 of *Lecture Notes in Computer Science*, pages 657–666, Kemer-Antalya, Turkey, October 2004. Springer.

[2] Stefano Zanero. Analyzing tcp traffic patterns using self organizing maps. volume 3617 of *Lecture Notes in Computer Science*, pages 83–90, Cagliari, Italy, September 2005. Springer.

[3] S. Zanero. Improving self organizing map performance for network intrusion detection. In *SDM 2005 Workshop on "Clustering High Dimensional Data and its Applications"*, 2005.

[4] Stefano Zanero and Sergio M. Savaresi. Unsupervised learning techniques for an intrusion detection system. In *Proc. of the 2004 ACM Symposium on Applied Computing*, pages 412–419. ACM Press, 2004.

[5] Stefano Zanero. *Unsupervised Learning for Intrusion Detection*. PhD thesis, Politecnico di Milano, 2006.

[6] K. Yamanishi, J.-I. Takeuchi, G. J. Williams, and P. Milne. Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Knowledge Discovery and Data Mining*, 8(3):275–300, 2004.

[7] Ke Wang and Salvatore J. Stolfo. Anomalous payload-based network intrusion detection. In *RAID Symposium*, September 2004.

[8] Stefano Zanero. My ids is better than yours... or is it ? In *Blackhat Federal 2006 Briefings*, 2006.

[9] John McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. on Information and System Security*, 3(4):262–294, 2000.

[10] M. V. Mahoney and P. K. Chan. An analysis of the 1999 DARPA / Lincoln laboratory evaluation data for network anomaly detection. In *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, pages 220–237, Pittsburgh, PA, USA, September 2003.

Research Contributions on IDS

Giuseppe Serazzi, Stefano Zanero
{serazzi,zanero}@elet.polimi.it
Dip. di Elettronica e Informazione, Politecnico di Milano

1 Introduction

Our research work focuses on the analysis and development of anomaly based intrusion detection systems based on *unsupervised learning algorithms*.

Unsupervised learning algorithms are natural candidates for the task of detecting anomalous and intrusive behavior in computer systems and networks, for a number of reasons:

Outlier detection: unsupervised learning techniques are capable of identifying "strange" observations in a wide range of phenomena; this is a characteristic we definitely need in an anomaly based IDS.

Generalization: unsupervised learning techniques are also quite robust and therefore can show better resistance to polymorphic attacks.

Unsupervised learning: we wanted to create a model totally orthogonal to the misuse based model, which is dependent on the input of expert knowledge, so we tried to develop an IDS which needed no a priori knowledge inputs.

Adaptation: a learning algorithm can be tuned totally to the specific network or system it operates into, which is also an important feature to reduce the number of false positives and optimize the detection rate.

2 Original research contributions

Our key original contributions in this area of research (which we have published in international conferences [1–4], and which have been the core of a doctoral thesis [5]) can be identified as follows.

Network Intrusion Detection is a particularly difficult field for the application of unsupervised learning algorithms. In particular, the varying size of the payloads of the datagrams, and their heterogeneous nature which defies a compact representation as a single feature, are the hardest problems to solve. Most existing researches on this topic avoid this problem altogether by discarding the payload and retaining only the information in the packet header, or by tracking connection-wide variables instead of analyzing single packets.

This, however, inevitably leads to information loss: most attacks, in fact, are detectable only by analyzing the payload of a packet, not the headers alone. We proposed instead in [4] a two-tier architecture for a network based anomaly

detection system capable of handling also the content of the payload of network packets.

In the first tier of the system, a Self Organizing Map (SOM) operates a basic form of pattern recognition on the payload of the packets, observing one packet payload at a time and "compressing" it into a byte of information (a "payload class" value). This classification is then added to a subset of the information decoded from the packet header and passed on to the second tier algorithm, which is an anomaly detector (Smart Sifter) capable of detecting outliers in multivariate time series.

We evaluated different algorithms for both tiers, and reported on our results [2]. We considered performance issues and proposed improvements and heuristics to increase the throughput of SOMs by almost three times, with marginal misclassification rates, to reach a speed which is suitable for online Intrusion Detection purposes [3].

SmartSifter [6] is an unsupervised algorithm for outlier detection in multivariate time series based on discounting learning. It is designed for online usage, and it uses a "forgetting factor" in order to adapt the model to non-stationary data sources. The output of SmartSifter is a value expressing the statistical distance of the new observation from the former ones. In order to automatically tune the threshold beyond which a data vector is considered an outlier, we modified SmartSifter by introducing a training phase during which the distribution of the anomaly scores is approximated, and an estimated quantile of the distribution is also computed. In this way we can directly set the IDS sensitivity as the percentage of packets we want to consider as outliers.

In order to evaluate how well the proposed system performs, we can compare it against two comparable state-of-the-art systems. The authors of SmartSifter claim a 18% detection rate, with a 0.9% false positive rate. Our algorithm can instead reach a 92% detection rate with a 0.17% false positive rate, thus demonstrating a highly superior performance.

PAYL [7] is a prototype which uses part of the payload of packets: in fact, it is the only instance in literature, besides our own work, where such a concept is applied. The best overall results for PAYL show a detection rate of 58.7%, with a false positive rate that is between 0.1% and 1%. Our architecture can reach the same detection rate with a false positive rate below 0.03%, thus an order of magnitude better than PAYL, or on the other hand it can reach a 88.9% detection rate with no more than a 1% rate of false positives.

3 Future work perspectives

We are now striving to improve the speed of the network based IDS system to make it suitable for Gigabit speed detection, as well as working to reduce the false positive rate as much as possible.

In this direction, we feel that an interesting perspective is the integration of the network based system with a novel host based systems we designed [5], in order to use the results of both to automatically filter out false positives and to improve correlation and alert quality. We are also trying to allow a human expert to refine the training of the system, with a "semi-supervised" approach. Additionally, we need to enhance the amount of information a human operator can get from the system, and to make it more user friendly and actionable.

Evaluation of an intrusion detection system is a difficult and open research topic [8]. It is very difficult to plan tests for the different performance metrics of an IDS system (such as throughput, detection capabilities, etc.), and it is even more difficult to combine these tests in a meaningful, overall evaluation. Most evaluations therefore consist of a simple run of the algorithms over a dataset containing background activities as well as attacks.

For privacy reasons, it is very difficult to gather the full payload traces of real networks. In addition, IDS researchers need clearly labeled data where attacks are described in full details, something which is usually impossible to achieve with real-world dumps. The only such dataset is the so-called "DARPA IDS Evaluation dataset", collected between 1998 and 1999 in order to evaluate detection rates and false positives rates of IDS.

Other datasets exist, but they are not labeled and do not contain "background traffic". Thus, most existing researches on network based IDSs use the DARPA datasets for evaluation. This is a crucial factor: any bias or error in the DARPA dataset has influenced, and will influence in the future, the very basic research on this topic.

Both the background traffic and the attack traffic are artificially generated specifically for IDS evaluation. In [9] there is a detailed analysis of the shortcomings of the 1999 traffic sample set. In particular, the author notes that no detail is available on the generation methods, that there is no evidence that the traffic is actually realistic, and that spurious packets, so common on the Internet today, are not taken into account. The same can be said for checksum errors, fragmented packets, and similars. The simulated network is flat, and therefore unrealistic. In [10] additional strange characteristics of the synthetic packets are detected, and the authors even propose a simple IDS system based on a single byte of the IP header (the third byte of the IP address, in particular), which achieves a 45% Detection Rate with just a bunch of false positives.

In [5] we also make it evident that the host based traces suffer from similar issues. The first problem is that there are too few execution instances for each software. The number of system calls used is also extremely limited, making execution flows very similars. Additionally, most of these executions are similars, not covering the full range of possible execution paths of the programs (thus causing overfitting of any anomaly model). The arguments show the same lack of variability.

Furthermore, since the last dataset in the IDEVAL series was created in 1999, attacks and programs are hopelessly outdated by now. Since most attacks in the dataset are buffer overflows, we were able to create a detector which finds all the attacks in the host based datasets without any false positive.

Therefore we think it is high time to study and create a more sound methodology for evaluating and testing intrusion detection systems. We are designing a toolset for generating synthetic traffic and superimposing attacks, and we will try to develop a methodology for evaluation which is both scientifically repeatable and sound with respect to real world usage requirements.

References

[1] Stefano Zanero. Behavioral intrusion detection. In Cevdet Aykanat, Tugrul Dayar, and Ibrahim Korpeoglu, editors, *Proceedings of ISCIS 2004*, volume

3280 of *Lecture Notes in Computer Science*, pages 657–666, Kemer-Antalya, Turkey, October 2004. Springer.

[2] Stefano Zanero. Analyzing tcp traffic patterns using self organizing maps. volume 3617 of *Lecture Notes in Computer Science*, pages 83–90, Cagliari, Italy, September 2005. Springer.

[3] S. Zanero. Improving self organizing map performance for network intrusion detection. In *SDM 2005 Workshop on "Clustering High Dimensional Data and its Applications"*, 2005.

[4] Stefano Zanero and Sergio M. Savaresi. Unsupervised learning techniques for an intrusion detection system. In *Proc. of the 2004 ACM Symposium on Applied Computing*, pages 412–419. ACM Press, 2004.

[5] Stefano Zanero. *Unsupervised Learning for Intrusion Detection*. PhD thesis, Politecnico di Milano, 2006.

[6] K. Yamanishi, J.-I. Takeuchi, G. J. Williams, and P. Milne. Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Knowledge Discovery and Data Mining*, 8(3):275–300, 2004.

[7] Ke Wang and Salvatore J. Stolfo. Anomalous payload-based network intrusion detection. In *RAID Symposium*, September 2004.

[8] Stefano Zanero. My ids is better than yours... or is it ? In *Blackhat Federal 2006 Briefings*, 2006.

[9] John McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. on Information and System Security*, 3(4):262–294, 2000.

[10] M. V. Mahoney and P. K. Chan. An analysis of the 1999 DARPA / Lincoln laboratory evaluation data for network anomaly detection. In *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, pages 220–237, Pittsburgh, PA, USA, September 2003.

Worm Detection Using E-mail Data Mining

Maurizio Aiello*, David Avanzini*, Davide Chiarella†*, Gianluca Papaleo†*

*National Research Council, Institute IEIIT, Genoa
†University of Genoa, Department of Computer and Information Sciences, Italy

***Abstract*—We propose a new technique to detect internet worm. We base our research on the fact that an indirect worm (a worm spreading by e-mail) needs to spread quickly and so it sends a lot of e-mail in a short while, producing an anomalous behaviour. Moreover we found stealthy worms through detecting traffic anomalies. We worked on a mail-server log of a real network and the results obtained drove us to detect indirect worm with different approaches based on various parameters (global e-mail flow, single host e-mail flow, reject, sender field analysis).**

***Index Terms*— Data Mining, E-mail, , Early Detection, Worm**

I. Introduction

ELECTRONIC mail has become, in the recent years, with the growth of internet one of the most used methods of communication among people, institutions and companies.

Some recent enhancement to the e-mail technology, like digital signature, drove to certified e-mail, which will substitute standard communications like registered letters. Therefore it is simple to bet that electronic messages will continue to increase in the near and far future. Due to this phenomenon, virus and worm creators choose electronic messages as a preferred way for the diffusion of their executable codes and, as a result, hosts on the net are constantly under attack by malicious programs attached to e-mails.

Worms are divided in two different kind: direct worms, which don't need a medium to propagate, because they use computer networks, exploiting operating systems bugs or weaknesses; indirect worms, which spread in an "indirect" way, using deceitful means like peer to peer file sharing or, as already said, e-mails.

Our experience in network administration leads us to analyze the e-mail traffic. In the known literature there are two main approaches to worm detection [1]: misuse detection and anomaly based. The first one is based upon the signature concept, it is more accurate but it lacks the ability to identify the presence of worms that do not fit a pre-defined signature, resulting not adaptive. The second one tries to create a model to characterize a user's normal behaviour: the system defines the expected network behaviour and, if there are significant deviations from the profile, raises an alarm. It is a more adaptive system, ready to counterattack new threats, but it has a lot of false negatives.

We choose the second one combined with an experience driven statistical method because every day new worms are created and sometimes the lag time [2] between virus generation and virus protection is a bit more than little. Another reason is that to protect hosts from a new worm it is necessary, according to misuse approach, to wait for the new signature. What we propose has no need of vendor update, so we think that this feature should be careful taken into account. First of all, in our study, we decide to analyze the overall e-mail flow of the network and in a second time the flow of a single host on the network. Analyzing the results of this first step we found that there are other features that permit to identify infected hosts.

This paper is organized as follows: Section 2 discusses previous researches on this topic, Section 3 presents our scenario and our tools, Section 4 describes our analysis and our results, Section 5 talks about practical implementations and finally in Section 6 we speak of future and on-going work.

II. Related work

In recent years, worm detection has become a very active branch of research also because the high claim from big and little business company. Both the private and the academic world have proposed a lot of ideas and solutions. In this scenario we find EMT [3] which is a data mining system that computes behaviour models of user e-mail account, based on clique theory [4], which identify groups of users which have a frequent e-mail traffic. They used a simulated viral e-mails to do their studies, approach that can mislead some results. A misuse system is Honeycomb [5], where attack signatures are automatically generated per port: they use pattern matching techniques and honeypots [6]. They rely on signature that can take time to be created but they work on a real network, a good thing from our point of view.

Don Towsley, et al. [7] with their MWC propose to search for worm epidemic pattern using Kalman filters [8] on illegitimate traffic (e.g. IP scanning). They calculate the traffic trend of growth and they compare it with an exponential epidemic model. MWC needs a lot of hosts to be effective (2^20) and it detects only direct worms (worm spreading not by e-mail and using system bugs). Moreover if a worm has a hit list (a list of real hosts to attack, got from Border Gate-way Protocol (BGP) [9] it is not detected by this approach. In literature another interesting approach in fighting worms is the Worm-killing worm or Counter-worm [10]. It doesn't detect the malware, but using a "patching worm" contrasts the speed

of infection healing the vulnerable hosts: it is clear that WKW is useful only against direct worm and this technique has a lot of legal, ethic and technical problems.

III. Our scenario

Our approach is highly experimental. In fact we work on ten local area network interconnected by a layer three switch and directly connected to Internet (no NAT policies, all public IP). In this network we have five mail-servers and one antivirus server. Since it is a research institution almost all the hosts are used by a single person and only few of them are shared among different people (students, fellow researcher etc.). We focus our attention on one mail server (Figure 1).

We analyze mail-server log of 500 days length period. To speed up the process we use LMA (Log Mail Analyzer [11]) to make the log more readable.

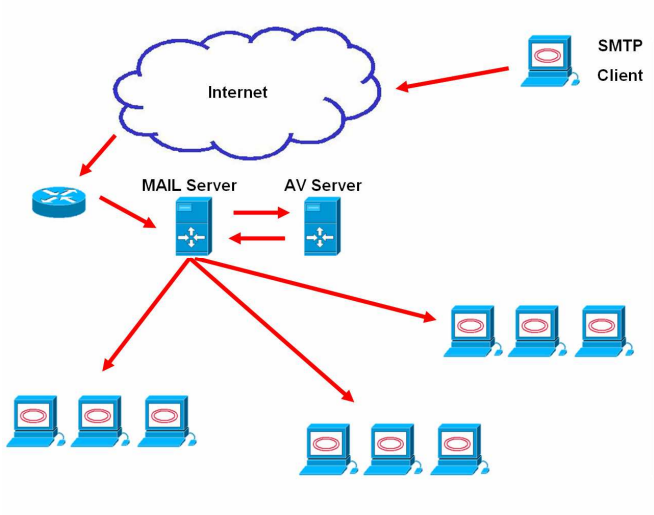

Figure 1

LMA [12] is a Perl program, open source, available on Sourceforge, which makes Postfix [13] and Sendmail [14] log human readable. It reconstructs every single e-mail transaction spread across the mail server log and it creates a plain text file in more simple format like. Every row represents a single transaction and it has the following fields:

Timestamp	It is the moment in which the e-mail has been sent: it is possible to have this information in Unix timestamp format or in julian format.
Client	It is the hostname of e-mail sender.
IP Client	It is the IP of the sender's host.
From	It is the e-mail address of the sender.
To	It is the e-mail address of the receiver.
Status	It is the server response (e.g. 450, 550 etc.).

With this format is possible to find the moment in which the e-mail has been sent, the sender client name and IP, the from and to field of the e-mail and the server response.

Lets make an example: if Paul@myisp.com send an e-mail on 23 march 2006 to Pamela@myisp.com from X.X.2.235 and all the e-mail server transactions go successful we will have a record like this:

23/03/2006 X.X.2.235 Paul@myisp.com
Pamela@myisp.com 550

After ordering the data we begin with the analysis.

IV. Analysis

Our analysis has been made on the e-mail traffic of ten C-class network in a period of 500 days, from January 2004 to April 2005.

In the first step of our analysis, we work on the global e-mail flow in a given time interval. We use a threshold detection [15], like other software do (e.g. Snort [16]): if the volume of traffic rises above a given threshold, the system triggers an alarm. The given threshold is calculated in a statistical way, where we determine the network normal e-mail traffic in selected slices of time: for example we take the activity of a month and we divide the month in five-minutes slices, calculating how many e-mails are normally sent in five minutes. After that, we check that the number of e-mails sent in a day during each interval don't exceed the threshold. We call this kind of analysis base-line analysis. Our strategy is to study the temporal correlation between the present behaviour (maybe modified by the presence of a worm activity) of a given entity (pc, entire network) and its past behaviour (normal activity, no virus or worm presence). Before proceeding, however, we preprocess the data subtracting the mean to the values and cutting all the interval with a negative number of e-mails, because we wanted to obfuscate the no-activity and few activity periods, not interesting for our purposes. In other words we trashed all the time slices characterized by a number of e-mail sent below the month average, with the purpose of dynamically selecting activity periods (working hours, no holidays etc). If we didn't perform this preprocessing we could have had an average which depended on night time, weekend or holidays duration. After this we calculate the baseline activity of working hours according to the following:

$$\text{Baseline} = \mu + 3\sigma$$

The mean and the variance are calculated for every month, modeling the network behaviour, taking into account every chosen time interval (e.g. we divide February in five-minutes slices, we count how many e-mails are sent in these periods and then we calculate the mean of these intervals). We used a similar approach counting only e-mails rejected by mail-server (550 and 450 errors). Both Global Flow e-mails and Rejected e-mails flow analysis were performed on a single host basis and on the traffic generated by the whole network. In the following sections we find the different type of analysis, which differentiate one from another by the kind of traffic considered or the information processed. In order we have Global Flow

analysis, Single IP flow analysis, From analysis, Rejected e-mails flow analysis and Single IP rejected e-mails flow analysis.

A. *Global Flow Analysis*

First kind of analysis has considered the global e-mail flow in each given period of time. Values have been compared with the baseline threshold and if found greater than it they have been marked. Analyzing the first five months with a five minutes slice we found too many alerts and a lot of them exceeded the threshold only for few e-mails. So we thought to correlate the alerts found with a five minutes period with those found with an hour period, with the hypothesis that a worm which has infected a host sends a lot of e-mail both in a short period and in a bit longer period. To clarify the concept lets take the analysis for a month: may 2004. The five minutes base-line resulted in 23 e-mails while the one hour base-line is 113.

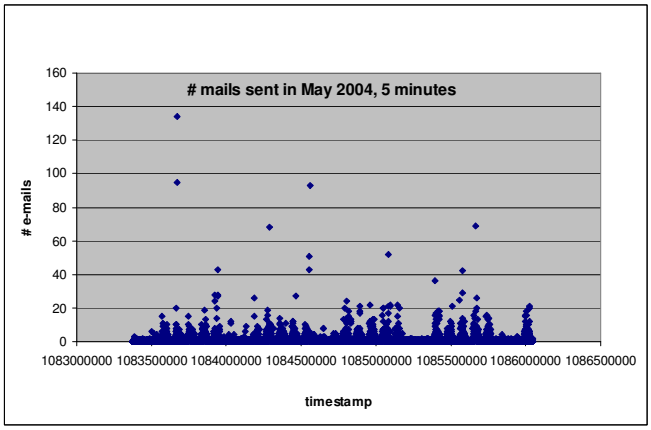

Figure 2

In five-minutes analysis we found twenty-three alerts, meanwhile in one-hour analysis only five. Why do we find a so big gap between the two approaches? In five-minutes analysis we have a lot of false alarms, due to the presence of e-mails sent to very large mailing lists while in one-hour analysis we find very few alarms, but these alarms result more significant because they represents a continuative violation of the normal (expected) activity.

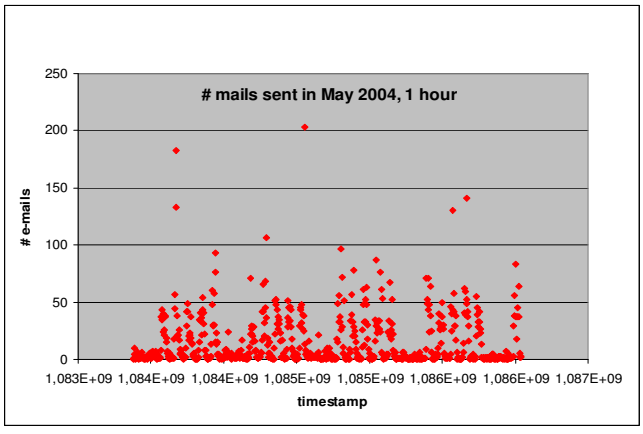

Figure 3

Correlating these results, searching the selected five-minutes periods in the five one-hour alert we detected that a little set of the five-minute alarms were near in the temporal line: after a deeper analysis, using our knowledge and experience on real user's activity we concluded that it was a worm activity.

B. *Single host flow Analysis*

Analogously we did with the traffic flow of a single host, which means all the e-mails with the same IP-Client sent in a selected period. We do this in order to find worms, which we call "stealthy" worms: very slow spreading or little activity worms. In fact the activity of this kind of worms results hidden in the global flow analysis, because they camouflage in the normal activity: the noise produced by legitimate e-mails exceeds signal produced by worm. We found out that there were two stealthy worm activities and that the remaining alerts were a subset of those worms found in the global flow.

C. *From field Analysis*

However, sometimes, peaks catch from flow analysis were e-mail sent to mailing list which are, as already said, bothersome hoaxes. This fact produced from analysis, where we analyze how many different e-mail address every host use: we look which from field is used by every host. In fact an host, owned by a single person or few persons, is not likely to use a lot of different e-mail addresses in a short time and if it does so, it is highly considerable a suspicious behaviour. So we think that this analysis could be used to identify true positives, or to suggest suspect activity. Of course it isn't so straight that a worm will change from field continuously, but it could be. Lets take a look to Figure 4 , it is almost clear that something wrong is happening in our network.

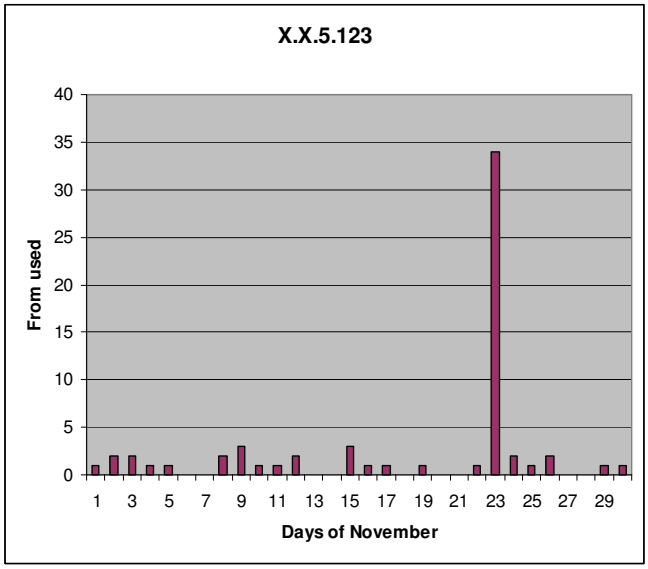

Figure 4

D. *Rejected e-mails analysis*

One typical feature of a malware is haste in spreading the infection. This haste leads indirect worms to send a lot of e-mail to unknown receivers or nonexistent e-mail address: this is a mistake that, we think, it is very important. In fact all e-mails sent to a nonexistent e-mail address are rejected by the mail-server, and they are tracked in the log.
In this step of our work we analyze rejected e-mail flow: we work only on e-mails referred by internet server. By this approach we identified various worm activity.

In all the methodologies where we considered the global traffic we acted in a similar way with single host traffic and vice versa.

V. RESULTS

The six approaches do detect various kind of worm (stealthy worms, lazy worms, hasty worms). The results we obtained are summarized in the table.

		GLOBAL FLOW			SINGLE IP		
Infection Date	Infected host	Baseline	From	Reject	Baseline	From	Reject
27/01/04	X.X.4.24	X	X			X	X
27/01/04	X.X.6.24				X	X	X
28/01/04	X.X.7.33				X	X	
29/01/04	X.X.5.201					X	X
02/02/04	X.X.5.136					X	
06/02/04	X.X.38.45						X
01/03/04	X.X.7.36					X	
22/04/04	X.X.2.126					X	
23/04/04	X.X.2.126					X	
26/04/04	X.X.88.36					X	
28/04/04	X.X.7.20	X	X	X	X	X	X
28/04/04	X.X.5.216	X	X		X	X	X
29/04/04	X.X.6.36	X	X		X	X	X
04/05/04	X.X.5.158	X	X	X	X	X	X
02/06/04	X.X.38.45			X		X	
15/06/04	X.X.5.195	X		X	X		X
16/06/2004 - 18/02/04	X.X.6.47						X
31/08/04	X.X.3.234	X		X	X		X
23/11/04	X.X.5.123	X	X		X	X	
10/09/04 – 17/09/04	X.X.9.202				X	X	X
02/11/04 – 08/11/04	X.X.3.237					X	

Figure 5

As we can see there isn't an approach which could detect all the worms, so we think that it might be a good idea to use all the approaches in a threshold system. The system can be implemented like in Figure 6.

In Figure 6 we can see that at every kind of detection is assigned a value (Global Email Flow: 1/ Global Email Flow-Reject: 0,33/ Global Email Flow-From: 0,33/Single IP Email Flow 0,5/ From 1 and Reject Email Flow 1) of alertness, if the sum of these values exceeds a given threshold an alert arises.

Future developments include a careful choice of the weight of each block composing the resolving module.

Another possibility could be to skill the system with an expert trainer (supervised learning for example) using neural network or similar methodology to achieve the best results in detection rate and low occurrences of false positives.

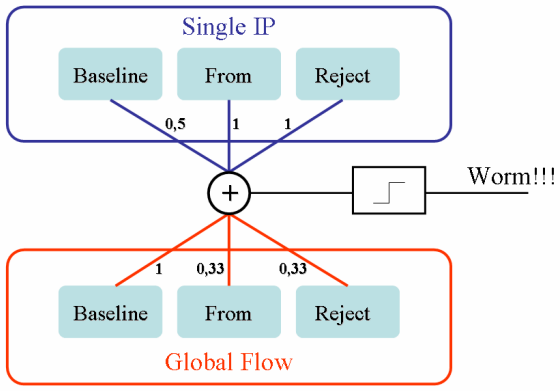

Figure 6

VI. CONCLUSION

In the near future it can be developed a tool that analyzes all the SMTP traffic directly on the cable, allowing to avoid log analysis and getting the system mail-server independent. Moreover we think to build a neural network to identify more features connected to worms activities.

VII. ACKNOWLEDGMENT

This work was supported by National Research Council of Italy, University of Genoa and PRAI-FESR Programme, Innovative Actions of Liguria.

REFERENCES

[1] S. Axelsson, \Intrusion detection systems: A survey and taxonomy," Tech. Rep. 99-15, Chalmers Univ., Mar. 2000.
[2] Preemptive Malware Protection through Outbreak Detection, Commtouch Software.
[3] Wei-Jen Li, Shlomo Hershkop, Salvatore J. Stolfo. 2004. Email Archive Analysis Through Graphical Visualization. ACM, pp. 4-5.
[4] C. Bron, and J. Kerbosch. 1973. Finding all cliques of an undirected graph. Comm. ACM 16(9), pp. 575-577.
[5] Christian Kreibich, Jon Crowcroft. Honeycomb. Creating Intrusion Detection Signatures Using Honeypots, 2003.
[6] David Dagon, Xinzhou Qin, Guofei Gu,Wenke Lee Julian Grizzard, John Levine,Henry Owen.HoneyStat: LocalWorm Detection Using Honeypots.
[7] C.C. Zou, L. Gao, W. Gong, and D. Towsley. Monitoring and Early Warning for Internet Worms. In 10th ACM Symposium on Computer and Communication Security, Washington DC, 2003.
[8] G. Welch, G. Bishop. 2004. An Introduction to the Kalman Filter.
[9] RFC 1105 - Border Gateway Protocol (BGP).
[10] On the functional validity of the worm-killing worm Hyogon Kim* and Inhye Kang,University of Seoul.
[11] http://sourceforge.net/projects/lma
[12] Log Mail Analyzer: Architecture and Practical Utilizations, Security on the Backbone: Detecting and Responding to Attacks, TNC2006.
[13] http://www.postfix.org/
[14] http://www.sendmail.org/
[15] Behaviour-Based Network Security Goes Mainstream, David Geer, Computer march 2006.
[16] http://www.snort.org/

Utilizzo di Advanced Forensic Format nell'Informatica forense

Stefano Fratepietro e Cesare Maioli
Cirsfid e Università di Bologna

Informatica forense e strumenti open di analisi

L'informatica forense è la disciplina che concerne le attività di individuazione, conservazione, protezione, estrazione, documentazione e ogni altra forma di trattamento e interpretazione del dato memorizzato su supporto informatico, al fine di essere valutato come prova nel processi giudiziari [1].

L'informatica forense ha strette connessioni con la sicurezza in quanto si ha interesse a raccogliere reperti digitali una volta che c'è l'ipotesi di un reato o di un comportamento irregolare che verosimilmente ha superato i meccanismi di controllo predisposti dagli amministratori di sistema per evitare intrusioni e alterazioni di una componente informatica.

Il prodotto leader del mercato per l'analisi forense è attualmente EnCase della Guidance Software [2] che consente di reperire, analizzare e presentare dati nell'uso professionale e investigativo da parte di numerose agenzie e forze dell'ordine in tutto il mondo e che è considerato in linea con gli standard internazionali per le analisi delle tracce informatiche. Esso utilizza un format proprietario per le immagine di dati digitali basato su ASR Data's Expert Witness Compression Format.

Il format del file Evidence File [3] contiene un bitstream fisico del disco acquisito - prefisso da un header che contiene meta informazioni sul caso in esame – intrecciato con i CRC per ciascun blocco di 64 settori (32 Kb), e seguito da un footer che contiene lo hash MD5 per l'intero bitstream. L' header contiene data e ora dell'acquisizione, il nome dell'operatore, note sulle acquisizione, una password opzionale e il proprio CRC. Il formato è comprimibile e su di esso si possono eseguire operazioni di search. La compressione si basa sui blocchi; tabelle di salto e puntatori sono mantenuti tra i blocchi e nell' header per migliorare le prestazioni. Le immagini di disco possono essere suddivise in file multipli (per esempio per memorizzare CD e DVD). I file non possono superare i due Gigabytes.

EnCase dunque memorizza l'immagine di un disco come una serie di pagine compresse univocamente individuabili e gestibili: ogni pagina può venire reperita in modo random e decompressa secondo le esigenze investigative.

EnCase consente inoltre di inserire meta informazioni sulle varie parti dei documenti sotto esame.

Sul primo punto va notato che compressori diffusi come gzip o bzip2 non consentono l'accesso random all'interno di un file compresso; sul secondo che la prassi corrente e inevitabile di inserire meta informazioni in un data base separato dal file sotto esame rende possibili smarrimenti, sovrapposizioni e disordine su informazioni spesso di interesse nei procedimenti giudiziari.

Il software EnCase è proprietario; in questi ultimi anni sono stati progettati con continuità e a forte ritmo [4] prodotti open source che presentano capacità analoghe a quelle di EnCase per la memorizzazione di copie di dati grezzi prelevati da hard disk che consentano di evitare la copia di enormi quantità di dati anche se il file in esame è di dimensioni contenute ovvero di poter accedere

selettivamente a parti di file compressi, oltre a gestire in modo efficiente meta informazioni come numeri identificativi dei drive in esame, le date, l'identificativo dell'operatore coinvolto in quella indagine e simili.

L'iniziativa Common Digital Evidence Storage Format

Il rischio che reperti e basi di prove per i procedimenti giudiziari vadano persi o divengano inammissibili in giudizio è causato dalla presenza di format diversi per le immagini digitali, tipi diversi di reperti (si va dai log di reti a memorie di dispositivi mobili), caratteristiche e comportamenti diversi degli strumenti di analisi forense ed è accresciuto dalla assenza di standard condivisi e tecnicamente robusti che consentano il congelamento della situazione rilevata, garantiscano la catena di custodia dei reperti, e siano analizzati con strumenti disponibili a tutte le parti coinvolte in un procedimento giudiziario; quest'ultimo punto suggerisce l'opportunità di soluzioni open source.

Le perdite di informazioni che si hanno convertendo dati grezzi rappresentati in formati diversi, la dimensione dei file sequestrati di cui vengono eseguite copie settore per settore, la generale mancanza di meta dati sono fattori ulteriori che complicano la situazione.

Oltre al citato EnCase sono significativi i seguenti formati di file: ProDiscover, PyFlag, RAID, SDi32, SMART della famiglia open source e ILook, SafeBack proprietari.

L'iniziativa Common Digital Evidence Storage Format [5] nasce per definire un formato open che risolva tali problemi basandosi sui format attuali, sulle esigenze dell'utenza e sugli standard giudiziari. In più la cura della catena di custodia, la cui best practice attuale sembra essere la trascrizione manuale in quadernetti o verbali delle forze investigative degli hash MD5o SHA-1 delle immagini acquisite dai supporti sotto esame, e la flessibilità per tener conto di più forme di reperto digitale (traffico in rete, dump di memorie, struttura logica dei file) suggeriscono la necessità di una *evidence bag* [6] e associata targhetta digitale in cui raccogliere tutti i reperti e le informazioni che li riguardano in maniera compatta e standardizzata come si suole in scene criminis più tradizionali.

L'adozione di un formato standard incoraggia lo sviluppo e la commercializzazione di prodotti più maturi per le analisi forensi e facilita la cooperazione tra forze investigative nazionali e internazionali.

Il format Advanced Forensic Format

Advanced Forensic Format (AFF) è una recente implementazione [7] open source ed estensibile distribuita sotto licenza BSD modificata [8] di un formato che analogamente a quello di EnCase memorizza l'immagine in maniera compressa e indirizzabile e, a differenza di quello di EnCase, consente di memorizzare le meta informazioni sia all'interno del file che in un file esterno collegato a quello di riferimento.

AFF è articolato in due layer per tener conto della compatibilità in avanti e in indietro in riferimento a un periodo temporale: il data storage layer descrive come una serie di coppie nome e valore sono memorizzate in uno o più file di disco, in maniera indipendente sia dal sistema operativo che dell'ordine dei byte; il disk representation layer definisce una serie di coppie nome e valore che vengono utilizzate per memorizzare le immagini del disco e le meta informazioni associate.

E' interessante osservare che i progettisti hanno rinunciato all'idea originale di implementazione del data storage layer tramite una distribuzione open source di b-tree ritenendo che l' articolazione delle informazioni contenute in un b-tree fosse troppo complessa da spiegare, laddove se ne presentasse la necessità, nella fase dibattimentale di un procedimento giudiziario; pertanto è stato adottato un approccio più semplice basato su una struttura detta AFF segment [9], ripetibile e di lunghezza variabile. Ogni AFF segment consiste di un header, un nome del segmento, un flag di 32 bit, una area dati di lunghezza variabile, un footer. La lunghezza è memorizzata sia nello header che nel

footer. Un file AFF inizia con un file header e termina con una directory che contiene la lista di tutti i segmenti del file e il loro offset in byte dall'inizio del file.
Il disk representation layer definisce nomi specifici di segmento per rappresentare informazioni sui dischi e meta informazioni. Queste possono essere memorizzate nello stesso AFF file dell'immagine oppure in un file separato; lo schema può essere memorizzato anche in un file XML.
I segmenti di dati hanno tutti la stessa ampiezza che viene determinata al momento della creazione del file immagine. I segmenti possono essere compressi con lo strumento open source zlib o lasciati non compressi secondo scelte da compiere al momento dell'esecuzione.
Per rendere più usabile il sistema e sollevare i programmatori dalla comprensione di molti dettagli implementativi è stata costruita la libreria AFFLIB che fornisce un'astrazione semplice dei file immagine AFF che appaiono come l'insieme di un data base nomi-valori e di un file standard che può essere aperto, letto, e acceduto in ricerca con chiamate di libreria.
Il codice AFF è fornito [10] assieme a un insieme di strumenti come un programma per eseguire l'imaging del disco, un programma di conversione da AFF a XML, un programma per convertire, nei due versi, file grezzi in file AFF.

Utilizzo di Advanced Forensic Format
Lo scopo che perseguiamo è di partecipare alla progettazione e implementazione di un "open EnCase" in linea con le iniziative dei paragrafi precedenti e riteniamo AFF uno strumento molto valido come formato di riferimento.
Per quanto riguarda AFF abbiamo eseguito alcune prove utilizzando:
- AFFlib compilato su Debian GNU/Linux 3.1;
- immagini grezze di reperti relativi ad alcuni casi giudiziari per la conversione da dati grezzi a AFF;
- storage USB di vario tipo per la creazione di immagini in formato AFF.

Le prove hanno avuto lo scopo di confrontare AFF con altri strumenti utilizzati in attività di consulenza in casi penali; si è rilevato che rispetto:
- a prodotti simili e a EnCase, AFF mentre esegue l'acquisizione del reperto calcola anche lo hash SHA1 e MD5 dell'immagine grezza consentendo un sostanzioso risparmio di tempo rispetto alle procedure a più passi;
- a prodotti simili, AFF consente di creare l'immagine dei dati grezzi compressa permettendo l'apertura e la lettura del file senza dover decomprimere in un secondo momento l'immagine;
- a prodotti simili, AFF consente una visualizzazione a elevata usabilità di informazioni dettagliate riguardanti i segment, meta informazioni, dati sullo hash dei singoli file;
- a prodotti simili e a EnCase, AFF produce file compressi più piccoli;
- a EnCase, AFF consente di superare il limite di creazione di immagine di due Gigabytes.

Gli strumenti di AFF che abbiamo sperimentato riguardano:
- aimage per la creazione di nuove immagini in formato AFF;
- aconvert per la conversione di immagini grezze in immagini AFF;
- acompare per confrontare un'immagine grezza con una in formato AFF;
- ainfo per visualizzare a video le informazioni dettagliate riguardanti l'immagine AFF;
- acat per creare un immagine grezza da un immagine AFF.

Attualmente le funzioni di lettura ed apertura delle immagini in formato AFF, la citata libreria AFFLIB, sono state implementate nella nuova release di Sleuthkit [11], più precisamente:
- af_open() per l'apertura delle immagini in formato AFF;
- af_read() per la lettura dei dati contenuti nell'immagine AFF;
- af_seek() per la ricerca di dati all'interno dell'immagine AFF;

- af_write() per la scrittura all'interno dell'immagine AFF.

Abbiamo eseguito alcuni confronti di prestazioni tra un immagine acquisita in formato AFF e un immagine acquisita utilizzando dd_rescue [12]; l'acquisizione è stata fatta da un'unità storage da 128 Mb e da un hard disk da 40 Gb cosi da poter valutare i comportamenti dei due programmi su diverse moli di dati.
Si ricordano due casi.
Nel primo le operazioni sulle immagini create sono state fatte utilizzando Autopsy [13], programma che permette un interfacciamento delle funzioni di Sleuthkit con un ambiente grafico. La creazione di un'immagine acquisita con dd_rescue sulla chiavetta da 128 Mb è terminata dopo 19 secondi, mentre l'immagine creata utilizzando AFF ha richiesto 31 secondi con la differenza che:

- l'immagine AFF è compressa al 51% (66 Mb sui 128 dell'immagine creata con dd_rescue);
- lo hash SHA1 e MD5 sono stati calcolati durante l'acquisizione;
- il file dell'immagine è stato cifrato in SSL.

IL secondo caso fa riferimento all'acquisizione da hard disk da 40 Gb con dd_rescue terminata dopo 58 minuti, mentre l'immagine creata utilizzando AFF ha richiesto 88 minuti.
Si può dedurre che l'acquisizione di un reperto in formato AFF comporta un netto risparmio di tempo e memoria migliorando l'organizzazione dei dati garantendone la sicurezza e l'integrità dell'immagine; si evidenzia che le prestazioni delle funzioni che permettono l'analisi dei dati, come af_open af_read e af_seek (eseguite entrambe con Autopsy), contenuti nelle immagini in formato AFF risultato analoghe a quelle di un'analisi fatta su immagini acquisite con dd_rescue.

Referenze

[1] Maioli C., *Dar voce alle prove: elementi di Informatica forense*, in P. Pozzi (a cura), *La sicurezza preventiva dell'informazione e della comunicazione*, FrancoAngeli, 2004
[2] http://www.guidancesoftware.com
[3] http://www.forensicswiki.org/index.php?title=Forensic_file_formats
[4] Panda B. e J. Giordano, D. Kalil, *Next-generation cyber forensics: introduction*, CACM 49, 2, February 2006
[5] CDESF, *Standardizing digital evidence storage*, CACM 49, 2, February 2006
[6] Turner P., *Unification of digital evidence from disparate sources*, 5th Annual Digital Forensic Workshop, New Orleans, 2005
[7] Garfinkel S.L. et alii, *Advanced Forensic Format: an open extensible format for disk imaging*, IFIP WG 11.9 International Conference on Digital Forensics, Orlando, January 2006
[8] http://en.wikipedia.org/wiki/BSD_License
[9] Garfinkel S.L., *AFF: a new format for storing hard drive images*, CACM 49, 2, February 2006
[10] http://www.afflib.org
[11] http://www.sleuthkit.org/
[12] http://www.garloff.de/kurt/linux/ddrescue/
[13] www.sleuthkit.org/autopsy

Cesare Maioli (cesare.maioli@unibo.it) è professore ordinario di Informatica giuridica e Informatica forense all'Università di Bologna e membro del Cirsfid; Stefano Fratepietro (fratepietro@cirsfid.unibo.it) è sistemista di rete al Cirsfid dell'Università di Bologna.

Privacy and Data Mining: the GeoPKDD Approach

Francesco Bonchi[1], Fosca Giannotti[1], Dino Pedreschi[2], and Franco Turini[2]

[1] Pisa KDD Laboratory, ISTI - CNR, Pisa, Italy
e-mail: {maurizio.atzori, francesco.bonchi, fosca.giannotti}@isti.cnr.it
[2] Pisa KDD Laboratory, Computer Science Department, University of Pisa, Italy
e-mail: {pedre, turini}@di.unipi.it

Extended Abstract

Privacy is essential for the provision of electronic and knowledge-based services in modern e-business, e-commerce, e-government, and e-health environments. Nowadays, service providers can easily track an individual's actions, behaviors, and habits. Given large data collections of person-specific information, providers can *mine* these data to learn patterns, models, and trends that can be used to provide advanced and personalized knowledge-based services. Data mining is a term widely used to indicate a broad range of analysis techniques aimed at *extracting* useful and actionable knowledge from large databases. The potential benefits of data mining are substantial, but it is evident that the collection and analysis of sensitive personal data arouses concerns about citizens' privacy, confidentiality and freedom. Source data of particular importance include, for instance, biomedical patient data, web usage log data, mobility data from wireless and sensor networks: in each case there exist substantial privacy threats, as well as a potential usefulness of knowledge discovered from these data.

When addressed at a technical level, privacy-awareness fosters the dissemination and adoption of emerging knowledge-based applications. Obtaining the potential benefits of data mining with a privacy-aware technology can enable a wider social acceptance of a multitude of new services and applications based on the knowledge discovery process. This consideration is at the basis of the GeoPKDD project – Geographic Privacy-aware Knowledge Discovery and Delivery, project number 01495 within the Future Emerging Technologies program of FP6-IST.

The GeoPKDD project

The general goal of the GeoPKDD project is to develop theory, techniques and systems for knowledge discovery and delivery, based on new automated privacy-preserving methods for extracting user-consumable forms of knowledge from large amounts of raw data referenced in space and in time. Particular emphasis is placed upon:

– Devising methods for representing, storing and managing moving objects, which change their position in time, and possibly also their shape or other features, together with their trajectories, with varying levels of accuracy and certainty;
– devising spatio-temporal knowledge discovery and data mining methods and algorithms for moving objects and their trajectories;
– devising native techniques to make such methods and algorithms intrinsically privacy-preserving, as data sources typically contain personal location-aware sensitive data.

The motivations for undertaking this direction of research are rooted in the consideration that spatio-temporal, geo-referenced datasets are, and will be, growing rapidly, due to, in particular, the collection of privacy-sensitive telecommunication data from mobile phones and other location-aware devices, as well as the daily collection of transaction data through database systems, network traffic controllers, web servers, sensors.

The large availability of these forms of geo-referenced information is expected to enable novel classes of applications, where the discovery of consumable, concise, and applicable knowledge is the key step. As a distinguishing example, the presence of a large number of location-aware wirelessly connected mobile devices presents a growing possibility to access space-time trajectories of these personal devices and their human companions: trajectories are indeed the traces of moving objects and individuals. These mobile trajectories contain detailed information about personal and vehicular mobile behaviour, and therefore offer interesting practical opportunities to find behavioural patterns, to be used for instance in traffic and sustainable mobility management, e.g., to study the accessibility to services.

Clearly, in these applications privacy is a concern. In particular, how can trajectories of mobile individuals be stored and analysed without infringing personal privacy rights and expectations? How can, out of privacy-sensitive trajectory data, patterns be extracted that are demonstrably privacy-preserving, i.e., patterns that do not disclose individuals' sensitive information?

Privacy preserving data mining

Privacy preserving data mining, i.e., the study of data mining side-effects on privacy, has recently captured the attention of many researchers and administrators across a large number of application domains. This is made evident by the fact that major companies, including IBM, Microsoft, and Yahoo, are allocating significant resources to study this problem. Despite such efforts, and many important research results in the last years, there are still many open issues that deserve further investigation. One of today's critical challenges is that, despite increasing interest in privacy from academia, corporations, and government agencies, there remains a lack of technology transfer in privacy preserving data mining technologies. This problem stems from different facts: firstly, privacy concerns and data mining endeavors vary across application domains, and it is

not straightforward how to generalize technical solutions from specific applications to principles; secondly, there exists an evident and obvious communication gap between scientists that develop theories and technical solutions, and the lawyers that define the regulations regarding privacy issues in data collection and analysis.

We believe that real solutions to the challenges posed by the applications, as those ones studied within GeoPKDD project, can only be achieved through a combination of technical tools, legal regulations and social norms. On one side, a regulatory context poses challenges and constraints for novel technical solutions; in turn, new solutions might provide feedback and opportunities towards better norms and regulations. To implement this optimistic synergy, it is needed a more frequent and fruitful cooperation between the scientists and the lawyers: both sides need to be constantly aware of the progress developed by the opposite side.

The GeoPKDD Privacy Observatory

In the context of the GeoPKDD project, while we investigate the technical advances needed to embed privacy into the data mining tools, we have activated a *privacy regulation observatory*, aiming at involving the representatives of the national and European privacy authorities, as well as non-governmental privacy-related associations. The observatory will be aimed at harmonizing the activity in the project with existing regulations, which may emerge from the privacy-preserving methods developed within the project.

The first steps in this direction have been the set up of regular relationships with the Italian Authority for Privacy (Autorità Garante della Privacy). Italy implemented the main European directive, Directive 95/46, in 1996 by law no. 675/96. The Authority analyses many cases every year, and establishes sanctions when they find that the rules are violated. The directives, both the European one and the national implementations, are, as usual for directives, very declarative and qualitative. Technical definitions, as for example *k-anonymity*, can be very useful for the Authority. *K-anonymity* establishes that privacy is guaranteed at a certain level if the individual is in a group of at least k individuals and there is no way to distinguish among them.

As a first technical result in this context, we have shifted the concept of *k-anonymity* from databases to patterns extracted by means of data mining techniques. In fact, it is generally believed that data mining results do not violate the *anonymity* of the individuals recorded in the source database. In fact, data mining models and patterns represent a large number of individuals and thus conceal individual identities: this is the effect of the minimum support threshold in association rule mining. In [1] we have shown that this belief is ill-founded. By shifting the concept of *k-anonymity* from data to patterns, we have formally characterized the notion of a threat to anonymity in the context of pattern discovery, and provided a methodology to efficiently and effectively identify all possible such threats that might arise from the disclosure of a set of extracted patterns.

In summary, our aim is to act for the Authorities as technical consultants in the field of privacy preserving data mining, as much as a mechanical engineer can help the judge in evaluating the speed of a car involved in an accident.

References

1. M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. k-anonymous patterns. In *Proceedings of 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05), Porto, Portugal, 2005.*

Preserving k-anonymity in spatio-temporal datasets and location-based services

Claudio Bettini Sergio Mascetti
DICo, University of Milan, Italy

(Extended Abstract)

Modern technologies make it relatively easy and inexpensive to collect a large amount of personal information. Several organizations are asked, for very different reasons, to release the personal data they have acquired. While in some cases it is acceptable to release information in a statistical form, hence easily avoiding privacy violation, in many others it is really necessary to release specific data (also called microdata). In recent years, anonymization techniques have received great attention as a tool to distribute microdata without endangering users' privacy. In particular, k-anonymization is a technique introduced in [5, 6] to protect the anonymity of data from so called *external linking* attacks. Indeed, it is not sufficient to hide data that explicitly identifies an individual, but attention should be paid to release values of attributes that could be found in external sources associated with other data that could lead to the identification. As an example, it is not sufficient to hide name and address of a person if we are also releasing her phone number and the number could actually appear in a public listing. Hence, the technique is based on the generalization of the values of the attributes that may be used to perform external linking, guaranteeing that in the released dataset each combination of values of these attributes appears associated to at least k individuals.

In our research we focus on the privacy issues that arise when spatio-temporal data is part of the collected information, and needs to be released. Spatio-temporal data is frequently collected; for example, each time an ATM machine is used or a payment is performed by a credit card, the time and the location associated with the transaction are recorded. The collection of spatio-temporal data will become more frequent in the next years as a consequence of the diffusion of location-based services, i.e., services that, based on the user current position, can provide location-aware information. Typical examples are map and navigation services, services that provide information on close-by public resources (e.g., gas stations, pharmacies, ...), services that provide localized news (e.g., weather forecasts, road constructions, etc.), as well as more personalized services like proximity marketing or friend-finder.

We aim to extend existing anonymization techniques in order to apply them to the case where spatio-temporal information is part of the data to be released.

The challenge is to guarantee the anonymity of the released information while reducing data generalization. Although in theory existing approaches to k-anonymity can be used to generalize spatio-temporal information, we argue that a better tradeoff between the degree of anonymity and the significance of the released microdata can be obtained by adopting specific techniques for spatio-temporal datasets. Our research follows three main directions.

Preserving k-anonymity in spatio-temporal datasets. Most of the existing approaches to k-anonymity in databases assume that in the table to be released there is at most one tuple for each user (e.g., [5, 6, 3, 2]). This assumption is made for the sake of simplicity and, indeed, it can be quite easily removed. However, when multiple tuples for each user are present, like in the case of spatio-temporal data, another form of inference can be performed, exposing the released data to respondents identification even if the data is proven to be k-anonymous according to standard definitions.

Example 1 *Consider a location-based service in which users send messages containing the current user location and some sensitive information. The server stores, in its local database, a tuple for each request and identifies the users by pseudonyms. When the database needs to be released, it is necessary to generalize the spatio-temporal data, because, if used together with external information, it may lead to personal identification. The following is a possible table to be released.*

Location	***Time***	***Pseudonym***	***Data***
l_1	*2005-11-22*	p_1	d_1
l_2	*2005-11-23*	p_1	d_2
l_1	*2005-11-22*	p_2	d_3
l_2	*2005-11-23*	p_3	d_4

Assuming Location and Time are the quasi identifiers (i.e., the attributes that could be used, through external sources, to reduce anonymity), according to the common definition of k-anonymity, the table is 2-anonymous. Indeed, there are two tuples for each pair of values Location and Time in the table. The first two tuples belong to the same user, but this is irrelevant in this case to satisfy 2-anonymity, since each combination of the quasi-identifier is associated with 2 users.

Now, consider an external source of information including the following tuples:

User	***Location***	***Time***
u_1	l_1	*2005-11-22*
u_1	l_2	*2005-11-23*
u_2	l_1	*2005-11-22*
u_3	l_2	*2005-11-23*

From this data, we derive that users u_1 and u_2 are associated with the quasi identifiers value pair $< l_1$,2005-11-22>, while users u_1 and u_3 with $< l_2$,2005-11-23>. However, we also derive that only user u_1 can be associated with both

tuples. If the table is released, considering the values of the Pseudonym attribute, a recipient may observe that the first two tuples belong to the same user and, using the reasoning from the external source, the recipient can derive that only u_1 can be associated with the first two tuples in the table, hence violating the intuitive notion of 2-anonymity.

Intuitively, the presence of multiple tuples for each user enables a form of *internal linking* that allows an attacker to group tuples associated with the same (unknown) user.

In order to solve the problem, we propose a stronger notion of k-anonymity and related generalization algorithms. The new definition supports the anonymization of tables with more than one tuple for each respondent, and guarantees that each tuple cannot be associated to less then k users, even considering attacks like the one illustrated above.

Spatio-temporal generalization. One of the most common generalization techniques used in k-anonymization approaches is based on partial string suppression; for example, the two zip codes `12345` and `12349` can be both generalized into `1234*`. While this approach can also be used for spatio-temporal attributes, it may not be an effective solution. For example, considering the timestamp `2006-06-22`, a generalization as the one presented above is necessarily based on a total order of time granularities, and it is usually limited to the most common ones, as year (`2006-*`) or month (`2006-06-*`). Hence, partial string suppression may sometimes lead to timestamps that are being generalized too much, possibly making data unusable.

A different approach consists in changing the time attribute in order to reflect the fact that its values denote granules of an arbitrary time granularity taken from a possibly large partially ordered set. In the example above, a generalization in terms of `weeks` (`2006-06-week(3)`) could be preferred to a generalization in terms of `months` if k-anonymity can be guaranteed with both of them, even if, formally, the granularity `week` is not *finer* than the granularity `month` and vice versa. In [4] we proposed a generalization algorithm specific to temporal attributes. In the future we are planning to develop a similar approach to spatial attributes and to the combination of spatial and temporal attributes.

Anonymization in location-based services. Some location based services are currently available through the mobile phone operators and many more will be offered soon. Since some third party service providers may be untrusted and the information transmitted to them may be sensitive, it is important to guarantee the anonymity of the user's requests. This can be obtained by using a trusted location server that collects the requests from the users and forward them to the service providers once properly anonymized.

As in the standard database anonymization problem considered above, we need to guarantee that the data recipient (the untrusted service provider) is not able to associate less then k possible users to a service request. We consider two situations:

- The service provider has no way to link different requests. In this case,

each request should be anonymized considering only the available external information about the users present in the same area at the same time.

- The service provider has access to the history of requests and is able to trace different requests performed by the same user. In this case the anonymization should also consider the information that was forwarded to the service provider for the previous requests. This case poses a problem similar to the one described in Example 1. The main difference here is that we are not dealing with a static database, but with a continuously growing dataset.

Our main challenge is to develop an efficient and practical solution to the latter situation explained above. A preliminary investigation appears in [1].

References

[1] Claudio Bettini, Xiaoyang Sean Wang, and Sushil Jajodia. Protecting privacy against location-based personal identification. In Willem Jonker and Milan Petkovic, editors, *Secure Data Management*, volume 3674 of *Lecture Notes in Computer Science*, pages 185–199. Springer, 2005.

[2] Roberto J. Bayardo Jr. and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the 21st International Conference on Data Engineering*, pages 217–228. IEEE Computer Society, 2005.

[3] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Incognito: efficient full-domain k-anonymity. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM Press, 2005.

[4] Sergio Mascetti, Claudio Bettini, X. Sean Wang, and Sushil Jajodia. k-anonymity in databases with timestamped data. In *Proc. of 13th International Symposium on Temporal Representation and Reasoning*. IEEE Computer Society, 2006.

[5] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.

[6] Latanya Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

A Mathematical Framework to Assess the Security of an Information Infrastructure

F.Baiardi, S.Suin, C.Telmon

Dipartimento di Informatica, Università di Pisa
L.go B.Pontecorvo 3, 56125 - PISA
{f.baiardi, s.suin} @unipi.it, claudio@next-hop.it

1 Introduction

We propose a mathematical framework to assess the security of an information infrastructure focused on the analysis of rights acquired by a threat through a sequence of attacks. The most critical problem to be faced when defining the framework is how to deduce all the rights acquired by a threat through a successful attack because these rights depend upon the relation among components of the considered infrastructure. As an example, a successful attack to control a component also enables the control of any component that depends upon the attacked one. To describe the relation, we model components as objects, each defining of a set of methods invoked by either a user or other components. Under this assumption the relations among components may be modelled through a labelled hypergraph where an oriented hyperarc denotes a dependency of the destination node from the source ones. Relations that are considered concern the ability of invoking, controlling or managing a method as well as that of determining a security attribute of a component. In this way, the framework supports a modular description of the infrastructure at distinct abstraction levels that makes it possible to focus the assessment on the most critical components. We show how the framework may be used to define an optimal ordering to remove vulnerabilities. Some programming tools have been developed according to the framework. They have been implemented through a logic programming language and exploit the hypergraph to automatically deduce information such as the attack sequences that a threat may execute against the infrastructure or the smallest set of countermeasures to be applied.

2 Infrastructure Hypergraph

The proposed framework describes the infrastructure as a set of related components. A component consists of some internal state and methods, each an operation the component implements. Legal users own the rights to invoke some methods while threats are interested in gaining some rights according to their goals. In the following, both legal users and threats will be denoted simply as user. Each user is paired with a set of rights, each defining a method it can invoke. Some of these rights may have been achieved through a sequence of attacks. For each infrastructure component, we determine a set of methods such that anyone that has the rights of invoking all the methods in the set then can also control some security attribute of the component. A further relation among the components describes how a security attribute of a component depends upon those of other components. Taking into accounts the rights of a user U, i.e.

the methods U can invoke, and the relations among methods and security attributes of components we can compute the transitive closure of the rights of U. This closure includes any attribute of a component that U can control because of its rights. Assume, as an example, that a method M of C determines the integrity of a component C and that the confidentiality of a distinct component D depends upon the integrity of C. Hence, any user that can invoke M controls the confidentiality of D. In another case of interest, a method M updates the state of a component ACL used to grants or revokes the right of invoking F. Obviously, if U can invoke M, then it can also grant or revoke the right of invoking F. Here, any user that can invoke M manages, i.e. controls the availability of F. In the most general case, the hypergraph includes three kinds of nodes that describe, respectively, users, components and methods. Any hyperarc of the hypergraph has one of the following kinds:

1. right hyperarc: from a user node to a set of method nodes. It is not labelled and it describes the methods a user can invoke;
2. source hyperarc: from a set of method nodes to a component node. It is labelled by one attribute of the component and it describes the methods that control the corresponding attribute of the component;
3. component hyperarc: from a set of component nodes to a component node. It is labelled by one attribute for each source component and for the destination one. It describes a dependence of the attribute in the destination component from those of the source ones;
4. destination hyperarc: from a set of components nodes to a method node. It is labelled by one attribute for each source node and for the destination one. It describes a dependency of a method from a set of components.

To compute a transitive closure, rights are propagated through the hyperarcs.

3 Vulnerability and Attacks

The framework characterizes an attack in terms of the vulnerabilities that enable it, the resources and the rights it requires and the rights achieved if it is successful. If A is an attack, V(A) is the set of the component vulnerabilities that enables A and C(A) is component that is the target of A. A user U can execute A if it can access all the resources in R(A), i.e. information, tools and know how about A. Furthermore, U can attempt A only if it satisfies pre(A), the precondition of A. Pre(A) is a set of rights and U satisfies pre(A) if it owns all the rights in the set. If U owns all the resource in R(A) and satisfies pre(A), then it can execute A and, if A is successful, it will acquire the rights in post(A).

To deduce the sequence of attacks U can execute, we consider that initially it owns the rights in Init(U) and it can execute the sequence A1...An if it owns the resources for any Ai $1 \leq i \leq n$ and if, after executing the sequence A1 ... Aj, it satisfies pre(Aj+i). Hence, the transitive closure of Init(U) should include pre(A1) and, after any Ai, $i \in 1..n-1$, the transitive closure of $Init(U) \cup post(A1) ... \cup post(Ai)$ should include pre(Ai+1). Each sequence satisfying these conditions is **feasible** for U. For simplicity sake, we neglect the dependency of feasible sequences of U from Init(U).

Since U is rational, it executes only those feasible sequences that enable it to achieve one of its goals. Each goals of U is a set of rights and U achieves a goal when and if it

owns all the corresponding rights. A feasible sequence that enables U to achieve one of its goals is an **evolution useful** for U. Useful evolutions depend upon:

1. the infrastructure hypergraph,
2. the vulnerabilities in the infrastructure components and the attacks they enable,
3. the attacks U can implement and its goals.

4 Security Evolutions of the Infrastructure

Security evolutions describe how n users in a set SU can achieve n goals, SR1, ..., SRn, through a sequence of attacks SA = A1 ... Am, where each attack is executed by just one user. To define evolutions, first of all we define the projection PR(SA, U) of the sequence SA onto a user U. PR(SA, U) includes the subsequence of SA with the attacks implemented by U. This subsequence is a useful sequence for U. If PR(SA, U) is empty, U is not involved in the evolution due to SA. Since each attack in SA is implemented by a user in SU as a step to reach a goal, there is a set $\{U1, ..., Uk\} \subseteq SU$ and a corresponding set of projections $\{PR(SA, U1), ..., PR(SA, Uk)\}$ where

1. each attack in SA belongs to one projection only
2. distinct projections are disjoint,
3. after the execution of attacks in PR(SA, Ui), Ui achieve its goal SRi.

Hence, each evolution results from the interleaving of useful sequences for the considered users. Two evolutions are equivalent, if any user in the considered set achieves the same rights. Since we assume user rights are never revoked, we can model monotonic evolutions only, where the set of user rights never decreases. Given a set of user, each with a goal, they can achieve their goals iff there is at least one corresponding evolution. Evolutions can be computed taking into account the hypergraph and useful sequences.

5 Ranking of Vulnerabilities

The proposed framework may exploit alternative metrics to rank vulnerabilities according to the evolutions they enable, the impact of these evolutions and so on. Here we present a metric based upon the smallest sets of countermeasures to prevent any user from achieving its goals or, alternatively, to stop all evolutions. A countermeasure for a vulnerability V is any strategy that removes V, i.e. it makes an attack that exploits V ineffective. In the simplest case, a patch that removes V is applied. Other countermeasures may change the dependencies among components so that even if a user owns a set of rights, it cannot control or manage a component. A set of countermeasures is complete if after applying its countermeasures, no evolution is possible and no user can achieve any of its goals. A set of countermeasures is minimal if it is complete and none of its subsets is complete. Minimal sets define the smallest sets of countermeasures to be applied to stop all the evolutions

To rank a vulnerability V, we consider the percentage of minimal sets with a countermeasure that removes V. To show that this percentage conveys a useful information consider that if no minimal set removes V, then evolutions can be stopped even if V is not removed. On the other hand, if any minimal set includes a countermeasure that removes V, then the only way to stop evolutions is to remove V.

Till now we have assumed that any two goals of any user are equivalent. Instead, in several cases, each goal may be paired with a weight proportional to its impact, i.e. to the corresponding loss for the infrastructure owner. In the same way, we can define a minimal set in terms of the cost of the countermeasures rather than of their number.

A set of tools has been implemented to support the assessment. They are written through a logic programming language and apply to the infrastructure hypergraph a set of analyses that compute:

1. the sets of rights each user may achieve,
2. alternative evolutions due to a set of users,
3. minimal sets of countermeasures,
4. the set of rights a user may achieve after applying a set of countermeasures.

References

1. P.Ammann, D.Wijesekera, S. Kaushik, *Scalable, Graph-based Network Vulnerability Analysis*, Proc. of the 9th ACM conference on Computer and communications security, November 18-22, 2002, Washington, DC, USA
2. F.Cuppens, A. Mie'ge, *Alert Correlation in a Cooperative Intrusion Detection Framework*,IEEE Symp. on Security and Privacy, p.202, May 12-15, 2002
3. J. Dawkins, C. Campbell, J. Hale, *Modeling Network Attacks: Extending the Attack Tree Paradigm*, Workshop on Statistical and Machine Learning in Computer Intrusion Detection, Johns Hopkins University, June 2002.
4. R. P. Goldman, W. Heimerdinger, and S. A. Harp. *Information Modeling for Intrusion Report Aggregation*,DARPA Information Survivability Conference and Exposition (DISCEXII), June 2001.
5. S. Jajodia, S. Noel, B. O'Berry, *Topological Analysis of Network Attack Vulnerability*, in *Managing Cyber Threats: Issues, Approaches and Challenges*, V. Kumar, J. Srivastava, A. Lazarevic (eds.), Kluwer Academic Publisher, 2003.
6. R. A. Martin, *Managing vulnerabilities in networked system*, IEEE Computer, November 2001. p. 32 - 38.
7. P. Moore, R. J. Ellison, R. C. Linger,*Attack modelling for information security and survivability*, CMU/SEI- 2001-TN001.
8. P. Ning , P.Cui , D. S. Reeves, *Constructing attack scenarios through correlation of intrusion alerts*, 9th ACM Conference on Computer and Communications security, Nov. 2002, Washington, DC, USA.
9. S. Jha, O. Sheyner , J. Wing, *Two Formal Analysis of Attack Graphs*, 15th IEEE Computer Security Foundations Workshop , p.49, June 2002.
10. C. Phillips, L. Painton Swiler, *A graph-based system for network-vulnerability analysis*,Workshop on New Security Paradigms, p.71-79, Sept.1998.
11. R. Ritchey, B. O'Berry, S. Noel, *Representing TCP/IP Connectivity For Topological Analysis of Network Security*, Proc. of the 18th Annual Computer Security Applications Conference, p.25, Dec. 2002.
12. O. Sheyner, J. Haines, S. Jha , R. Lippmann, J. M. Wing, *Automated Generation and Analysis of Attack Graphs*, Proc. of the 2002 IEEE Symposium on Security and Privacy, May 12-15, 2002 .
13. O. M. Sheyner, *Scenario Graphs and Attack Graphs*, CMU-CS-04-122,2004.
14. L.P. Swiler, C. Phillips, D. Ellis, S. Chakerian, *Computer-Attack Graph Generation Tool.* Proc. of the DARPA Information Survivability Conference, June 2001.
15. V.Swarup, S.Jajodia, J.Pamula, *Rule-Based Topological Vulnerability Analysis*, Proc. of MMM-ACNS 2005, Sept. 2005.

SockMi: how to migrate SSL sessions

Massimo Bernaschi
Istituto Applicazioni del Calcolo, CNR
V.le del Policlinico, 137, 00161 Rome, Italy
<massimo@iac.rm.cnr.it>

Luigi V. Mancini
Università degli Studi di Roma La Sapienza
P.le Aldo Moro 5, 00185 Rome, Italy
<lv.mancini@di.uniroma1.it>

Paolo Tassotti
Università degli Studi di Roma La Sapienza
P.le Aldo Moro 5, 00185 Rome, Italy
<tassotti@di.uniroma1.it>

Abstract

Migrating a network connection means to replace one of the peers with another process that can be on the same or on a different machine. There are a number of situations in which the migration of connections can be useful. For instance, when there are requirements of load balancing, quality of service, fault tolerance or security (e.g. honeypotting).

We present a connection migration system for Linux systems called *SockMi*, a general-purpose solution that can be easily adopted for a wide range of applications. The main idea behind SockMi is quite simple: as connection state is stored in a bunch of well-known data structures residing inside the kernel, it is feasible to achieve migration merely by copying and tranferring this data on another host.

As a consequence of this approach the mechanism is perfectly symmetric. That is, both server-side and client-side migration are supported. Another interesting feature is the unawareness of non-migrating peer, descending from the fact that we do not require any further protocol to be implemented.

Subsequently we describe how to extend SockMi in order to handle the application layer. In particular we show how to migrate an endpoint of an encrypted protocol such as SSL. The tecnique used is fairly similar to the *copy & transfer* approach mentioned before.

1 SockMi architecture

The main idea behind SockMi's connection migration system is quite simple: as connection state is stored in a set of well-known data structures residing inside the kernel, it is feasible to achieve transport layer migration merely by copying and tranferring these data on another host.

We used the same approach also for the application layer. In that case, information needed to migrate a SSL session (e.g. session key) is stored inside OpenSSL internal data structures as we will describe in section 1.4.

As a consequence of our approach the mechanism is perfectly *symmetric*. That is, both *server-side* and *client-side* migration are supported. However, in this paper we restrict our survey only to server-side migration. In section 3 we discuss how a client-side migration could be implemented.

In the following, we denote with *"export phase"* the act of reading session information on the original host and *"import phase"* the act of storing these data on the foreign host.

SockMi architecture is made of three main components: a loadable kernel module (LKM), a daemon process and a packet redirection system. In the next sections we'll describe them in details and then illustrate how they cooperate in order to achieve session migration.

1.1 SockMi module

The SockMi loadable kernel module is in charge of reading/writing socket state during the export/import phase. So, the first step is exactly define what "socket state" (or "connection state") means and what data structures are involved.

The state of a connection is defined by the Transmission Control Block (TCB), as stated in TCP protocol specification [1]. TCB includes addresses, ports, sequence/acknowledgement numbers, sliding/congestion window parameters and so on. In Linux implementation TCB data are stored in `sock` and `tcp_opt` structures[9].

Moreover we have to consider so-called *"in-*

flight data". These falls into two different categories:

- Packets received by the host but not yet read by the application (*receive queue*);
- Packets to be sent, or packet already sent but not yet acked (*transmit queue*);

In Linux implementation both of these queues are a linked list of `sk_buff` structures that contain packet payload, network headers (MAC, IP and TCP) and some additional information about the packet itself (e.g. length, checksum, etc.).

Handling transmit queue requires some additional care. In fact, when there are some unsent packets on the trasmit queue, we have to enable the TCP retrasmission timer on the import side, otherwise these packets won't be sent until the corresponding application sends another packet.

Note that since migration procedure introduces a delay, original retransmission timer value doesn't make sense on the import side[1]. The simplest solution is to set the timer to a default value. This obviously breaks congestion avoidance policies.

SockMi module provides an interface to the user-space via `sysctl()` system call. This allows a process to export one of its sockets in very simple manner.

1.2 SockMid daemon

The SockMid daemon works in combination with the SockMi module to support the socket migration mechanism. The daemon carries out different tasks depending on the situation. During the export phase, it reads the state of exporting socket from the SockMi module internal buffers. During the negotiation phase, it communicates with other SockMi daemons running on other hosts in order to choose where to migrate the socket. Finally, during the import phase, it writes the state of importing socket to the SockMi module internal buffers.

Since the module lives in the kernel address space whereas the daemon is a normal user process, it is not possible to resort to standard Inter Process Communication (IPC) mechanisms to pass data between them. To overcome this difficulty we implemented a buffer sharing system via the `mmap()` primitive.

[1]Timer values are CPU clock dependant, so it wouldn't make sense in any case.

We also used some other standard tecniques for coordinating module and daemon. During the export phase we used a signal USR1 from the module to the daemon in order to *"wake up"* the latter and start the migration procedure. During the import phase we used an `ioctl()` to notify the module that the socket data had been succesfully transferred.

1.3 Packet redirection system

SockMi includes a packet redirection system providing network layer migration. When a socket migrates to a different host it is necessary to redirect the packets coming from the peer towards the host that imports the socket. Moreover the packets sent to the peer must have the same IP source address of the original host (otherwise the peer replies with a RST packet).

To deal with this problem we resort to a special combination of Network Address Translation (NAT) operations. In particular, we employ a Destination NAT (DNAT) on the exporting host and a Source NAT on the importing host.

Moreover, we had to prevent any unexpected connection termination during and after the migration. Two possible cases exist:

- When the exporting host receives packets that it should redirect to the importing host, the TCP layer automatically sends RST packets because the connection is considered closed;
- When the process that exported the socket terminates, the TCP layer sends the FIN sequence that causes the shutdown of the connection.

We solved both these cases by defining a filter that drops all RST and FIN packets sent by the exporting host to the peer after the migration.

Both packet redirection system and unexpected connection termination avoidance was implemented with Netfilter[10].

1.4 SSL session migration

We now make a short digression on how a SSL session is established and which data structures are involved in OpenSSL implementation.

The preliminary task in order to establish a SSL session is to create a *"SSL Context"*. The SSL context (struct `SSL_CTX`) contains the

following information (we annotate associated data structures in OpenSSL implementation):

- Supported ciphers (`EVP_CIPHER_CTX`);
- Compression Algoritms (`COMP_CTX`);
- Digest algoriths (`EVP_MD`;
- Certificates (`X509`);
- Public/private keys (`EVP_PKEY`);

After context creation a new SSL session associated with that context can be established[2]; this phase is called *"SSL Handshake"*. The information related to a SSL session (struct `SSL`) are:

- Ciphers state (`EVP_CIPHER_CTX`);
- Compression Algoritms state (`COMP_CTX`);
- Digest algoriths state (`EVP_MD`);
- Session data:
  - Session Key (`EVP_PKEY`);
  - Peer Certificate (`X509`);

We point out that not all data structures listed before are strictly needed in order to migrate a session. In fact, certificates and public/private keys can be ignored as they are required only in the handshake phase.

However, migrating all information can be useful as well. Copying certificates and public/private keys enables the import side to open new SSL sessions with the same SSL context making migration mechanism much more powerful.

As we mentioned before, the same *"copy & transfer"* approach employed for TCP/IP layers has been used for application layer handling. In fact, all the information needed in order to migrate a SSL session, including context, can easily retrieved having a pointer to a `SSL` struct.

2 Application programming interface

SockMi provides a simple Application Programming Interface (API) in order to allow applications to activate the socket migration mechanism. The API consists of two functions:

- `import_socket();`
- `export_socket();`

These functions hide the implementation details of the migration mechanism and provide applications with an easy-to-use method for importing and exporting sockets.

To import one or more sockets, an application calls the `import_socket()` library function. This function is designed to poll the availability of exported sockets matching the import criteria specified by the application. If one or more matching sockets are available, then the function replaces the local sockets referenced by the input descriptors with the exported ones. Otherwise, if no matching socket is available, the function waits until either a timeout occurs or one or more exported sockets become available for import. The import criteria let the application define the properties of the socket to be imported. Such criteria are the set of allowed socket states (bound, listening or connected), the local and remote IP addresses, and the local and remote TCP ports.

We design in a similar manner the API for SSL session migration:

- `import_ssl();`
- `export_ssl();`

The `export_ssl` function takes as an argument a pointer to `SSL` struct from which it is possible to retrieve all needed information about SSL session. The `import_ssl()` applies the scheme used for `import_socket()` function. with timeout and import criteria specification. Multiple session import is also supported.

3 Future work

In these section we'll outline some future perspective.

First we'll make a short digression on possible extensions of SockMi, particularly regarding to other cryptographic protocols similar to SSL such as SSH and S-FTP. Then we survey the feasibility of porting SockMi to Windows OS.

Another important issue we're looking for is the extension of our migration mechanism to other similar protocols such as SSH[7], particularly regarding to OpenSSH[8] its *open-source* implementation. In order to correctly migrate a

[2] A context may have multiple sessions associated.

SSH session we have to deal with some typical features of remote terminals such as flow control, signals, environment variables, and so on. However, since OpenSSH relies upon OpenSSL for many of its cryptographic features, extending our mechanism should be a reasonable effort.

Up to this point we've been dealing only with open-source software and Linux OS. Nevertheless it is possible to apply the same scheme also in Windows environments. Since OpenSSL is available also for Windows, the main problem is to see whether is feasible to port the connection migration mechanism. However, REMUS[11] and WHIPS[12] projects, two IDSs developed at Università degli Studi di Roma "La Sapienza", have shown that the same kernel-based techniques can be applied either to Linux or Windows systems. Due to the lack of documentation and source code, hacking with Windows kernel is certainly much more challenging, but not impossible. From our experience we can state that connection migration technique can be fairly easily ported to Windows systems.

Acknowledgements

We would like to thank Samuele Ruco and Francesco Casadei for their major contribution during master thesis work.

References

[1] J. Postel, "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[2] A. Frier, P. Karlton, P. Kocher, "The SSL Protocol, Version3.0", Transport Layer Security Working Group Internet Draft, November 1996.

[3] T. Dierks, C. Allen, "RFC 2246: The TLS Protocol Version 1.0", Internet Engineering Task Force, January 1999.

[4] OpenSSL Project: `http://www.openssl.org`

[5] L. Stein, D. McEachern, "Writing Apache Modules with Perl and C", First Edition, April 1999

[6] Network Security Services (NSS): `http://www.mozilla.org/projects/security/pki/nss/`

[7] C. Lonvick, "RFC 4251: The Secure Shell (SSH) Protocol Architecture", Internet Engineering Task Force, January 2006.

[8] OpenSSH Project: `http://www.openssh.com`

[9] Linux Kernel Archives: `http://www.kernel.org/`

[10] Nefilter Project: `http://www.netfilter.org`

[11] M. Bernaschi, E. Gabrielli and L.V. Mancini, "A Patch to Linux for Making Buffer Overflow Harmless", Proceedings of the SANS Workshop on Securing Linux, San Francisco, December 1999.

[12] R. Battistoni, E. Gabrielli, L.V. Mancini, "A Host Intrusion Prevention System for Windows Operating Systems", ESORICS 2004, 9th European Symposium On Research in Computer Security Sophia Antipolis, French Riviera, France - September 13-15, 2004.

Common Criteria Security Certification of Complex, "Network Centric" ICT Systems

M. Lisi, G. Tassone
Telespazio SpA, via Tiburtina 965, 00156 Roma (Italy)
E-mail: marco_lisi@telespazio.it; gaetano_tassone@telespazio.it

Abstract

Today's world and its knowledge-based economy need the development of network-centric "welfare" systems, that is systems or "systems of systems" able to convey data seamlessly throughout a number of different possible media and to deliver useful information after a data fusion process.
In February 1995, in the "Annual Report to the President and the Congress", it was first established by the USA the strategy of the "information superiority", that is the capability to maintain control over all the sensitive areas of the world through a continuous flow of information in a system integrating Command, Control, Communications, Computers and Intelligence (C4I) (figure 1).

Figure 1: C4I Tactical Scenario

This strategy, together with the parallel development of new technologies (Intelligent Weapons, Unmanned Aerial Vehicles and Unmanned Combat Air Vehicles), led to a sudden increase in bandwidth capability and to the concept of "Dual Use Technology", mainly in the field of satellite communications and Earth observation.
In the recent years, the architectures of systems conceived mainly for "warfare" applications and those of systems for "welfare" applications (e.g. Earth monitoring, disaster recovery, surveillance, security) have been converging to a common "system of systems" approach, integrating management and control capabilities and a wide range of air, land, naval and space platforms within a global network-centric infrastructure (figure 2).

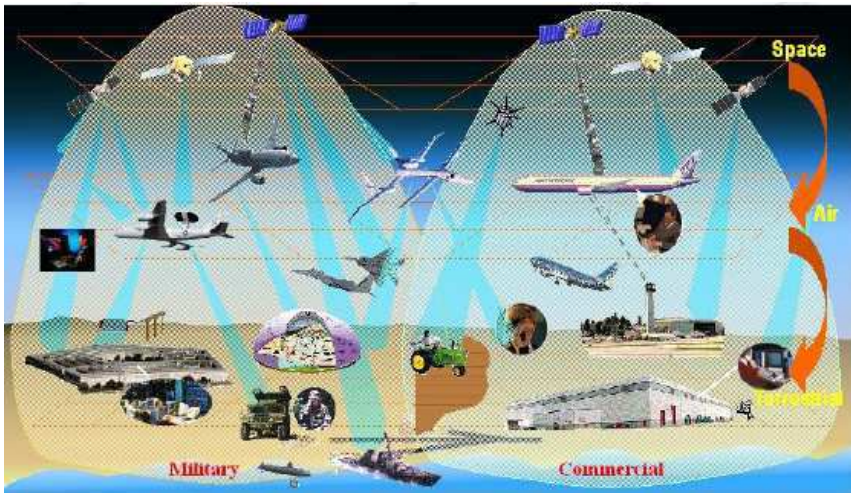

Figure 2: Convergence of Network Centric Warfare and "Welfare" Systems

In a “system of systems”, communications and information technologies (ICT’s) play a vital role, not only guaranteeing a rapid exchange of data, but also providing the required information after a process of “data fusion”.
The ICT structure must rely upon a solid telecommunications backbone, in which all cells are connected using the best suited media among strategic network, wide, local and mobile subsystems; comparable services are offered to all users, regardless of the media selected.
Satellites are key elements of the global infrastructure, both as sensors and as components of the telecommunications backbone; in the satellite ground segment the actual convergence of different media is implemented and the integration of data is performed.
One example of network centric space infrastructure is given by the European GMES system.
GMES (Global Monitoring for Environmental and Security) is a space and in-situ based Earth observation system, which will be the European contribution to the international Global Earth Observation System of Systems (GEOSS), established at the Third Earth Observation Summit held in Brussels in February 2005.
The overall GMES architecture comprises four major elements: services, space observations, in-situ observations, and data integration and information management.
Despite the importance of space-based technologies (satellites playing a double role as sensors and key elements of the communications infrastructure), GMES is not a “space-centric”, but rather a “network-centric” system.
Another example of network centric architecture is that of the GALILEO satellite radio navigation system, the joint initiative by the European Union and the European Space Agency.
The GALILEO system is based on a constellation of 30 satellites and on a number of Control Centers, implemented on European ground, to provide for the control of the constellation, to perform the navigation mission management and to monitor the system performances (figure 3).

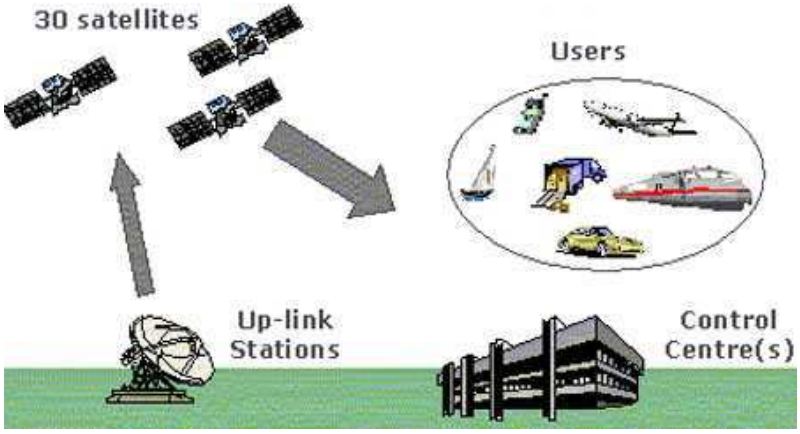

Figure 3: GALILEO System Architecture

GALILEO will provide information concerning the positioning of users, allowing the deployment of value added services in many sectors, such as transport (vehicle location, route searching, speed control, guidance systems, etc.), social services (e.g. aid for the disabled or elderly), the justice system and customs services (location of suspects, border controls), public works (geographical information systems), search and rescue systems, or leisure (direction-finding at sea or in the mountains, etc.).
A third example of network-centric satellite infrastructure and of “dual-use” system is the Italian COSMO-Skymed Earth observation system.
COSMO-SkyMed (COnstellation of Small Satellites for Mediterranean basin Observation) is an Earth observation program of the Italian Space Agency (ASI), co-funded by the Italian Defense Administration and developed by an industrial team of national companies, led by Alcatel Alenia Space (an Alcatel/Finmeccanica company). Telespazio, a Finmeccanica/Alcatel company, will be responsible for the development of the ground segment, the management of all operations, for the provision and distribution of products and value-added applications (figure 4).

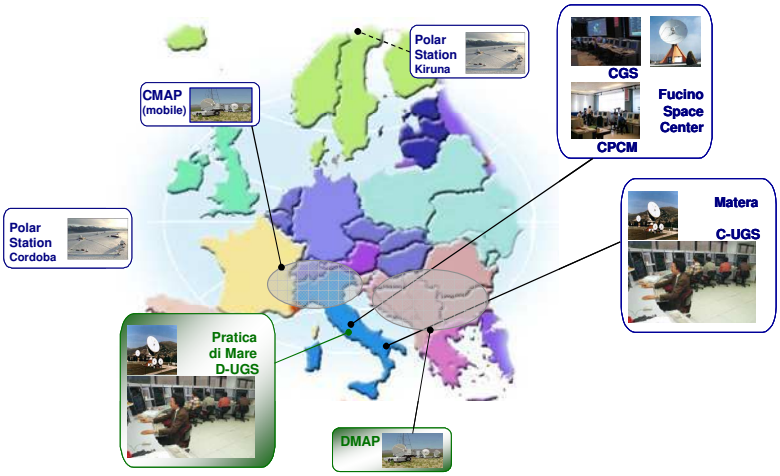

Figure 4: COSMO-Skymed Ground Segment Architecture

One last example of "network-centric" topology about satellite communications and services, is the VSAT (Very Small Aperture Terminal) broadband satellite solution provided to the global market in order to satisfy the increasing demand for bandwidth and Internet access and offering a variety of high-speed multimedia service solutions (figure 5). It is worth noting the strategic role of satelites in the communications infrastructure and the high degree of integration with other communications media (terrestrial backbone, Wi-Fi, mobile cellular, etc.).

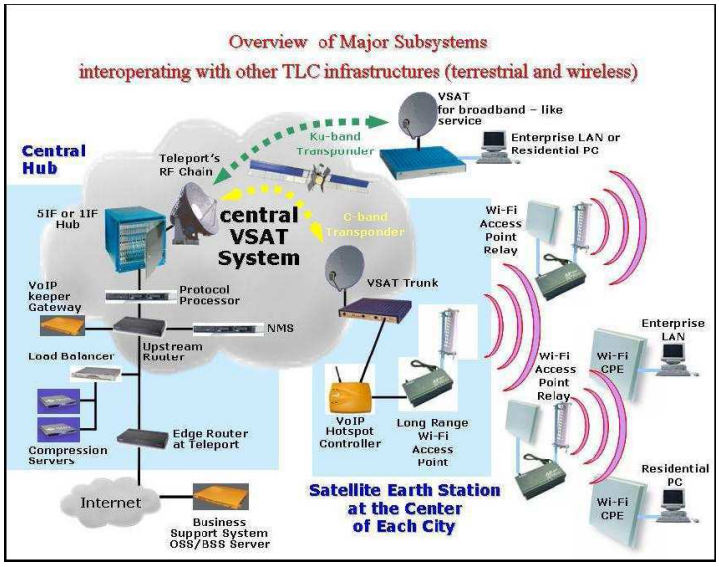

Figure 5: VSAT Broadband Satellite Network Architecture

Our society is nowadays heavily depending on Information and Communication Technology (ICT).
Information and Communication Technology has pervaded in all traditional infrastructures, rendering them more intelligent but more vulnerable at the same time.
As all critical infrastructures of our society rely on ICT systems, their confidentiality, availability, integrity, continuity and quality of service have to be guaranteed and protected against intentional and non-intentional attacks. Information security is no longer a "nice to have", but rather a "must have" option.
This is not only true for "dual use" systems, where military and civilian applications coexist, but in general for all those systems devoted to emergency services, disaster recovery, crisis management, homeland security, environment monitoring and control.
Complex ICT systems are inherently "network centric": broadband networks, both dedicated (Intranets, VPN's) and public (ISDN, Internet), are the backbone of their architectures.
Inside a complex ICT system classified and non-classified networks most often coexist and need to operate together. This is particularly true when the system needs to have access to Internet or aims at offering services on the Web.

The security certification of a complex ICT system according to an international standard, such as the so called Common Criteria (standard ISO/IEC IS 15408), implies a number of problems and technical constraints. They can be summarized in the following main categories:

- Long time required for the execution of the evaluation/certification process;
- High cost of the evaluation/certification process. Careful definition of what really needs to be evaluated and certified and against what (figure 6);
- Coexistence of classified and non-classified domains. Need for "air-gap" technologies at the classified/unclassified boundaries;
- Availability of jointly certified hardware and software platforms;
- Severe limitations in the use of commercial off-the-shelf (COTS) software products, because they seldom fulfil the certification requirements (traceability, documentation, source code availability). This creates interesting opportunities for open source operating systems (e.g. Linux) and software products;
- Limitations in the use of commonly adopted communications protocols (e.g. TCP/IP);
- Loss of certification in case of even minor modifications of the certified configuration. Need to cope with the rapid obsolescence of both hardware and software;
- Adoption of "encapsulation" techniques for the utilization of non-certified components (databases, libraries or even applications);
- Encryption of data over non-certified broadband networks.

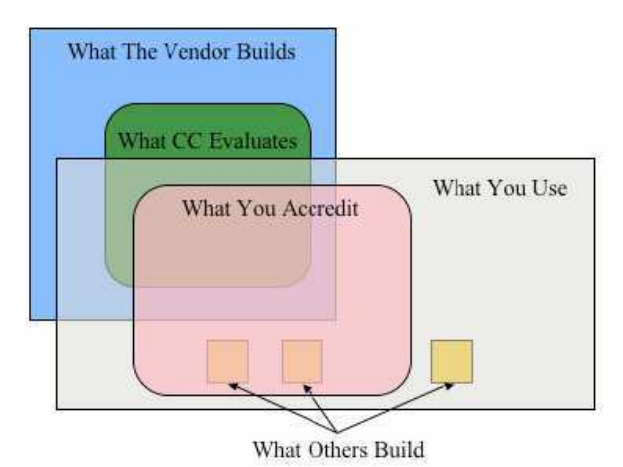

Figure 6: Common Criteria Evaluation/Accreditation Boundaries

In the perspective of a Common Criteria certification, a satellite network system presents the following features:

- outroute data traffic is encrypted in hardware using unique keys for each terminal and each type of traffic;
- a multilevel encryption scheme is employed that utilizes both hardware and software keys to prevent unauthorized system access;
- a Conditional Access System is used to control the data traffic that the terminal can receive.
- Each terminal has a hardware cryptofacility that decrypts the received traffic in real time using the keys received from the Network Operations Center (NOC);
- terminals are sent their own unique keys to ensure that they can only receive data on the satellite link that they are authorized to receive.

Data encryption is achieved by using "keys" to encrypt and decrypt messages. These keys are used with an algorithm designed to convert text or other data into digital gibberish and then restore it to its original form. Modern satellite systems utilize the Data Encryption Standard (DES) with a 56-bit key length as the bulk data encryption algorithm. The triple-DES algorithm, which in effect takes the input data and encrypts them three times, is used for additional security to protect the transmission of the satellite data encryption keys to the remote terminals. Two 56-bit keys are used by the triple DES algorithm for key distribution. A tamper-proof Application-Specific Integrated Circuit (ASIC) is adopted, which implements both encryption and Conditional Access (CA) control.

StemCerts: customizable X.509 v3 certificates for higher security, flexibility, and convenience

Giovanni Chiola and Paolo Gasti
DISI, University of Genoa
via Dodecaneso 35, 16146 Genoa, Italy

April 28, 2006

Extended Abstract

Despite the high degree of security that can be offered and the wide availability of high quality open source software implementations for the creation, the management, and the use of X.509 certificates, Public Key Infrastructures (PKIs) have not enjoyed the widespread diffusion that security experts had envisioned after the introduction of public key cryptography. We believe that there are many reasons for this lack of practical success, including a relatively high cost for obtaining high security grade certificates from external Certification Authorities (CAs), a complete inflexibility of the certificates with respect to changing/evolving user needs, and the practical need for on-line verification of revocation lists for higher security applications.

In order to address such weak points we introduce the notion of StemCerts that — in analogy with Stem Cells in Biology — offer many attractive features for certain important classes of applications. A particular X.509 certificate — the StemCert — signed by the CA, lets the user derive other certificates on his own, not requiring any further interaction with the Authority. These certificates can differ from the StemCert only in some parts, defined by the CA. So, every time a user needs to edit some fields of a certificate, he does not need to require a new one, if she derives it from a StemCert. For instance, the certificate owner might be allowed to generate new certificates with a customizable "name" field, and/or "expiration date" field, and/or "public key" field deriving them from the original StemCert, while keeping the validity of the CA's signature.

From the technical point of view the result is achieved by adopting Chameleon Hash functions instead of the usual cryptographic hash functions. By allowing the owner to change the "name" field, the CA grants the user the possibility of using pseudonyms instead of her real identity, thus ensuring anonymity in transactions involving public key cryptography such as payments in E-commerce applications. Allowing the owner to change her "public key" field in a certificate improves flexibility and security while reducing cost, by encouraging the user to change her key pair whenever needed, without recontacting (and paying) the CA. Allowing the owner to off-line change the "expiration date" may greatly improve the security of the mechanism and reduce the need for consulting revocation lists.

The X.509 certificate standard requires the use of revocation list when verifying a certificate, due to their relatively long lifetime. This is an expensive task and is often not implemented, thus both violating the standard and reducing the overall level of security. Our proposed solution allows the owner to keep her certificate almost always "elapsed," and to "renew it" for a very small interval of time just before its use. With this customization the certificate can virtually become a "one time certificate," which virtually eliminates the need for consulting revocation lists (as long as the Chameleon trapdoor that allows the owner the exclusive modification right is not compromised, of course).

In order to demonstrate the viability of StemCerts we implemented a proof-of-concept prototype that allows the creation, the successive customization, and the use of certificates in the context of a client-server web connection.

X.509 Certificates

Public key certificates adopt a digital signature to bind together a public key with an identity. They are generally used to verify that a public key belongs to an individual. X.509 [1] is as ITU-T standard for public key infrastructure. It specifies, among other things, standard formats for public key certificates and a certification path validation algorithm. In the X.509 system, a Certification Authority issues a certificate binding a public key to a particular *Distinguished Name* or to an *Alternative Name* [2] such as an e-mail address. An organization's trusted root certificates can be distributed to all employees so that they can use the company PKI system.

An X.509 certificate must contain some information which identifies the CA which released it, the period in which the certificate can be used, the algorithms used to calculate the signature, the version of the certificate, and a serial number, which uniquely identifies the certificate released by that particular authority.

The third version of X.509 certificates introduced a field called *Extensions*, which provide methods for associating additional attributes with users or public keys and for managing a certification hierarchy. In this way communities are allowed to define private extensions to carry information unique to those communities.

Chameleon Hash Functions

A Chameleon Hash function is associated with a pair of public and private keys (the latter generally called a *trapdoor*). Indeed, a Chameleon Hash function [3] is a trapdoor collision-resistant hash function: without knowledge of the trapdoor information, a Chameleon Hash function has the same characteristics of any cryptographic Hash function, such as pre-image and collision-resistance. However, collisions and second pre-images can be easily computed once the trapdoor is known.

A simple construction of a chameleon signature, presented in [3], employed as Hash function the Chaum-Pedersen trapdoor commitment. More precisely, a potential recipient chooses and publishes a regular discrete logarithm-based public key $y = g^x$, where g is the generator of a cyclic group G and x is the secret key. Then, a user who wishes to sign message m can compute the Chameleon hash value $h = y^m g^r$, where r is an auxiliary integer chosen uniformly at random by the signer. The message m must be a short binary message that has value smaller than the order of the group G when interpreted as the binary expansion of a non-negative integer. However, in order to extend the scheme to arbitrary length messages it is sufficient to first hash the long message using a regular, cryptographic hash function. Notice that if the recipient forges the signature, and two pairs (m, r) and (m', r') become known to the signer (during a dispute), the signer can recover the secret key x of the recipient from $h = g^m y^r = g^{m'} y^{r'}$, giving $x = \frac{m-m'}{r'-r}$.

This is a highly undesirable outcome, so we adopted the Chameleon Hash function presented in [4] instead. In [4] Ateniese and De Medeiros propose several schemes which provide better performance than the one presented in [3]. In particular, they adopted a scheme related to a twin Nyberg-Rueppel signature (signature introduced in [5]).

Customizable X.509 Certificates

Our scheme allows the user to modify some of the fields of the certificate in a limited and controlled fashion, without any further interaction with the CA after the certificate has been signed. The user can change only a subset of the fields contained in an X.509 v3 certificate, chosen by the CA. Only the legitimate owner can modify the information contained in the certificate, as long as she is the only one that knows the trapdoor associated with the Chameleon Hash used to generate the signature over the changeable fields.

The CA can choose which fields are editable, following a predefined policy: for example, if it signs a certificate which lets the user modify her *Distinguished Name* or her *Alternative Name*, it offers her the possibility to use the certificate anonymously.

In our prototype implementation we decided to let the owner change a field called "username", which is a free text field. It should contain the information needed to identify a user, such as her name, e-mail address, DNS name, and so on. We also allow the user the opportunity to edit the

"validity period" field. In this way, the certificate is considered not valid until the user decides to "enable" it.

The scheme used for the Chameleon Hash can be summarized as follows.

Key generation: choose a safe prime p of bit length k, and a generator g of the subgroup of quadratic residues Q_p of Z_p^*. The owner of the certificate chooses as secret key x at random in $[1, q-1]$, and his public key is computed as $(g, y = g^x)$.

Hashing scheme: To commit a value m, it is sufficient to choose random values $(r, s) \in Z_q \times Z_q$, and compute

$$e = H(m, r); \text{ and } \text{Hash}(m, r, s) = r - (y^e g^s \bmod p) \bmod q.$$

where H is a collision-resistant hash function, mapping arbitrary-length bit strings to strings of fixed length l.

Collision finding: Let C denote the output of the Chameleon Hash on input the triple (m, r, s). A collision (m', r', s') for a random m' can be found by computing r' and s' such that:

$$e' = H(m', r'); \text{ and } C = r' - (y^{e'} g^{s'} \bmod p) \bmod q.$$

The owner of the certificate chooses a random value m', a random value $k' \in [1, q-1]$, and computes

$$r' = C + (g^{k'} \bmod q,\ e' = H(m', r'), \text{ and } s' = k' - e'x \bmod q.$$

For further details about collision resistance, semantic security and absence of key exposure, please refer to [4]

The main goal of our prototype implementation is to keep the StemCert structure as similar as possible to the original X.509 v3 structure. This should provide a smooth transition between the current X.509 certificates and our proposed StemCerts. Therefore we decided to keep the standard X.509 v3 structure unchanged. To achieve this result, we added three new extensions: CHAM_KEY, which contains the public part of the private key; USERNAME, which specifies the user's identity; and VALIDITY, which defines a period in which the certificate can be used.

CHAM_KEY is composed by four fields: p, q, g, and y. They represent the values which defines the public part of the Chameleon key as previously described. These values are chosen by the user when she generates the certificate request, together with the value of x (also needed to calculate y) which must be known only to the user. Once the certificate is signed by the CA, nobody can modify the values contained in this extension.

USERNAME is utilized by the user to define his identity, independently from the certification authority. It consists of five fields: *id*, which is a random value (its use will be described later), *name*, which is an arbitrary string used to define user's identity, r and s, two random values used to calculate the Chameleon Hash as described in the previous section, and *hash*, the value of the Chameleon Hash calculated on *id*, *name*, r, and s. In other words, $hash = \text{Hash}(f(id, name), r, s)$ where f is a bijective function of the form $f : string \times string \rightarrow string$.

The structure of the VALIDITY extension is very similar to that of USERNAME. It contains five fields, four of which — *id*, r, s, and *hash* — have the same semantics of those in USERNAME, while the other one — *validity* — represents a period in which the certificate can be considered valid. This period is freely changeable by the owner.

The generation of a certificate request for a StemCert is quite similar to that for a standard X.509 certification request. The user needs to generate a Chameleon key of appropriate length, and then include it in the appropriate extension. After that, she can choose which of the changeable fields are to be included. The *id* and *username* (or *validity*) values contained in the USERNAME (resp. VALIDITY) extension included in the certification request must be set to a predefined value, in order to clarify that the certificate obtained just after signing the request should not be used. The value of the *hash* field of the two extensions must be set to a non-standard value, namely the hash value computed over the respective fields as described above.

Once the CA receives the certification request it examines it. Of course the CA is free to remove any of the changeable fields that possibly violate its predefined policy before signing the StemCert. The last check is done by the CA over the *id* value. If it is the predefined value, the StemCert can be signed.

The CA substitutes the values contained in the fields *id*, *r*, *s*, and *username* (or *validity*) for the USERNAME (VALIDITY) extension with standard values, and computes the signature value of the obtained request as for a standard X.509 v3 signature. In this way the structure of the certificate (the editable extensions contained), the value of the public Chameleon key and the values of the hash over the editable fields are signed and cannot be modified by anyone.

If the owner can change the username value, she can choose which string to include in the StemCert just before using it. She can use it in an anonymous way, hiding her identity, or in a privacy-aware way, publishing only the personal information that are strictly needed for the particular transaction that implies the use of the certificate. In either ways, she simply puts a new value in the username field, then calculates *r* and *s* such that the Chameleon hash function output is equal to the (immutable) hash value contained in the extension.

If the certificate contains two (or more) editable extensions, the *id* field of each of them must have the same value. Otherwise, an attacker who knew the values of two editable certificates released from the same user, could build a third certificate containing one extension from the first and one from the second. The *id* is a random number of adequate length (in our prototype implementation it is a 128 bit value).

If another user wants to verify such a certificate, he has to substitute the values contained in the editable extensions as done by the CA during the signing operation. Then he can calculate a standard signature value over the obtained certificate. If it is valid, he has still to verify that the Chameleon Hash value corresponds to the one calculated over the changeable fields, and whether the *id* value of all the extensions is the same.

When using a StemCert the level of security available to who verifies is very high. Even if the certificate is used anonymously, the CA can still know the identity of the owner of the StemCert, because the serial number on it is immutable. In case of dispute, a subject that received a StemCert can show it in Court to a judge who has the power to force the CA to reveal the true identity of the owner. In this way, whoever receives a StemCert can trust the user who sent it even if she uses a pseudonym.

Implementation and experimental measurements

Our prototype implementation incorporates OpenSSL 0.9.8a library routines for cryptographic functions and X.509 certificate handling. We added the appropriate extensions to OpenSSL code and modified the routines that generate and verify certificates according to what is described in the previous sections.

All tests were run under Linux on an Athlon XP 1600+ laptop. The key length for Chameleon signature and for signing the certificate is 1024 bit. The performance of the implemented algorithm were good, generally adding a negligible delay over the one for handling standard X.509 certificates, apart for the key generation for a key length of 1024 bit. Being a probabilistic algorithm, the measured key generation times where quite different for repeated measurements, ranging from two to about ten seconds on the test machine. Considering that this operation must be done only once for each certificate, it should not be a concern.

References

[1] Request For Comments 3280 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile (http://www.ietf.org/rfc/rfc3280.txt)

[2] ITU-T Recommendation X.500 (1993) — ISO/IEC 9594-1:1994, Information Technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services

[3] Krawczyk, H.., Rabin, T.: Chameleon signatures. In: Proceedings of NDSS 2000. (2000) 143-154

[4] Ateniese, G., de Medeiros, B., On the Key Exposure Problem in Chameleon Hashes

[5] Naccache, D., Pointcheval, D., and Stern, J.: Twin signatures: an alternative to the hash-and-sign paradigm.