Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 6 (2011) pp. 443–451

Adaptive Response System based on the Success Likelihood of Ongoing Attacks

Wael Kanoun[1,2], Nora Cuppens-Boulahia[2], Frédéric Cuppens[2], Samuel Dubus[1], Serge Papillon[1] and Antony Martin[1]

[1]Bell Labs, Alcatel-Lucent
route de Villejust, 91620 Nozay, France
wael.kanoun@alcatel-lucent.com

[2]Telecom Bretagne
2, rue de la chataigneraie, 35576 Cesson-Sévigné, France
nora.cuppens@telecom-bretagne.eu

Abstract: **Nowadays, response systems are used jointly with preventive measures, to ensure an enhanced security level for a given system. In particular, previous papers focus on balancing the cost of the response with the impact of the attack. However, even if an attack was detected, it may not be able to achieve its objective. In this paper, we present a novel attack response system, based on the assessment of the likelihood of success of attack objectives. First, the ongoing potential attacks are identified, and their success likelihood are calculated dynamically. The success likelihood depends mainly on the progress of the attack and the state of the monitored system. Second, candidate countermeasures are identified, and their effectiveness in reducing success likelihood are assessed. Finally, candidate countermeasures are prioritized with respect to their effectiveness.**
Keywords: response system, success likelihood, attack objective, dynamic Markov model.

I. Introduction

Current information systems are steadily growing in size and complexity. On the other hand, such systems are targeted by higher numbers of vicious attack attempts. Moreover, these attacks are growing in sophistication. In the past, an attack consists in executing a simple action (e.g. sending a malformed IP[1] packet). Today, a single attack consists in executing multiple and organized actions/steps in order to achieve an attack objective.

To counter these threats, intrusion response systems are often used with preventive measures to provide a higher security level of a monitored system. A response system launches the appropriate countermeasure(s) in order to stop the detected attack(s). In order to select the best countermeasure(s) a response system needs an efficient diagnosis to identify ongoing attacks. For the selection (and prioritization) procedure, several criteria should be considered: (i) identification of potential attack objectives, (ii) success likelihood assessment of potential attack objectives, (iii) impact assessment of the attack and the countermeasure. Ultimately, risk-awareness provides to the response system a comprehensive evaluation of the monitored system's state, in order to identify the most effective countermeasure: success likelihood and the impact of the attack, and the impact of the response, are all considered.

While existing response systems consider the cost (or impact) of the detected attacks to prioritize and launch the countermeasures, we adopt a different yet complementary approach which considers the Success Likelihood *SL* of the detected attacks. In this paper, we use the *SL* of ongoing attacks to present a novel response system. For simplicity, we consider in this paper that all the detected attacks have the same impact on the monitored system, and that the response system handles only known attacks. The *SL* is a relative logarithmic metric derived from the time needed to accomplish the ongoing attack [1]: it indicates how close the attacker is to achieve his objective(s). Using this metric, the proposed response system evaluates the effectiveness of each countermeasure in reducing the success likelihood for the detected attacks. Finally, the model prioritizes candidate countermeasures with respect to their effectiveness. This can be useful in the case: (i) when several responses which cannot be activated simultaneously, or (ii) when responses have a cost or side effects, or even (iii) when a single response is effective against several potential attacks. Therefore, the administrator has to select among several response the most 'urgent' and effective one(s). A total defensive-centric view is adopted: we do not aim to find the most likely intrusion objective sought by the attacker. In fact, "85% of breaches were the result of opportunist attacker" [2].

This paper is organized as follows. Section II shows how elementary attack are modeled, and how attack graphs are constructed. Section III presents how the *SL* of each potential attack is calculated. In Section IV, we propose a response model based on a real-time assessment of the likelihood of success for ongoing attacks, and the effectiveness of candidate countermeasures. In Section V, a VoIP use case of an enterprise environment with numerical results are presented to illustrate our proposal. Section VI discusses existing and related work.

[1]Internet Protocol

II. Attack Modeling

First, efficient response systems have to recognize first the ongoing attack(s) in order to respond properly. Attack graphs depicts the attack steps that were executed on the monitored system, and may even show potential future steps. Thus, we rely on attack graph generation techniques to monitor the attack progress. Attack graphs techniques has been extensively investigated, and several models were proposed in the last decade. In particular, the *semi-explicit* approach (e.g. [3, 4]) relies on the description of the pre/post-conditions, which represents the prerequisites and effects of the elementary attack actions. This approach then finds causal relationships between these elementary actions and connects them when such a relationship exists. The correlation procedure then consists in building a scenario that corresponds to an attack graph. The semi-explicit approach is generic and flexible because only elementary steps are specified. In other words, administrators are not required to specify every potential attack scenario. Several attack languages may be used to specify the elementary attacks (e.g. LAMBDA [5], JIGSAW [6] and CAML [7]). Since we successfully used LAMBDA (LAnguage to Model a dataBase for Detection Attacks) with the *semi-explicit* [4] correlation during previous work, it will be retained in the remainder of this paper. However, the proposed response system can be used with other attack graph models.

A. LAMBDA Language

We present below a short description of LAMBDA used to describe elementary attack steps. For a formal and complete description, interested readers can refer to [5]:

- *pre-condition*: it describes the information system state required so that the attacker performs the step. It contains one or several logical predicates.
- *post-condition*: it describes the information system state after the execution of the step. It contains one or several logical predicates.
- *sk*: introduced in [8], it indicates the minimum level of skill and/or internal knowledge required to execute the step successfully. In this paper we consider that $0 < sk < 1$, and that step *A* is 'easier' than *B* if $sk_A > sk_B$.
- *detection*: it is used to map the LAMBDA attack model to the appropriate alert signature(s).

For example (see Figure 1), the elementary attack *sip_malformed_packet* on the machine *H2* can be executed successfully only if (i) the attacker *A* can access to *H2*, (ii) *H2* is on and vulnerable, (iii) the attacker knows that user is registered as *Sipext1*. Moreover, the crash of the machine *H2* is the consequence of this elementary attack.

B. Semi-Explicit Correlation

We say that two LAMBDA models *A* and *B* are correlated if the postcondition of *A* matches the precondition of *B*. Thus, it provides a precise diagnosis of the ongoing intrusion scenario by constructing the attack graph [4]; and predicts potential future steps and attack objectives [9]. An example is shown in Figure 1: If an attacker launches a *sip_user_discovery*, he (or she) will discover (i) that the victim registered, and (ii) that the victim is using machine *H2*. Knowing that, the attacker may send malformed crafted packets to crash the victim's machine. Thus, *sip_user_discovery* is correlated with *sip_malformed_packet* by matching the two predicates *is_on(H2)* and *Knows(A, useraccess(Siptext1, H1, udp, user))*.

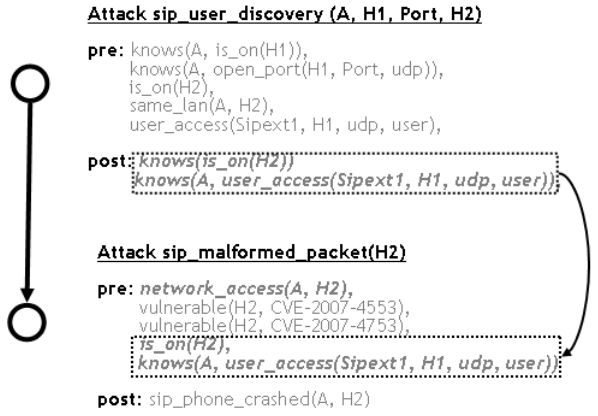


Figure. 1: Example of semi-explicit correlation

Using *semi-explicit* correlation, attack graphs are constructed from generated alerts, using LAMBDA models. The alerts generated by the Intrusion Detection Systems (IDS) are first aggregated and regrouped into meta-alerts. For each meta-alert, the associated elementary attack (specified with LAMBDA language) is instantiated. Moreover, potential future steps and candidate attack objectives are identified with the *semi-explicit* correlation. Thus, the non-detection of some attack steps will not cripple the response system. Moreover, as soon as one of the following steps is detected, the non-detection of a previous step will have no effect on the response system's effectiveness.
On the other hand, since attack graphs are constructed using pre-specified LAMBDA models, our response system can handle only known attacks. *Zero-day* attacks cannot be represented in the attack graphs for one of the following reasons: (i) the 'new' attack signature is not yet associated to one of the existing LAMBDA model, or (ii) the LAMBDA model representing the 'new' attack is not yet specified. We view *zero-day* attacks as an issue related to the detection process, and not to the response process.

III. Assessment of the Success Likelihood of Ongoing Attacks

A node in the attack graph represents an elementary attack that were executed and observed successfully, or a potential step that can be executed in the future (i.e. not yet observed). These nodes lead to the *attack objectives*, which constitute the terminal nodes in the attack graph. For each evolution of the attack or system state, a new attack graph is instantiated. This can be due to a new observed attack step: a future step in the previous graph turns to be executed in the new instantiated graph if the appropriate alert(s) was raised. Additionally, a new attack graph can be also instantiated if a predicate state of a future step changes (e.g. from *true* to *false* or vice

versa); which switches the concerned step state (executable or unexecutable). Therefore, the model will be applied for each instance of the attack graph. We can summarize the procedure to the following phases (see [1] for more information):

1. decompose the attack graph to several subgraphs (i.e. one subgraph for each attack objective),

2. transform each subgraph into a dynamic Markov model [10], and calculate the *SL* metric,

A. Decomposing the Attack Graph

First, the generated attack graph is decomposed into several subgraphs; each subgraph is associated to an attack objective. For instance, an attack graph with n attack objectives (i.e. terminal nodes) is decomposed into n subgraphs. In result, each subgraph contains all the future (i.e. not yet observed) nodes which lead to the associated attack objective, and also contains the already observed steps adjacent to the future steps. Figure 2 is an example of an attack graph with three attack objectives, with its decomposition into three subgraphs.

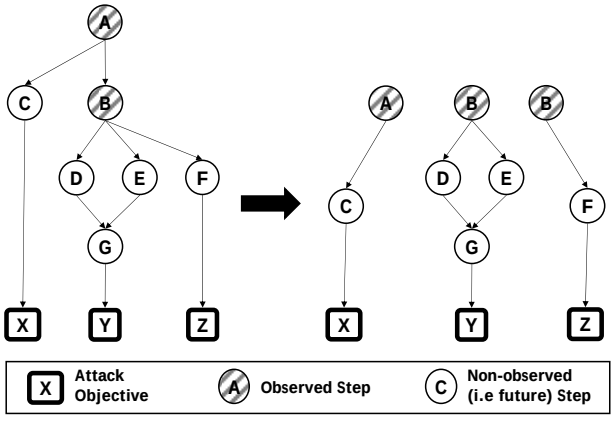


Figure. 2: Decomposition of an attack graph into subgraphs

B. Instantiating Markov Models and Assessing the SL

Each subgraph is transformed into a dynamic Markov Model that considers the progress of the ongoing attack(s) and the evolution of the monitored system state. The *transition probabilities* and *sojourn mean time* for each step in the subgraph are calculated. Thus, the *transition matrix* and *exit rate matrix* are instantiated for the Markov model associated to the subgraph. Interested readers may refer to [1] for more details. Markov Model was chosen because it adds to the attack graphs a 'temporal' dimension, which is needed to calculate the *SL*. This is exactly the same principle used in cryptography: greater the time needed to decipher an encrypted message, lower is the success likelihood to obtain the plain message.
For each Markov model, the Mean Time to attack objectives *MTAO* is calculated. Finally, for each candidate attack objective X, we calculate its success likelihood SL_x. We use the logarithmic formula proposed in [1], similar to the one used to express the magnitude of a physical quantity (current, voltage, power, etc.). The success likelihood depicts the variations of the *MTAO* metric:

$$SL_x = -20 \times \log_{10}\left(\frac{MTAO_X - MTAO_{min}}{MTAO_X}\right) \quad (1)$$

The success likelihood $SL_X = f(MTAO_X)$ of an attack objective X increases rapidly if $MTAO_X$ decreases, and $SL_X \to 0$ if $MTAO_X \to \infty$. Thus, if the attacker is closer to attack objective X, SL_X grows exponentially. Ultimately, if the attacker achieves the attack objective, we will have $SL_X \to \infty$.

IV. Response System Based on the Success Likelihood Metric

This section presents a response model based on real-time assessment of the *SL* for the ongoing attacks. The model takes in consideration the real-time evolution of both the attack and the information system. An evolution could be the result of a new executed and detected attack step, or the modification of a precondition of a future attack step in the scenario. First, candidate countermeasures are identified. Second, each candidate countermeasure will be simulated, and *SL* values will be re-calculated. Finally, the candidate countermeasures are prioritized w.r.t. their *SL* mitigation effectiveness.

A. Identifying Candidate Countermeasures

The anti-correlation approach [11] allows to identify the candidate responses along with the scalability consideration: we do not need to statically associate each countermeasure to one or several attacks. First, all the responses are modeled with LAMBDA. Then, the association is performed dynamically using anti-correlation: A countermeasure *C* is anti-correlated with an attack *A* if the postcondition of *C* matches the precondition negation of *A*. The anti-correlation approach is based upon finding the appropriate countermeasures that turn elementary future steps unexecutable, due to precondition(s) modification. Therefore, the response system can identify, from a predefined library, the countermeasures which are capable of blocking an ongoing attack. An example is shown in Figure 3: countermeasure *drop_sip_traffic* is capable of blockong the attack *sip_user_discovery* by transforming the precondition predicate *network_access(A,H2)* to *false*.

B. Simulating the Activation of Countermeasures and Recalculating the SL

Candidate countermeasures are identified using the anti-correlation approach to block future attack steps. As a result, these countermeasures reduce the *SL* of one or several attack objectives. Therefore, for a given instance of the attack graph, each candidate countermeasure will be simulated, and new values of the success likelihood for the attack objectives will be calculated. Thus, the effectiveness of a given countermeasure in reducing the *SL* of attack objectives can be assessed, and compared to other countermeasures.

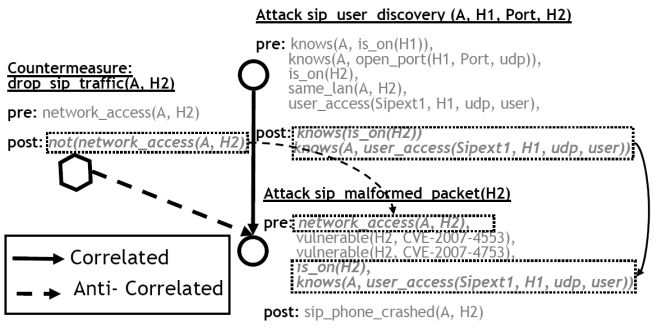


Figure. 3: Example of semi-explicit correlation and anti-correlation

When simulating of the activation of a given countermeasure CM_V, the same procedure as described in Section III is applied. The main difference is that the attack steps anti-correlated with CM_V are considered as blocked. In other words, the time needed by the attacker to execute successfully these attack steps will be very high and almost infinite. Consequently, a high mean sojourn time is assigned to the attack steps blocked by countermeasure CM_V. In other words, a low value (e.g. 10^{-3} in this paper) will be assigned to the Markovian parameter *exit rate* of these attack steps.
For the K^{th} instance of the attack graph, with the countermeasure CM_V activated, we denote the new value of the success likelihood for the attack objective X by $SL_{K,V,X}$. Moreover, we denote by $\overrightarrow{SL}_{k=K,v=V}$ the vector that contains sorted (descending order) *SL* values of all the attack objectives, during the step $k = K$ of the attack progress, while the countermeasure $CM_{v=V}$ is activated:

$$\overrightarrow{SL}_{K,V} = [SL_{K,V,1}, \quad SL_{K,V,2}, \quad SL_{K,V,3}, \quad \cdots] \qquad (2)$$

C. Prioritizing Candidate Countermeasures

The objective of this phase is the prioritization of candidate countermeasures with respect to their mitigation effectiveness of the *SL* of candidate attack objectives. During the K^{th} attack step, and for each candidate countermeasure CM_V the success likelihood vector $\overrightarrow{SL}_{K,V}$ that contains the *SL* of the candidate attack objectives is calculated. During the K^{th} step in the attack progress, we say that countermeasure CM_V has a higher priority than $CM_{V'}$ if $\overrightarrow{SL}_{K,V} < \overrightarrow{SL}_{K,V'}$; where the $<$ operator is a lexicographic comparison.

Proposition 1 If $\overrightarrow{SL}_{K,V} < \overrightarrow{SL}_{K,V'}$ then $CM_V >_{priority} CM_{V'}$ at K^{th} step of the ongoing attack.

CM_V has a higher priority because it reduces more significantly the *SL* of candidate attack objectives than $CM_{V'}$. The descending order of $\overrightarrow{SL}_{k=K,v=V}$ ensures that the most 'urgent' attack objectives are considered first, and their *SL* are reduced. Therefore, candidate countermeasures are prioritized and sent to the administrator or to the response management module. The prioritization can be also useful to determine which countermeasure should be launched first, when several countermeasures cannot be activated simultaneously.

V. VoIP Use Case

The case study is a SIP[2]-based VoIP enterprise service. The VoIP service (see Figure 4) is composed of a SIP server on a dedicated network; which acts as a SIP registrar for the HTTP Digest authentication, and as a SIP router/proxy for call routing. OpenSER [12] is used as the SIP server, while the authentication is delegated to a collocated RADIUS server, based on FreeRADIUS [13]. There are three SIP User Agents (UA) networks: the first for softphones (i.e. X-Lite [14], S-JPhone'[15] and Linphone [16]), and the second for hardphones (i.e. Thomson, Linksys, Zyxel) which was divided into wired and wireless networks. The intrusion detection infrastructure relies on Snort [17]. For the *Alert Collection and Correlation Engine* module, we use the CRIM prototype [18] that (i) aggregates the collected alerts and (ii) generates a pre/post-condition graph adopting the semi-explicit approach. Moreover, CRIM identifies candidate countermeasures using anti-correlation. Finally, a Matlab [19] module calculates the *SL* of each attack objective in the attack graph, and prioritizes the candidate countermeasures.

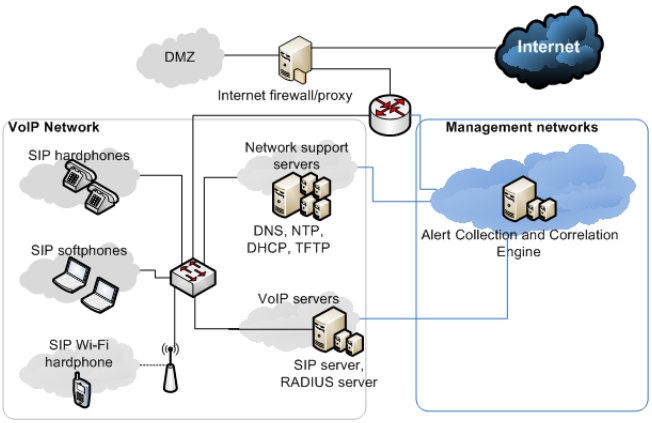


Figure. 4: VoIP testbed

In order to demonstrate our work, we implemented a set of elementary attacks. Both SIP related attacks, based on flaws in the protocol design [20] and flaws in software implementation, were identified and implemented in the VoIP testbed. On the other hand, six attack objectives which violates the operation and security policy were specified (e.g. SIP server DDoS, user highjacking, injecting audio traffic, SPIT, etc.). Moreover, candidate countermeasures were implemented using various shell script languages. Eight countermeasures are available for the system (e.g. blocking the traffic between the attacker and the server (or the user), changing the user's credentials, encrypting the media traffic, etc.). The number of LAMBDA models (i.e. elementary attack steps and objectives) used in the attack graph is thirty one. A correlation engine, using the semi-explicit approach, generates the attack graph (see Figure 5). The attack graph can be divided into two parts: during the first part, the attacker sends spam mail with a malicious link to infect potential victims in the enterprise network: it is the remote-to-local part of the attack. In the following scenario, three machines in the enterprise

[2]Session Initiation Protocol

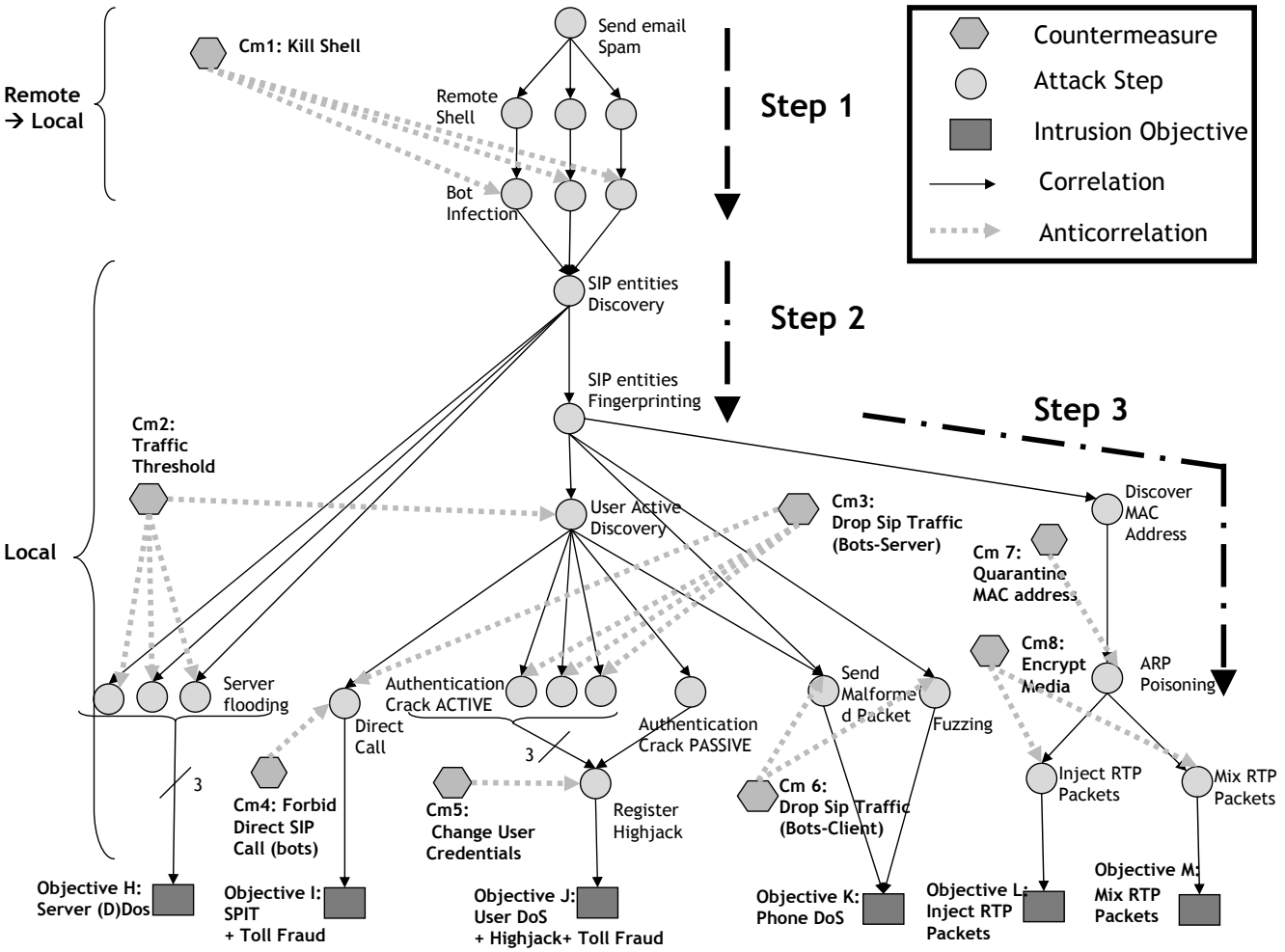


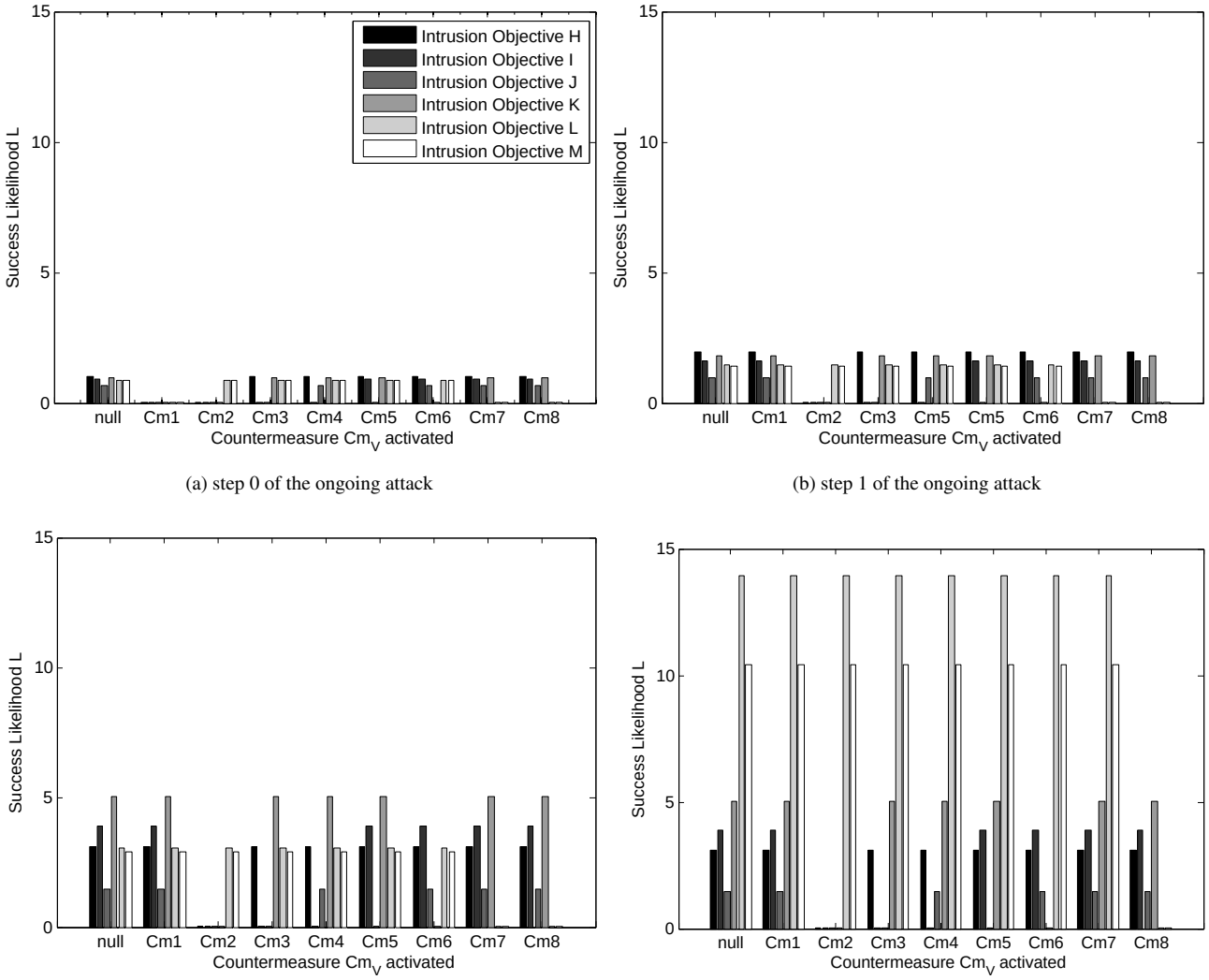
Figure. 5: Attack graph of the VoIP use case

network are infected with a bot. In the second part, the attacker being 'inside', is now able to perform several types of elementary actions to achieve one of the attack objectives.

Step 0 of the ongoing attack The attacker did not execute any attack yet. Having six attack objectives, the attack graph is decomposed into six subgraphs. Then, the *SL* for each attack objective is calculated. Figure 6a shows the *SL* of the attack objectives (H, I, J, K, L and M), considering the eight candidate countermeasures. The *SL* of all the attack objectives have relatively low values because the attacker did not yet execute successfully any attack step. It is obvious that CM_1 has the highest priority because it is able to stop all (future) candidate attacks. On the other hand, other candidate countermeasures can block some, but not all, candidate attack objectives.

Step 1 of the ongoing attack The attacker gains a remote shell and successfully infects three internal machines. Obliviously, this affects the *SL* of all candidate attack objectives. Figure 6b shows the *SL* of the attack objectives, considering the eight candidate countermeasures at step 1. We notice that the *SL* of all attack objectives increased. Since the machines are now infected with bots, the countermeasure CM_1 (kill remote shells) is no more effective. As in Step 0, the highest priority is for CM_2 because it is capable of blocking four objectives with the highest *SL*. On the other hand, CM_3, CM_7 and CM_8 can block two attack objectives. Finally CM_4, CM_5 and CM_6 can block only one attack objective.

Step 2 of the ongoing attack The response system should stop the ongoing attack as early as possible. However, the response system might not activate a countermeasure for several reasons: the first steps of the attack were not properly detected (e.g. false positive, false negative, etc.), or the administrator did not launch the appropriate countermeasures because they cost too much, or because it is too late to launch the candidate countermeasure (e.g. killing a remote shell after the bot infection of a machine does not stop the ongoing attack). We consider in this step that the attacker proceeds and launches an *active user discovery* attack with *SIP entities fingerprinting* attack. We notice that the *SL* of all the attack objectives increased (see Figure 6c). We can also note that the *SL* of attack objective K rose dramatically; this can be explained by the fact that the attacker has only one remaining step (i.e. sending malformed packet) to cause a Phone DoS. Therefore at this step, CM_2 has the highest priority because it is capable of stopping attack objective K (and also H, I and J), which has the highest *SL* (i.e. the most 'urgent'). CM_6 has the second highest priority because it also can stop attack objective K.



(a) step 0 of the ongoing attack

(b) step 1 of the ongoing attack

(c) step 2 of the ongoing attack

(d) step 3 of the ongoing attack

Figure. 6: *SL* of the attack objectives, w.r.t. attack's progress and activated countermeasure

Step 3 of the ongoing attack The attacker performs a *MAC address discovery* and *ARP poisoning*. After re-evaluation, Figure 6d shows that the *SL* of attack objectives L and M increased dramatically. At this step, CM_8 (i.e. Encrypting RTP Media Traffic) has the highest priority, because it is the only candidate countermeasure capable of blocking these two attack objectives (i.e. L and M).

For each evolution of the attach graph, the administrator or the response system is supported with a prioritized list of candidate countermeasures. This prioritization allows the administrator or the response system to launch the most effective and 'urgent' countermeasures first, which is useful in case of countermeasures that cannot be activated simultaneously.

Normally, the attack must be stopped as soon as possible (e.g. kill the shells or disinfect victim machines during step 1). Since intrusion detection systems are not perfect, the detection of one or several attack steps can be missed. For instance step 1 may be executed undetected, and thus CM_1 was not activated. Hence, since the machines were successfully infected, closing the remote shells becomes an obsolete countermeasure. That is why the response system will re-prioritize the countermeasures at each time a new attack step is detected. Moreover, a given countermeasure cannot be executed due to system constraints and limitations (e.g. the enforcement of the countermeasure fails, or the impact of the countermeasure is too high, or another opposite countermeasure is already active, etc.). Therefore, even if the attack was detected, a quick response may not be feasible. However, the proposed response system can handle such cases perfectly, by re-prioritizing at each time the candidate countermeasures.

VI. Related Work

Recently, several intelligent intrusion response systems were proposed. Toth and Kruegel [21] proposed a cost sensitive approach that balances between intrusion damage and response cost in order to choose a response with the least impact. Lee *et al.* [22] also discuss the need to consider the cost of intrusions damage, the cost of manual and automated response to an intrusion, and the operation cost, which measures constraints on time and computing resources. Similar approaches were also proposed in [23], [24] and [25]. A gen-

eral framework for advanced response systems based on risk analysis approach is defined in [26]; where likelihood of success and impact are combined to calculate the risk of detected attacks.

The generation of attack graphs generation has been an active research field in the last decade. During the MIRADOR project, Cuppens *et al.* presented in [4] the semi-explicit approach to correlate elementary attacks described using LAMBDA [5] . In [3], Ning *et al.* combined complementary types of alert correlation methods: (i) those based on the similarity between alert attributes; and (ii) those based on prerequisites and consequences of attacks. The work is very close to Cuppens and Miège's work which was done independently and in parallel. Similar models were presented in [6] and [7]. In [27], Sheyner *et al.* used a model of exploits (possible attacks) in terms of their preconditions and postconditions to construct possible sequences of attacks. By contrast, our method constructs high-level attack scenarios from low-level intrusion alerts, and reasons about attacks possibly missed by the IDS. While the previous vulnerability analysis techniques are focused on analyzing what attacks *may happen* to a given system, our approach constructs what *is happening* to a given information system according to the alerts reported by IDS.

[1] presents how to calculate the success likelihood of candidate attacks using generated attack graphs. Therefore, candidate attack objectives can be prioritized. However, [1] considers that each countermeasure may affect a single attack objective. Thus the prioritization of candidate attack objectives was equivalent to the response prioritization which is obviously not always true.

Madan *et al.* in [28] proposed a general framework to assess the MMTSF (Mean Time to Security Failure) using a Markov Modeling approach. The main drawback of this framework is that it does not specify how to calculate transitions rates, neither how to model atomic attack actions and relation between these actions. Moreover, this framework does not take into consideration neither the dynamic nature of an ongoing attack, nor the real-time state of the monitored system.

In [29], McQueen *et al.* proposed to calculate the Risk Reduction due to installing or modifying security measures (e.g. updates, firewalls, etc.). The calculation is based on the existing vulnerabilities in the system. In this approach, an attack graph is composed of nodes that represent the attack stages. The edges are associated with time to compromise, which is calculated in function of the number and types of vulnerabilities. This paper does not discuss intrusion response systems. Furthermore, since this paper presents an offline analysis model, it does not consider the real-time nature of the monitored system.

VII. Conclusion

In this paper, a novel response system based on a real-time assessment of the success likelihood for the ongoing attacks, is presented. This model takes in consideration the state of the attack progress and the monitored system state. The *SL* metric calculated dynamically can be relevant for the administrators, and helps them to prioritize and handle the ongoing attacks. Our model can also offer valuable input for intelligent and automated response systems, which may be risk-aware or cost-sensitive. Moreover, our model can help to prioritize and launch countermeasures that cannot be activated simultaneously. Finally, the proposed model was successfully validated in a VoIP use case using complex attack scenarios that violate operation and security policies. However, the prioritization considered only the *SL* of the potential attack objectives. Therefore, we see that our model have to be combined with cost-sensitive models to take in consideration the impact of the attacks and the reactions. In other words, the response system has to consider the risks (i.e. the success likelihood and the impact) of ongoing attacks and the impact of candidate countermeasures. In the future, the effectiveness of our model to select the best countermeasure(s) will be explored by combining the Likelihood and the Impact (thus assessing the real-time risk) of a given attack. We will explore also the use of Hidden Markov Models in the *SL* assessment model, to take into consideration the uncertainty of the attack progress and the monitored system state.

References

[1] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, S. Dubus, and A. Martin, "Success likelihood of ongoing attacks for intrusion detection and response systems," in *International Conference on Computational Science and Engineering, 2009 (CSE '09)*, Vancouver, August 2009.

[2] Verizon Risk Business Team, "2008 Data Breach Investigations Report."

[3] P. Ning, D. Xu, C. G. Healey, and R. S. Amant, "Building attack scenarios through integration of complementary alert correlation methods," in *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS04)*, 2004, pp. 97–111.

[4] F. Cuppens and A. Miège, "Alert correlation in a cooperative intrusion detection framework," *Security and Privacy, IEEE Symposium on*, vol. 0, p. 202, 2002.

[5] F. Cuppens and R. Ortalo, "Lambda: A language to model a database for detection of attacks," in *Third International Workshop on Recent Advances in Intrusion Detection (RAID'00)*, Toulouse, France, 2000.

[6] S. J. Templeton and K. Levitt, "A requires/provides model for computer attacks," in *NSPW '00: Proceedings of the 2000 workshop on New security paradigms*. New York, NY, USA: ACM, 2000, pp. 31–38.

[7] S. Cheung, U. Lindqvist, and M. W. Fong, "Modeling multistep cyber attacks for scenario recognition," *DARPA Information Survivability Conference and Exposition,*, vol. 1, p. 284, 2003.

[8] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, and J. Araujo, "Automated reaction based on risk analysis and attackers skills in intrusion detection systems," in *Third International Conference on Risks and Security of Internet and Systems, 2008 (CRiSIS '08)*, Oct. 2008.

[9] F. Cuppens, F. Autrel, and A. M. et S. Benferhat, "Recognizing malicious intention in an intrusion detection

process," in *Second International Conference on Hybrid Intelligent Systems*, Santiago, December 2002.

[10] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. Wiley, January 1968, vol. 1.

[11] F. Cuppens, F. Autrel, Y. Bouzida, J. Garcia, S. Gombault, and T. Sans, *Anti-correlation as a criterion to select appropriate counter-measures in an intrusion detection framework*, January 2006, vol. 61, no. 1-2, ch. Annals of Telecommunications.

[12] "The official website of OpenSER: http://www.opensips.org/."

[13] "The official website of FreeRADIUS: http://freeradius.org/."

[14] "The official website of X-Lite: http://www.counterpath.net/X-Lite-Download.html."

[15] "The official website of SJphone: http://www.sjlabs.com/sjp.html."

[16] "The official website Lindphone: http://www.linphone.org/."

[17] "The official website of snort: www.snort.org."

[18] F. Autrel and F. Cuppens, *CRIM : un module de corrélation d'alertes et de réaction aux attaques*, September-October 2006, vol. 61, no. 9-10, ch. Annals of Telecommunications.

[19] "The offcial website of MATrix LAboratory (MATLAB): http://www.mathworks.com/products/matlab/."

[20] D. Endler and M. Collier, *Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions*. McGraw-Hill Osborne Media, 2006.

[21] T. Toth and C. Kruegel, "Evaluating the impact of automated intrusion response mechanisms," in *18th Annual Computer Security Applications Conference ACSAC02*, 2002.

[22] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response," *Journal of Computer Security*, vol. 10, no. 1/2, pp. 5–22, 2002.

[23] N. Stakhanova, S. Basu, and J. Wong, "A cost-sensitive model for preemptive intrusion response systems," in *21st International Conference on Advanced Information Networking and Applications (AINA'07)*, May 2007, pp. 428–435.

[24] H. Wei, D. Frinke, O. Carter, and C. Ritter, "Cost-benefit analysis for network intrusion detection systems," in *28th Annual Computer Security Conference (CSI'01)*, October 2001.

[25] N. Kheir, H. Debar, N. Cuppens-Boulahia, F. Cuppens, and J. Viinikka, "Cost evaluation for intrusion response using dependency graphs," in *IFIP International Conference on Network and Service Security (N2S'09)*, June 2009.

[26] W. Kanoun, N. Cuppens-Boulahia, and F. Cuppens, "Advanced reaction using risk assessment in intrusion detection systems," in *Second International Workshop on Critical Information Infrastructures Security (CRITIS07)*, Springer, Ed., Malaga, Spain, 2007.

[27] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing, "Automated generation and analysis of attack graphs," in *Proceedings of IEEE Symposium on Security and Privacy*, 2002.

[28] B. B. Madan, K. Goševa-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "A method for modeling and quantifying the security attributes of intrusion tolerant systems," *Perform. Eval.*, vol. 56, no. 1-4, 2004.

[29] M. A. McQueen, W. F. Boyer, M. A. Flynn, and G. A. Beitel, "Quantitative cyber risk reduction estimation methodology for a small scada control system," in *HICSS '06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, 2006.

Authors Biographies

Wael Kanoun is a Member of Technical Staff (MTS) in the Security Department at Bell Labs, Alcatel-Lucent. He holds a PhD degree in Networks and Systems Security from Telecom Bretagne, one of the most prestigious Engineering Schools in France. He conducted his PhD work as a full time R&D engineer at Bell Labs, which allowed him to acquire a solid expertise and knowledge in systems and networks security. Dr. Kanoun's professional interests are in risk management, security and response policies, intelligent supervision and response systems for large and critical infrastructure (Telecom, Smart Grids, etc.). His work has been published and presented in international conferences, or filed as international patents. He also holds a Master degree in "Systems and Networks Architecture", from Telecom Bretagne, and a Telecom and Electrical Engineering Diploma from the Lebanese University.

Nora Cuppens-Boulahia is a teacher/researcher at the TELECOM Bretagne LUSSI department. She holds an engineering degree in computer science and a PhD from SupAero and an HDR from University Rennes 1. Her research interest includes formalization of security properties and policies, cryptographic protocol analysis, formal validation of security properties and thread and reaction risk assessment. She has published more than 60 technical papers in refereed journals and conference proceedings. She has been member of several international program committees in information security system domain and the Programme Committee Chair of Setop 2008, Setop2009, SAR-SSI 2008, CRiSIS 2010 and the co-general chair of ESORICS 2009. She is the French representative of IFIP TC11 "Information Security" and she is he co-responsible of the information system security axis of SEE.

Frédéric Cuppens is a full professor at the Telecom Bretagne LUSSI department. He holds an engineering degree in

computer science, a PhD and an HDR. He has been working for more 20 years on various topics of computer security including definition of formal models of security policies, access control to network and information systems, intrusion detection, reaction and counter-measures, and formal techniques to refine security policies and prove security properties. He has published more than 150 technical papers in refereed journals and conference proceedings. He served on several conference program committees and was the Programme Committee Chair of ESORICS 2000, IFIP SEC 2004, of SARSSI 2006 and general chair of ESORICS 2009.

Samuel Dubus is a Member of Technical Staff (MTS) in the Security Department at Bell Labs, Alcatel-Lucent. He has a solid experience working for National and European research projects in various domains of communication security, and he currently coordinates the Eureka/CELTIC BUGYO Beyond collaborative research project (CP05-003). He also has an extensive experience working as a security consultant. His interests in the security of communication systems are mainly focused on threat management, security assurance, alert correlation, intrusion detection and reaction. He holds an engineering degree of the Ecole Polytechnique des Ingnieur de Lille (Polythech'Lille, 1997).

Serge Papillon is a Member of Technical Staff (MTS) in the Security Department at Bell Labs, Alcatel-Lucent. He holds an Engineering Diploma from the prestigious French School "Ecole Centrale de Lyon", specializing in Computer Science. He joined the research group of Alcatel-Lucent in 1997, being responsible for improving the designs and architectures of high availability real-time systems. In 2002, M. Papillon was among the founders of the Security Research Department. He participated actively in several national and international research projects. His professional interests are in risk management methodologies and tools, security assurance, access control and authentication. He holds eight French and international patents.

Antony Martin is a Member of Technical Staff (MTS) in the Security Department at Bell Labs, Alcatel-Lucent. He is interested in a variety of areas including Network Security and VoIP Security. He holds a CCNA (Cisco Certified Network Associate) Security Certification, and a Snort Professional Certification. He holds an Engineering Diploma from Telecom Lille 1, France.