2011 Sixth International Conference on Availability, Reliability and Security

Using CVSS in attack graphs

Laurent Gallon
LIUPPA
University of Pau
Mont de Marsan, France
Email: laurent.gallon@univ-pau.fr

Jean-Jacques Bascou
LIUPPA
University of Pau
Mont de Marsan, France
Email: bascou@univ-pau.fr

***Abstract*—Derived from attack models, attack graphs are providing an efficient way to model attack scenarios intended against computer networks. Such graphs are using CVE database in which all known vulnerabilities are gathered. The CVSS framework is aiming to give numeric scores to each vulnerability recorded in the CVE database, which represent its characteristics and quantify its security impacts. In this paper we adapt attack graphs definition in order to be able to use them in conjunction with CVSS framework. The aim of our work is to provide a way to give an assessment of the impact of attacks on the hosts of the target network. This assessment is made using a host damage score and a network damage score, which take into account the characteristics and consequences of each atomic attack constituting an attack scenario.**

***Keywords*-IT vulnerabilities, attack graphs, CVSS framework**

I. Introduction

One of the main objectives of computer security is to defend against network attacks. When an attacker wants to reach an objective, as gaining root privileges on a critical server, it is rare that he exploits a single vulnerability on a single target host. In most of the cases, he uses several single attacks (we call them *atomic attacks*) on several hosts, in order to progress towards this objective. Consequently, it is quite important for administrators to be able to know and to understand attack mechanisms that could be used by attackers on their networks.

In attack graphs [1][5][6][8], succession of possible atomic attacks (we call them *complex attacks* or *multi-stage attacks*) are represented as a graph, in which the initial state represents the safe state, and deadlock states represent states in which the attacker has reached his objective. Each edge of the graph models an atomic attack. A particular path from the initial state to a deadlock state represents one possible complex attack which leads the attacker to a success.

One of the main drawback of attack graphs is that they don't give any information on the severity level of complex attacks they model. So it is difficult to assess the damages caused by these attacks on the network, and on each of its hosts.

CVSS [2][3][4][10][11] is a framework which assess the severity of IT vulnerabilities, giving a severity score to each of it. This score is computed using three categories of metrics, which assess the intrinsic characteristics of the vulnerabilities (base metrics), its evolution over time (temporal metrics), and the user environment in which the vulnerability is detected (environmental metrics). In this paper, we use CVSS in attack graphs, in order to assess the damages of complex attacks, taking into account correlation between successive atomic attacks.

Note that the idea of using CVSS for attacker behavior modelling is not completely new. For example, in [12], the authors propose a framework for security requirements elicitation and analysis centered on vulnerabilities. They argue that CVSS is a good method for evaluating the criticality of vulnerabilities. They also notice that attack trees or attacks graphs can be used to build attacker templates. But they do not give any key on how to combine CVSS and attack graphs. In [13], the authors propose an attack graph-based probabilistic metric. This metric is compared to values given in CVSS, but they don't conclude on how to merge these metrics. Our main contribution is to go further than these previous works on how to use CVSS in attack graphs.

The rest of the paper is structured as follows: in section II and section III, we briefly present the CVSS framework and the attack graph formalism. In section IV, we explain how to use CVSS in attack graphs. We illustrate our proposal through an example in section V. Finally, section VI concludes this paper and gives some perspectives.

II. CVSS framework

CVSS (Common Vulnerability Scoring Systems) [10] provides an open framework which assess the severity level of IT vulnerabilities. It associates a severity score (CVSS score) to each IT vulnerability, which ranges from 0.0 to 10.0 . It is computed by the use of metrics, which result from three vectors:

- the base metrics vector represents the intrinsic characteristics of the vulnerability. Six different metrics can be found in this vector, which can be divided into two sub-groups:
  - the exploitability level needed by the attacker in order to be able to launch the attack. In this sub-group, we find three metrics: *access vector AV* (the

978-0-7695-4485-4/11 $26.00 © 2011 IEEE
DOI 10.1109/ARES.2011.18

type of access you need to exploit the vulnerability); *access complexity AC* (the complexity for a hacker to exploit the vulnerability); *authentication AU* (the level of authentication needed to exploit the vulnerability). An *Exploitability Score (ES)* is computed using AC, AV and AU.
  - damages caused by the attack on the target host. In this sub-group, we find three metrics, representing the impact of the attack on the three classical security properties: *confidentiality impact CI*; *integrity impact II*; *availability impact AI*. An *Impact Score (IS)* is computed using CI, II and AI.

  The CVSS *Base Score (BS)*, which assesses the dangerousness of the vulnerability, is computed using both ES and IS. Formulas for BS, ES and IS can be found in [10]. The assessment of Base metrics and Base score of all known IT vulnerabilities can be found in most vulnerability databases, like NVD [2].
- the temporal metrics vector represents the characteristics of vulnerabilities which change over time, but not through the user environment. This vector is optional, that is we can use CVSS framework even if this vector is not given. Three metrics are defined: *exploitability E* (the level of exploitability of the vulnerability: code and exploit techniques available, ...); *remediation level RL* (the level of remediation for this vulnerability, patches availability, ...); *report confidence RC* (the level of confidence we can have on the publication of this vulnerability: vulnerability reported by several organisms, vulnerability exploitation reports, ...). A *temporal score (TS)* is computed using the base score, E, RL and RC.
- the environmental metrics vector represents the characteristics of the vulnerability which are user dependent. This vector is optional. Two sub-groups of metrics can be defined:
  - the impact on the user environment. Two metrics are defined: *collateral damage potential CDP* (loss of life, physical damages, ...); *target distribution TD* (measurement of the potential propagation of the vulnerability in the user environment).
  - modification of security impact (base metrics) depending on the importance of security properties for the user: *confidentiality requirement CR*; *integrity requirement IR*; *availability requirements AR*. These metrics increase or reduce the weight of confidentiality / integrity / availability impacts, according to the importance of these properties for the organization.

  An *environmental score (EnvS)* is computed using the temporal score if it exists (otherwise we use the base score), CDP, TD, CR, IR and AR.

The main shortcoming of CVSS is that it only assesses atomic attacks. Furthermore, the values of base metrics are given supposing that the user environment is safe. It don't take into account the damages already suffered by the network. One of the main goal of this paper is to modify the Base metrics assessment in order to take into account damages induced by previous atomic attacks in a multi-stage attack.

III. Attack model / attack graphs

An attack model is a tool which illustrates all possible multi-stage and multi-host attacks on an enterprise network. Each state gives values of variables, which represent the network configuration (services on each host,...), or attacker privileges gained through successive atomic attacks. Each edge represents an atomic attack which can be exploited by the attacker. Prerequisite conditions of the atomic attack must be verified in the state of departure, and the state of arrival gives the new network configuration, and attacker privileges after applying the corresponding atomic attack.

An attack graph is a subgraph of an attack model, which consists of all the paths in an attack model where the attacker succeeds in achieving a particular goal. The initial state of an attack graph is the "safe" state, representing the network state before any attack. Deadlock states represent states in which the attacker has reached his goal. A path starting from the inital state, to a deadlock state, is called *an execution*. It represents a successfull multi-stage attack, i.e. succession of atomic attacks. The set of all executions (i.e. an attack graph) gives all possibilities for the attacker to reach his goal, exploiting weaknesses on the target network.

We now define more formally the notions of attack model and attack graph. Definitions below are those given by Sheyner and Wing in [1].

Definition 1: An attack model AM is a finite automaton $AM = (S, \tau, s_0)$, where S is a set of states, $\tau \subseteq S \times S$ is a transition relation, and $s_0 \in S$ is an initial state. The state space S represents a set of three agents $I = \{E, D, N\}$. Agent E is the attacker, agent D is the defender, and agent N is the system under attack. Each agent $i \in I$ has its own set of possible states S_i, so that $S = \times_{i \in S_i}$.

Definition 2: A finite execution α of an attack model is a finite sequence of states $\alpha = s_0 s_1 ... s_n$, such that for all $0 \leq i < n$, $(s_i, s_{i+1}) \in S$.

Definition 3: Given a security property P, an execution α is correct with respect to P if P is verified in all states $s_i \in \alpha$. An execution α is failing with respect to P if at least one state $s_i \in \alpha$ violates P.

Definition 4: Given an attack model AM and a security property P, an attack graph AG of AM with respect to P is the set of failing executions of AM with respect to P.

In this paper, we will focus on network attack graphs. In these graphs, N is a computer network, E a malicious agent, and D both administrators and security softwares, like for example IDS. N is composed of three components: H, which is the set of hosts connected, C which gives the network topology and the inter-host connectivity, and T which is the expression of trust relationships between hosts. Details on this components can be found in [1].

IV. Using CVSS in attack graphs

A. Overview

The aim of our work is to use the principle of CVSS framework in attack graph in order to compute the severity level, no longer of an atomic attack, but of a complete attack scenario. From the attacker point of view, in a multi-stage attack, the first atomic attack is launched without any privilege gained on the target host before. The second atomic attack can enjoy privileges obtained by the first attack. And so on, until the last atomic attack, which enjoys privileges obtained by all previous atomic attacks.

In CVSS framework, privileges needed to exploit a vulnerability are assessed through exploitability metrics, that are AV (access vector), AC (access complexity) and AU (authentication). We compute from these metrics the Exploitability Score ES. The assessment of AV, AC and AU is made supposing that the user environment is safe, i.e. supposing that the attacker has not gained any privilege on the target host before. Our contribution is to take into account, in the exploitability vector of each atomic attack, the privileges gained previously by the attacker. Consequently, we are able to modify the values of AV, AC and AU found in IT vulnerability databases, depending on the complex attack which is currently performed.

CVSS impacts metrics assess the damages induced by the atomic attack on the target host. They don't depend on previous vulnerabilities. However, each atomic attack deals damages to the target host. If we want to be able to evaluate entire damages suffered by this host, we must introduce the notion of *cumulative damages*, which are a kind of "sum" of damages suffered by each atomic attack. Note that other hosts than the target host could suffer damages. For example, if a trust relationship exists between the target and another host, not directly under attack, this host could be impacted by the multi-stage attack. So cumulative damages should be defined for each host of the network, in order to assess all damages suffered by the entire network.

Temporal metrics are evolving in time, but are not designed to evolve in a short run. They don't rely on user environment or current context. They mainly rely on the known exploits and patches currently available. Thus, they stay independent from instant context changes as a on-going attack. Also, the environmental metrics are defined according to the administrator choice, usually after a risk analysis performed against the company information system. As temporal metrics, they are not concerned by instant context changes.

For these main reasons, we have chosen, in a first run, to focus our work on how only base metrics (especially impact metrics) can evolve according to context changes induced by a multi-stage attack.

B. A very simple example

Let's take a simple example to illustrate our purpose. Vulnerability CVE-2002-1842 is a failure in *PerlBot* program (IRC client), which allows the attacker to execute arbitrary commands via shell metacharacters in a word that is being spell checked or an e-mail address. This vulnerability has a CVSS Exploitability Score of 10, computed with an AV of 1 (network access needed), which means that the attacker can exploit this vulnerability from any computer on Internet. The CVSS Impact Score equals 6.4 (all security impacts are assessed to Partial), and the Base Score is 7.5.

Vulnerability CVE-2002-2396 is a failure in the *atftp* program (versions 0.5 and 0.6), which allows a local user (attacker) to execute arbitrary code on the system with elevated (root) privileges. The Impact Score is 10 (all security impacts are assessed to Complete), and the Exploitability Score is 3.9, with AV metric set to 0.395 (Local Access required), which means that the attacker needs a physical access to the target host. The Base Score is 7.2.

If Perlbot and atftp programs are both installed on the same host, it is possible for an attacker to use Perlbot vulnerability, then atftp vulnerability (a complex attack, composed of two atomic attacks), for executing commands on this host with root privileges.

In NVD database, AV value of atftp vulnerability is set to Local Access. But, if Perlbot vulnerability has been exploited before, Local Access to the target host is already gained. So we set AV metric of atftp vulnerability to "Network access", because it is the most favorable AV value that the attacker can gain on the target host during the complex attack. The modified Exploitability Score of atftp vulnerability then equals 10, and the modified CVSS base score equals 10.

As the concept of cumulative damages seems to be quite intuitive, it is not worth detailing it here.

C. Formal definition of CVSS network attack model

In network attack model [1], we modify component $H = \{h_i\}$ to take into account CVSS framework. We also define the notion of *host cumulative damage*, which

represents damages induced by successive atomic attacks on the target host.

Definition 5: A host $H_i \in H$ is a tuple $(id_i, svcs_i, sw_i, vuln_i, exp_i, dmg_i)$ where:

- id_i is the host identifier
- $svcs_i$ is the list of active services on the host
- sw_i is the list of softwares operating on the host
- $vuln_i = \{v_{ij}\}$ is the list of vulnerabilities on host h_i which can be exploited by an attacker. Each vulnerability v_{ij} is a tuple $(id_{ij}, BV_{ij}, bs_{ij})$ where:
  - id_{ij} is the identifier of the vulnerability (i.e. its CVE reference)
  - $BV_{ij} = (AV_{ij}, AC_{ij}, AU_{ij}, CI_{ij}, II_{ij}, AI_{ij})$ is the CVSS base vector associated with the vulnerability
  - bs_{ij} is the CVSS base score of the vulnerability, computed according to formula given in CVSS framework [10]

  Notice that we could add temporal vector and environmental vector in the definition of $vuln_i$, but as we do not use these vectors in this paper, we prefer to give a simplified definition.
  In the rest of the paper, base vectors of vulnerabilities v_{ij} are those found in NVD database.
- $exp_i = (CAV_i, CAC_i, CAU_i)$ gives the current level of exploitability gained by the attacker on host h_i. CAV_i (resp. CAC_i and CAU_i) is the maximum value of AV metric (resp. AC and AU metrics) for all vulnerabilities previously exploited on host h_i:
  - $CAV_i = max(AV_k)$
  - $CAC_i = max(AC_k)$
  - $CAU_i = max(AU_k)$

  with $k \in$ {vulnerabilities already exploited on the host}.
- $dmg_i = (CD_i, ID_i, AD_i, hd_i)$ assesses the damages suffered by h_i during the multi-stage attack, i.e. cumulative impact on the three security properties of atomic attacks. CD_i (resp ID_i and AD_i) is the cumulative confidentiality (resp integrity and availability) damage, and hd_i is the host cumulative damage score of h_i. We define each security cumulative damage as the maximal security impact suffered by the host on the corresponding security property:
  - $CD_i = max(max(CI_k), max(CD_l \times T(h_i, h_l)))$ where $\{CI_k\}$ is the set of values of CI metric for the k previous atomic attacks, $\{CD_l\}$ is the set of Confidentiality Damages of the l hosts of the network, and T the trust relationship matrix (definition of T can be found in [1]).
  - $ID_i = max(max(II_k), max(ID_l \times T(h_i, h_l)))$
  - $AD_i = max(max(AI_k), max(AD_l \times T(h_i, h_l)))$

  Intuitively, a cumulative security damage is the maximum value of the corresponding security impacts of all vulnerabilities already exploited on the target host, and the corresponding cumulative security damages of other hosts which are trusted by the target host. This is because we consider that if a trusted host is compromised, the attacker can enjoy the same privileges on the target host than on the comprimised host.
  Computation of hd_i is made following the impact score formula of CVSS framework:

  $$hd_i = 10.41 \times (1 - (1 - CD_i) \times (1 - ID_i) \times (1 - AD_i))$$

  hd_i ranges from 0 to 10, and is computed using CD_i, CI_i, and CA_i with the same weights. But this formula can be changed, in order to match administrator requirements. For example, if we are only concerned by the confidentiality property, we can decide that hd_i equals CD_i. We can also use any weighted combination of cumulative damages.

D. CVSS network attack graph

Notion of CVSS attack graph is the same than classical attack graph. We do not detailled its computation in this paper, because it is strickly the same as for classical attack graphs. The reader can refer, for example, to [6] for more details on this point.

In our opinion, CVSS attack graphs can be used into four different ways:

- if the goal of the attacker is to gain some privileges on a target host, for example get root privileges, the structure of CVSS attack graph is the same than for classical attack graph. The contribution is that in each state, we can now know damages suffered by all the hosts of the network, and in particular by the target host. This is not possible in classical attack graphs.
- when building an attack graph, we search for all executions which lead to a state in which the attacker has reached his goal. We can use CVSS attack graphs in another way. Is is possible to search for all executions which lead to a state in which at least one host h_i have $hd_i \geq L$, where L is a certain level of damage ($L = 0..10$). In order to build such a graph, in definition 4, we must replace the classical security property P by P', such as "$\forall h_i, hd_i \leq L$" . Notice that in this case, we do not focus on a particular target host, but on all the hosts of the network.
- however, it is also possible to combine the two previous points, and to focus on a certain level of damages of a particular host h_k, using propery P" "$d_k \geq L$". In this case, we do not focus on a particular goal reached on host h_k, but on every execution which leads to a damage impact on host h_k greater than L, whatever privileges gained by the attacker are.

- if we define the notion of network damage nd, i.e. an assessment of damages suffered by the entire network instead of a single host, than we can use property P''' "$nd \leq L$". The main difficulty is to define nd. Several solutions are possible:
  - it could be the value of the greater damage score for all hosts: $\forall i, nd = max\{ds_i\}$
  - it could be the mean value of damage scores: $nd = mean\{hd_i\}$
  - it could be the value of damage score of a particular host h_j, which is very important in the organization: $nd = hd_j$. In this case, we have the same kind of property than P".

  In fact, administrators can choose the definition of nd which is the most relevant for them. We give some possibilities here, but other definition can be proposed, without any change for the computation of CVSS attack graphs.

V. Example

We use an example which is inspired from those given by Ghosh and al. in [9], and Sheyner in [7]. The network topology is given in figure 1. It consists in four Hosts:

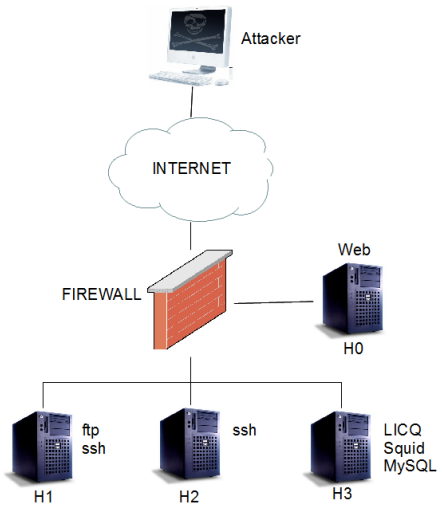

Figure 1. Example of network

- H_A, the attacker machine (outside the target network)
- H_0, a Web server (Windows NT 4.0) in a DMZ
- H_1, a Windows domain server (Windows 2000 SP1)
- H_2, a client (Windows XP SP2)
- H_3, a Linux server (Red Hat 7.0)

H_1, H_2 and H_3 are in the private area of the target network. The list of vulnerabilities on each host of the network is given in table I. We use base metrics of similar vulnerabilities found in NVD database.

Host	Name	Description	ES	IS	BS
H_0	iisbof	IIS buffer overflow	10	10	10.0
H_1	rhost	ftp rhost overwrite	8.6	2.9	4.3
	rsh	rsh login	10	2.9	5.0
H_2	nns	netbios null session	10	4.0	5.8
H_3	rtu	LICQ remote to user	10	6.4	7.5
	locbof	Local buffer overflow	10	10	10.0

Table I
List of existing vulnerablities on hosts of the network

As defined by Sheyner and al. in [7], each vulnerabilitity has a set of preconditions and effects. We do not detailled all these vulnerabilities here. We only give some signifiant examples below, in table II.

	Preconditions		Effects	
Vulnerability	Intruder	Network	Intruder	network
iisbof(S,T)	$user(S)$ $\neg root(T)$	$w3_T$ $R(S,T,80)$	$root(T)$	$\neg w3_T$
rtu(S,T)	$user(S)$ $\neg user(T)$	$licq_T$ $R(S,T,5190)$	$user(T)$	
rhost(S,T)	$user(S)$ $\neg user(T)$	ftp_T $R(S,T,21)$	$Trust(T,S)$	

Table II
Description of some vulnerablities

For vulnerability $rhost$, $Trust(T,S)$ means that if the attacker launches this attack against T, from S, then T trusts S. In other words, the attacker gains the same privileges on host T than on host S.

Moreover, we always consider that $root(X) \rightarrow user(X)$, which means that if an attacker has gained root privilege on host X, then he also enjoys user privilege on this host.

Host	H_A	H_0	H_1	H_2	H_3
H_A	All	80	None	None	None
H_0	None	All	All	All	All
H_2	None	80	All	All	All
H_2	None	80	All	All	All
H_3	None	80	All	All	All

Table III
Firewall policy (function R)

The firewall limits the connectivity between hosts. Briefly, it allows public hosts (H_A) to access port 80 of the Web server (H_0), and it forbids public hosts (H_A) to access hosts of the private area (H_1, H_2 and H_3). H_1,H_2,H_3 are not allowed to access Internet, but they can access port 80 on the Web server. This policy is summarized in table III. Function R implements this policy (for example, $R(S,T,80)$ means that the firewall allows the connectivity from host S to host T on port 80).

We assume that there is no trust relationship between hosts on the target network, i.e. $\forall (i,j), T(i,j) = 0$.

We build a regular attack graph, focusing on the attacker goal "gaining root privilege on host H_3". Figure 2 shows

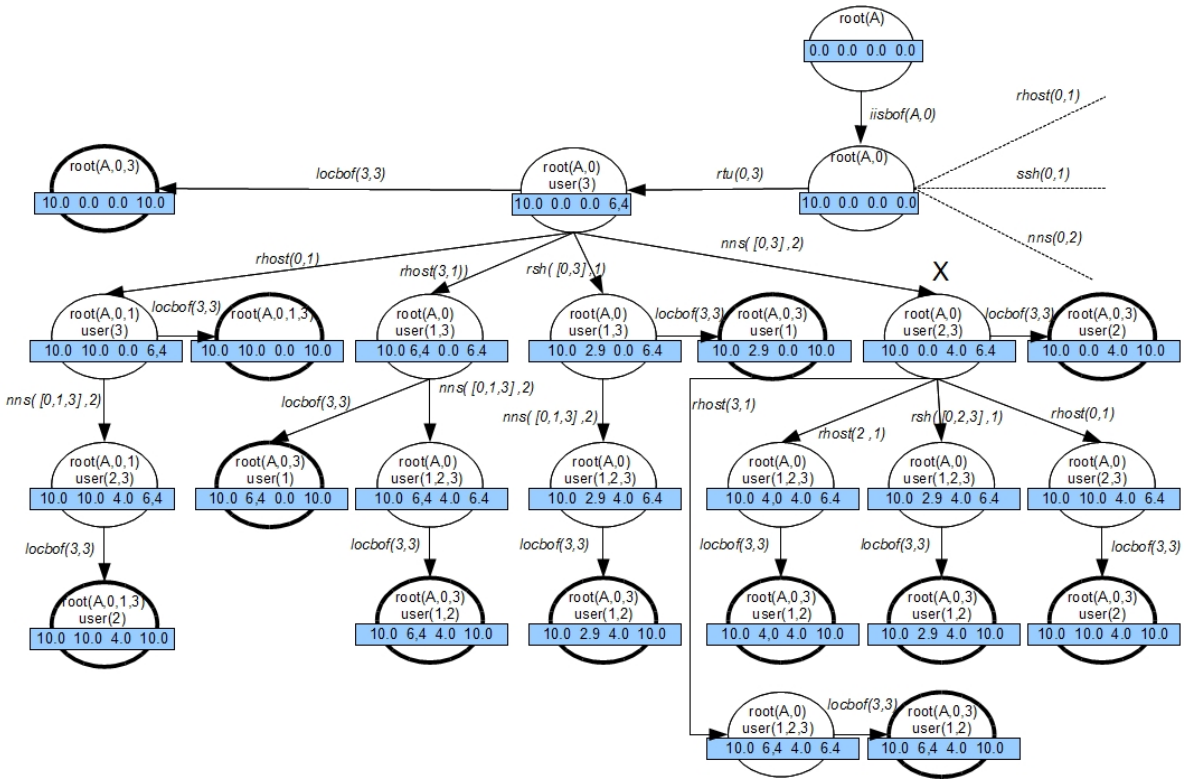

Figure 2. CVSS attack graph, focusing on property "gaining root privilege on host H_3"

the corresponding CVSS attack graph. Unfortunately, we can only represent a portion of the graph because of its large size (only executions which begin with $iisbof(A, 0)$ then $rtu(0, 3)$ are shown). Each circle represents a network state after the exploitation of an atomic attack. The first state (on top-right of the graph) is the initial state, in which the attacker have not launched any attack against the target network. Each deadlock state (thick circle) is a state in which the attacker has gained root access on host H_3. Edges between states are labelled with the name of the atomic attack which leads to the state change. The first (resp. second) number into parenthesis is the host from (resp. against) which the atomic attack is launched.

In each state, we can find several informations;

- the list of privileges gained by the attacker on the different hosts of the network.
  - *root(X)* (resp. *user(X)*) means that the attacker has gained root (resp. user) privilege on host X.
- the cumulative host damages scores on each host (in grey rectangles): $[hd_{H0}\ hd_{H1}\ hd_{H2}\ hd_{H3}]$

The main interest of this graph is the quantification of the impact of atomic attacks on hosts. In deadlock states, we can obtain 9 different host damages vectors : [10 0 0 10], [10 2.9 0 10], [10 10 0 10], [10 2.9 4 10], [10 10 4 10], [10 6.4 0 10], [10 6.4 4 10], [10 0 4 10] and [10 4 4 10]. These vectors show and assess how different can be the impact of the attacker, depending on the succession of atomic attacks he uses.

Note that the use of vulnerability $rhost$ can lead to different impact scores, depending from which host it is launched. For example, if we focus on state X of figure 2, vulnerability $rhost$ can be exploited from hosts H_0, H_2 or H_3 (because the attacker has already gained at least user privilege on each of these hosts). But the impact of this vulnerability on host H_1 is not the same, depending on the source host. If the atomic attack is exploited from H_0, the attacker gains root privilege on host H_1 ($trust(1, 0)$). Then the corresponding host damage equals 10. If the attack is exploited from host H_2 or H_3, then the attacker gains "only" user privilege. The host damage will be 4.0 from host H_2 ($trust(1, 2)$), and 6.4 from H_3 ($trust(1, 3)$). Moreover, at the same stage (state X), if the attacker choose to use vulnerability $rsh([2, 3], 1)$ rather than $rhost([0, 2, 3], 1)$, he gains the same privilege as using $rhost(2, 1)$ or $rhost(3, 1)$, i.e user privilege on host H_1, but the host damage on host H_1 is 2.9 (no trust relationship as for $rhost$, 2.9 is the impact score of vulnerability rsh). Even if the privileges gained by the attacker on a given host stays the same, the impact can be different, depending on the hosts from which the attacks are issued. These differences of impacts can't be measured

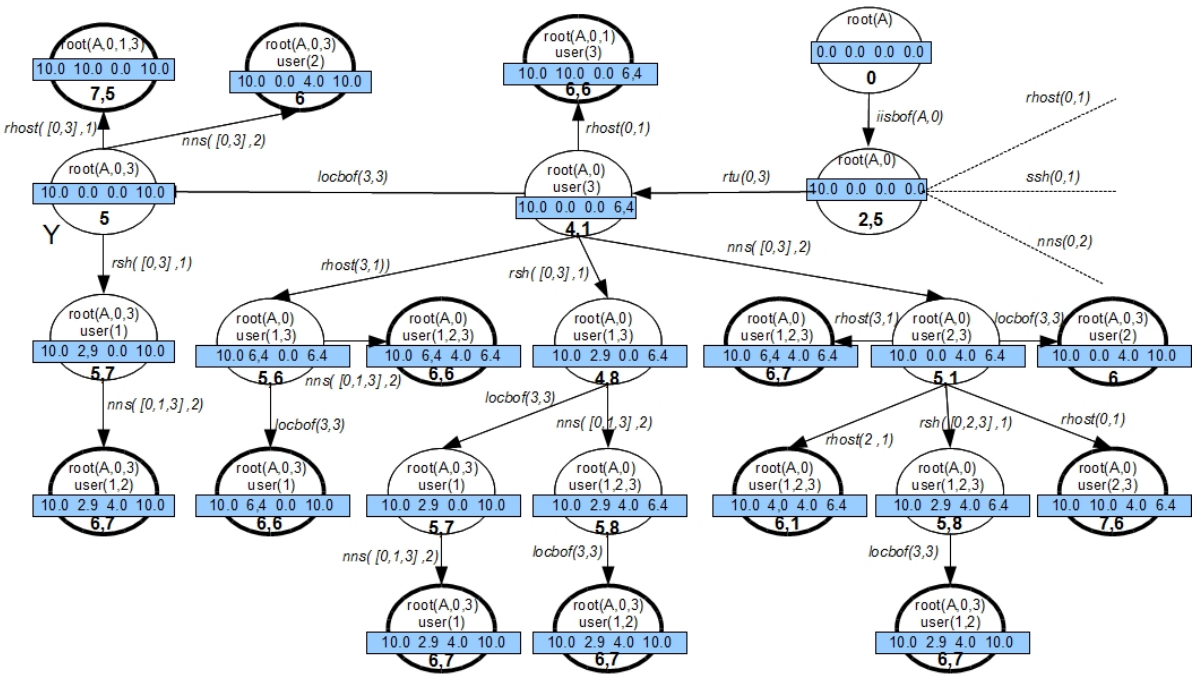

Figure 3. CVSS attack graph, focusing on property "$nd \leq 7$"

by a regular attack graph. CVSS attack graphs are able to show subtleties of a multi-stage attack, where the attacker deals with trust relationships of the target network.

In section IV-D, we have proposed to build CVSS network attack graphs, focusing on several properties. In particular, we defined the notion of *network damage nd*. If we consider, here, that nd is the mean of host damages ($nd = mean(hd_0, hd_1, hd_2, hd_3)$), then we argue that we can use this metric as an indicator of the stealthness of the attack. Stealthness stands for the fact that the attacker uses low impact atomic attacks rather than high impact atomic attacks, in order to make the detection of its actions more difficult. Table IV gives value of nd for each deadlock state. If nd value is low, then it means that the attacker have used a few number of atomic attacks to achieve its goal. For example, for host damages vector [10 0 0 10], nd equals 5, and corresponds to execution $iisbof(A,0)/rtu(0,3)/locbof(3,3)$, in which only 3 atomic attacks are used. On the other hand, a high value of nd means that the attacker has exploited a lot of vulnerabilities, or strong vulnerabilities. For example, for host damages vector [10 10 4 10], nd equals 8.5, and corresponds to execution $iisbof(A,0)/rtu(0,3)/nns([0,1,3],2)/rhost(0,1)/locbof(3,3)$, in which 5 atomic attacks are used. Of course, using a mean value in te definition of nd is a very (too) simple choice, but it seems to be still a good first indicator of attack stealthness.

Host Damages	nd	execution (begins with $iisbof(A,0)/rtu(0,3)/\ldots$)
[10 0 0 10]	5	$locbof(3,3)$
[10 2.9 0 10]	5.7	$rsh([0,3],1)/locbof(3,3)$
[10 0 4 10]	6	$nns([0,3],2)/locbof(3,3)$
[10 6.4 0 10]	6.6	$rhost(3,1)/locbof(3,3)$
[10 2.9 4 10]	6.7	$rsh([0,3],1)/nns([0,1,3],2)/locbof(3,3)$
[10 4 4 10]	7	$nns([0,3],2)/rhost(2,1)/locbof(3,3)$
[10 10 0 10]	7.5	$rhost(0,1)/locbof(3,3)$
[10 6.4 4 10]	7.6	$rhost(3,1)/nns([0,1,3],2)/locbof(3,3)$
[10 10 4 10]	8.5	$nns([0,1,3],2)/rhost(0,1)/locbof(3,3)$

Table IV
NETWORK DAMAGES ASSESSMENT IN EACH DEADLOCK STATES

Another proposition that we make consists in building CVSS attack graph using a quantitative property rather than focusing on privileges gained on hosts. For example, figure 3 gives the CVSS attack graph build focusing on property "$nd \leq 7$". The aim of this graph is to show the possible malicious activity which does not lead to a network damage greater than 7. In other words, we want to see the malicious activity which is under a certain level of stealthness. Once again, because of the large size of the graph, we only give executions which begin with $iisbof(A,0)/rtu(0,3)$. Value of nd is the bold number in each state.

Let's compare two different executions of this graph:

1) $iisbof(A,0)/rtu(0,3)/rhost(0,1)$, $nd = 6.6$, attacker privileges = root(A,0,1), user(3)
2) $iisbof(A,0)/rtu(0,3)/locbof(3,3)/rsh([0,3],1)/nns([0,1,3],2)$, $nd = 6.7$, attacker privileges =

root(A,0,3), user(1,2)

These two executions lead to approximatively the same value of network damage, but are really differents. Execution 1) is the less stealthy, because it reaches $nd = 6.6$ using only 3 atomic attacks, which corresponds to a mean network damage by atomic attack of 2.2 ($6.6/3$). On the other hand, execution 2 is the more stealthy, because it reaches $nd = 6.7$ after exploiting 5 atomic attacks, which corresponds to a mean network damage by atomic attack of 1.3. Moreover, privileges gained by attacker using execution 1 are lower than those gained through execution 2. So we think that this kind of graph can be also useful to determine executions are the more stealthy. These executions are the combinations of atomic attacks which are the more difficult to detect for network administrators.

VI. Conclusion

In this paper we adapt the definition of attack graphs in order to be able to use them in conjunction with CVSS framework. We propose the definition of host damage hd and network damage nd, in order to assess the cumulative impact of the different attacks launched against a host or a network. The impact of each atomic attack is found in NVD database, with respect to CVSS framework.
With these definitions of new damage metrics, we add quantitative informations to regular attack graphs. But we also can build new kinds of attack graphs, focusing on quantitative properties (for example, $\forall i, hd_i \leq 6$) rather than qualititative properties, i.e. privileges gained by the attacker on different hosts. These graphs can be used, for example, to analyse the possible stealthy malicious activity.

In future works, we need to improve this first proposition. For example, in CVSS framework, the environmental part has been designed to allow network administrators to modify the initial assessment of vulnerabilities made by NVD experts. More precisely, it is possible to amplify or attenuate the impact of a vulnerability on the three security properties, taking into account the security policy of the network and the security level of the target host (use of security requirements). It could be very interesting to add this functionality to our proposition, in order to take into account the potentiality of defense of each host of the network against each atomic attack.
As seen in section IV, exploitablity metrics of base vector can also be used in CVSS attack graphs. By this way, we will be able to take into account changes induced by previous atomic attacks on the next one.
A prototype will be proposed shortly, with tools for the analysis of our CVSS attack graphs. In particular, we want to propose quantitative tools in addition to regular qualitative tools, in order to analyse attack graphs from a vulnerability impact point of view.

References

[1] Sheyner, O., and Wing, J., *Tools for generating and analyzing attack graphs*, Second International Symposium on Formal Methods for Components and Objects, Leiden, The Netherlands, November 2003.

[2] National Vulnerability Database (http://nvd.nist.gov/).

[3] Gallon, L., *On the impact of environmental metrics on CVSS scores*, The Second IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT-2010) - Symposium on Secure Computing (SecureCom-10) Minneapolis, Minnesota, USA, August 2010.

[4] Gallon, L., *Vulnerability discrimination using CVSS framework*, 4th IFIP International Conference on New Technologies, Mobility and Security Paris, France, February 2011.

[5] Jha, S., Sheyner, O. and Wing, J., *Two formal analyses of attack graphs*, 15th IEEE Computer Security Foundations Workshop (CSFW'2002), Nova Scotia, Canada, June 2002.

[6] Sheyner, O., Haines, J., Jha, S., Lippmann, R. and Wing, J., *Automated Generation and Analysis of Attack Graphs*, IEEE Symposium on Security and Privacy (S&P'2002), Berkeley, California, USA, May 2002.

[7] Sheyner, *Scenario Graphs and Attack Graphs*, PhD thesis, Carnege Mellon University, 2004.

[8] Mehta, V., Bartzis, C., Zhu, H., Clarke, E. and Wing, J., *Ranking attack graphs*, 9th International Symposium on Recent Advances in Intrusion Detection (RAID'2006), Hamburg, Germany, September 2006.

[9] Ghosh, N. and Ghosh S.K. *An Intelligent Technique for Generating Minimal Attack Graph*, First Workshop on Intelligent Security (Security and Artificial Intelligence), SecArt '09, Thessaloniki, Greece, September, 2009

[10] Mell, P., Scarfone, K., and Romanosky, S., *A Complete Guide to the Common Vulnerability Scoring System (CVSS) Version 2.0*, Forum of Incident Response and Security Teams (http://www.first.org/cvss/cvss-guide.html), June 2007.

[11] Scarfone, K. and Mell, P., *An analysis of CVSS version 2 vulnerability scoring*, 5th International Workshop on Security Measurement and Metrics (MetriSec'09), Orlando Florida, USA, October 2009.

[12] Elahi, G., Yu E.and Zannone N., *A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities*, IEEE Requirements Engineering, vol 15, number 1, pages 41-62, Atlanta, USA, september 2009.

[13] Wang, L., Islam T., Long. T., Singhal A. and Jajodia S., *An attack graph-based probabilistic security metric*, 22nd annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec 2008), London, UK, july 2008.