

Risk Assessment: Intercepting VoIP Calls

Marco Benini and Sabrina Sicari
Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria
{marco.benini, sabrina.sicari}@uninsubria.it

Abstract

This paper addresses the problem of intercepting a phone call based on the voice-over-IP (VoIP) protocols. The importance of this problem arises both from the interest that the VoIP service is gaining in the Internet, and from the peculiarities of the VoIP communication systems. In this respect, we show how to adapt a general methodology for risk assessment towards the specific aspects of the VoIP service by means of a concrete case study: protecting the VoIP phone calls from being intercepted.

I. INTRODUCTION

The voice-over-IP (VoIP) services have seen a great raise of interest and popularity in the last years. The reason of their success lies both in the appearance of simple yet effective products, e.g., *Skype* [6], and in the promise of an high-quality and low-cost substitute for the traditional telephony.

Although the VoIP services are now mature enough to partly fulfil the previous expectations, they also begin to show a new brand of problems.

In fact, the most problematic aspect of VoIP is its security: in the world of traditional telephony, privacy and security of conversations are guaranteed up to the physical layer of the network and of the callers; a phone call can be heard by an intruder either by directly listening to the call, e.g., being in the same room, or by violating the physical security of the phone network or of its devices.

On the contrary, the substitution of the traditional telephony with the VoIP services allows an intruder to listen to a private conversation by capturing the content of an Internet connection. This is possible since the network is shared among the caller, the callee and the intruder.

The problem of VoIP security has been signalled by many researchers in the telecommunication area and in the Internet security area (see Section V for a discussion). A clear emerging point is that VoIP security is more than just Internet security applied to a new service because of the distinct characters of VoIP services.

Despite the clear recognition of the peculiarities of VoIP services, there is still no systematic study of the specific threats and of their countermeasures.

The main consequence of this gap in the VoIP technology is a lack of consciousness about the risks associated with the adoption of a VoIP-based solution; in fact, on the one side there are mature, stable and solid VoIP products offering important economical benefits, while, on the other side, there is a lack of evaluation in the security/privacy risks of their use.

Therefore, the present paper shows the application of a simple yet effective risk assessment methodology to a case study of VoIP services. In particular, the risk to intercept a VoIP call is evaluated assuming that the caller and the callee reside in *secure* networks interconnected by the Internet.

The analysed situation is common since it is the case of most organisations wanting to move from traditional telephony to VoIP for their internal vocal communications; when the organisation is spread on the territory, it is economically convenient to link the various branches to Internet and to allow the flow of inter-organisation communications to travel across the Internet.

This paper analyses the risk of call interception in the described scenario and shows how the *usual* Internet-based countermeasures are only partially effective, mitigating the risk, but unable to eliminate it.

The paper is organised as follows: in Section II we describe the VoIP system architecture. In Section III, the risk assessment methodology is summarised, while, in Section IV, we apply the methodology to the VoIP vulnerabilities. Finally, Section V is a brief overview of the related works, and Section VI concludes the paper.

II. THE VOIP ARCHITECTURE

The standard *pure* VoIP architecture is based on a set of *IP phones*, hardware or software telephones operating over the IP protocol, interconnected by an internet; the IP network can be connected to a traditional phone system (PSTN) by means of a *VoIP gateway*, transforming VoIP calls and conversations into phone calls and conversations to/from a PBX. In addition, the IP phones may benefit of a *Voice server* providing auxiliary support to VoIP services, e.g., translation from user names to IP address and vice versa.

As stated in the Introduction, our goal is to evaluate the *wire tapping* risk in a VoIP system, i.e., the risk a live conversation between two IP phones can be successfully intercepted. The centre of our chosen approach [16] to risk evaluation is to consider the dependencies among the system vulnerabilities: evidently, these dependencies are strictly related to the system architecture.

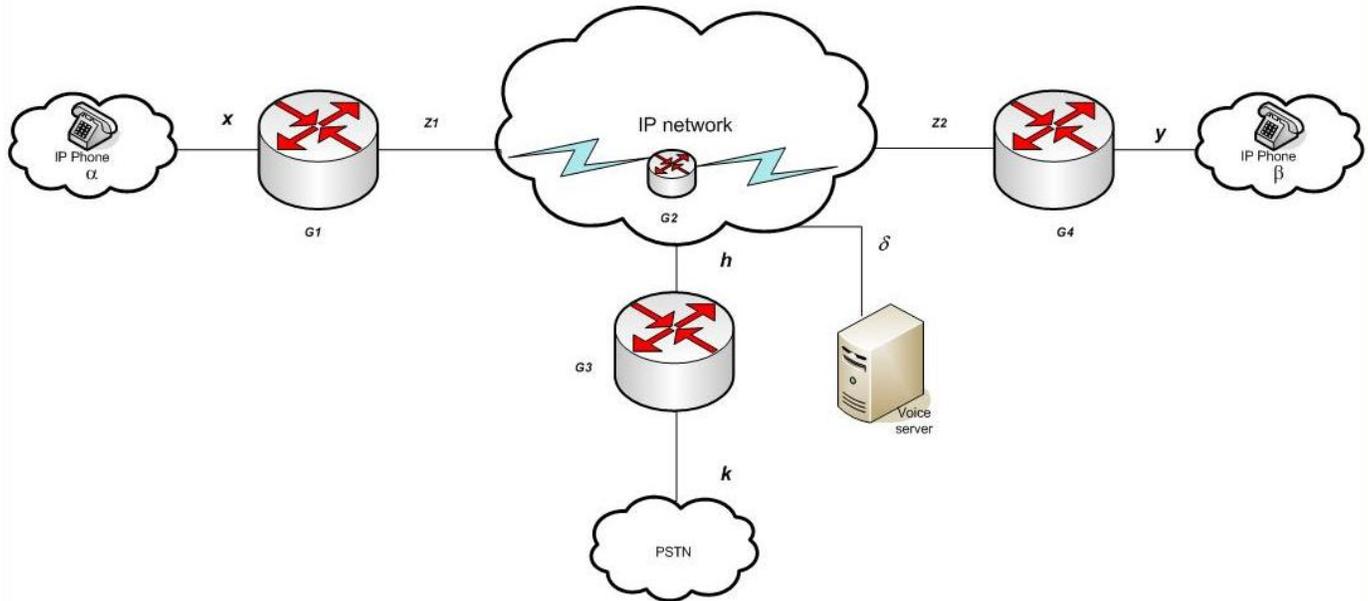


Fig. 1. The reference architecture

Therefore, it is important to fix a reference architecture detailed enough to enable the application of a risk assessment methodology, and abstract enough to model a vast amount of real-world situations. Our choice is shown in Figure 1, whose main components are:

- *IP phone*: a terminal that has native VoIP support and can directly connect to an IP network;
- *VoIP gateway*: a network device that converts signals to/from the telephony interfaces (POTS, T1/E1, ISDN, E&M trunks) and VoIP;
- *Voice server*: a network server that provides management and administrative functions to support the routing of calls across the network; in a system based on H.323, the server is known as a *gatekeeper*; in SIP/SDP, the server is called a *SIP server*; in a system based on MGCP or MEGACO, the server is named a *call agent*;
- *IP network*: an interconnection structure based on the TCP/IP protocols; the IP network can be a private network, an intranet, or the Internet.

In Figure 1, the IP phones α and β lie in two private networks that are linked respectively to the G_1 and G_4 network gateways (the *border gateways*). The G_2 gateway represents one of the many gateways that are inside the internet; we call G_2 an *internal gateway*. The G_3 VoIP gateway manages the communication towards the PSTN.

We would like to remark that the chosen architecture is general, i.e., most settings of VoIP applications are based on it. This fact is rather obvious since, differently from the standard VoIP architecture, we require that the IP phones lie in private networks connected by means of a public network. Our assumption is that the private networks are *safe*, while a malicious user may operate only in the public internet.

III. MEASURING RISK

The goal of risk assessment is to quantitatively evaluate the risk of an undesired event in a given environment. In our case, the environment has been fixed in Section II as the reference architecture of a VoIP system; in addition, the undesired event is clear, i.e., intercepting a phone call.

Therefore, the aim of this Section is to define the notion of risk and to describe the methodology we are going to employ in our case study; the analysis of the formalisation and of the properties of the risk assessment procedure is beyond the scope of this paper and the interested reader is referred to [16].

In our approach, the risk is a function on two variables: the damage potential and the level of exploitability. The damage potential measures the ability to damage as, e.g., the average economical loss caused by an attack; the level of exploitability measures the ease to perform an attack, as analysed in [11].

Our risk assessment procedure consists of four steps:

- **Step 1:** A threat to the system under examination is modelled by using an attack tree: the root node represents the attack goal and, recursively, children can be alternative subgoals, each one satisfying the goal (or subtrees) or partial subgoals, whose composition satisfies the goal (and subtree). The leaves of the tree are the *potential* vulnerabilities of the system that should be matched with the *actual* known vulnerabilities.

To each vulnerability v is associated a numerical index E , called *exploitability*, that measures how probable is that v will be successfully used to break the security of our system. The evaluation of E can be rough: in order to apply the methodology it suffices that the partial order of indexes among dependent vulnerabilities (see below) reflects the relative difficulty of exploitation.

A meaningful assessment of E is a matter of both experience and ingenuity, but only the *relative* ease of exploitability has to be estimated, a judgement on which experts often agree.

- **Step 2:** Dependencies among identified vulnerabilities are introduced. A vulnerability A depends on a vulnerability B if and only if when B is already exploited, then A becomes easier to exploit. Dependencies should be analysed and weighted by taking into account contextual, architectural and topological information, as detailed in the following.
- **Step 3:** The index E of each single vulnerability is updated according to the dependencies, as described later.
- **Step 4:** The risk associated to the threat under examination is finally computed by recursively aggregating exploitabilities along the attack tree. The exploitability of an or subtree is the easiest exploitability (maximum value) of its children, and the exploitability of an and subtree is the most difficult exploitability (minimum value) of its children. The aggregated exploitability measures the level of feasibility of the attack and can be combined with the damage potential to assess the risk of the threat.

A. Exploitability of dependent vulnerabilities

The dependencies identified in Step 1 influence the values of the exploitabilities of the single vulnerabilities. The exact evaluation of this influence is the core of Step 3 and of the whole risk assessment methodology.

Therefore, to understand how to numerically evaluate dependencies and to correctly apply the methodology to our case study, we describe the formalisation of dependencies in our approach.

The system architecture can be described as a graph $S = \langle C, L \rangle$ where C is the set of *components* and L is the set of *links* between components.

The components (links) are exposed to the set of vulnerabilities V_C (V_L) where an element $(v, \nu) \in V_C$ (V_L) means that the component (link) v is susceptible to be subverted thanks to the flaw ν . The set of all vulnerabilities is $V = V_C \cup V_L$. To ease notation, we denote as $element(\nu) \in C \cup L$ the element of S to which the vulnerability ν applies.

Initially, during Step 1 of the risk assessment procedure, a numerical measure of exploitability is given for every vulnerability; we call this value the *exploitability* $E_0(\nu)$ of the vulnerability ν in the system S .

The notion of exploitability is generalised as the $E: \mathbb{N} \times V \mapsto M$ maps vulnerabilities to M , a total ordered set of degrees of exploitability, e.g., $\{x | 0 \leq x \leq 10\}$ where 0 means “not exploitable at all”.

However, as already noticed in Step 3, the system architecture imposes dependencies among vulnerabilities. For example, we need to understand if it is easier to exploit a vulnerability of a component given that an input link attached to it was already compromised or a component attached to any of its input links was already compromised.

We denote with $E(\alpha|\beta)$ the exploitability of the vulnerability α given that the vulnerability β has been already exploited.

Dependencies among vulnerabilities can be represented in the dependency graph $G = \langle V, D \rangle$. The edge $(\beta, \alpha) \in D$ if $E(\alpha|\beta) \geq E_0(\alpha)$, i.e., if it is easier to compromise $element(\alpha)$ when one has compromised $element(\beta)$.

To each node α in the dependency graph G we associate the value $E(i, \alpha)$, written as $E_i(\alpha)$. Initially, each node α has the value $E_0(\alpha)$ that is the initial measure of how difficult is to exploit the vulnerability. The conditional exploitabilities are represented by the values $E(\alpha|\beta)$ on the edges (β, α) of G .

To calculate the influence of dependencies, we apply the following formula

$$\forall \nu \in V. E_{i+1}(\nu) = \max(E_i(\nu), \{\min(E(\nu|\gamma), E_i(\gamma)) : (\gamma, \nu) \in D\}) \quad (1)$$

whose rationale is: at any iteration $i + 1$, the function E_i is updated considering if a vulnerability ν becomes easier to exploit at time $i + 1$ thanks to a dependency (ν, γ) . Since the method is guaranteed to converge in finite time, we apply the Formula 1 until a fixed point is reached (equilibrium).

IV. APPLYING METHODOLOGY TO VOIP

We are interested in evaluating the risk associated to a VoIP wire tapping attack. A wire tapping attack succeeds when the attacker gains a reproducible copy of a telephone conversation. We apply step by step the risk assessment methodology (see Section III).

A. Step 1: construction of attack tree

We draw the attack tree under these hypothesis:

- 1) The physical access to LAN IP phones is denied: an hacker has just the possibility to access by Internet. Moreover, we assume that operating inside the LANs is impossible: although this appears as a limitation, this is usually the case that bothers when a structured organisation wants to protect their phone conversations.
- 2) The hacker knows the addresses of the border gateways delimiting the LANs where IP phones are in. This assumption is reasonable when we consider the nature of his attack: he does not want to steal a generic conversation, but a call between two distinct locations (IP phones) on the Internet, thus he can reconstruct in standard ways the external structure of the target networks (tracerouting and/or consulting the registration data of networks in the standard databases of IANA by means of the `whois` service and/or using the DNS database).
- 3) The hacker does not know the exact location of phones, neither their IP addresses. This assumption is reasonable since the internal structure of the attacked LAN is not usually published.

Usually, the security analyst starts enumerating all possible attacks and combines them to form a large attack tree. But a VoIP system is a complex scenario and there are many ways/paths to realise an attack; actually Internet introduces many attack vectors. Hence, in order to give a complete and sound view we construct a *master* tree that describes the principal way to reach the goal and, after, we construct and specify two sub-trees that point out the system vulnerabilities. In this way, reading and understanding an attack tree becomes simpler and useful. The synthesised global attack tree is depicted in Figure 2.

The subgoals *access gateway* and *replace gateway* require interesting operations and they point out characteristic vulnerabilities of VoIP System. In fact, a VoIP call operates on a double connection: a control channel, used to determine the parameters of the communication, to start and to end the voice transfers, to identify the connection for the media channel, etc.; a media channel, whose goal is to move the sound (voice) from one end-point to the other. Therefore, the subgoals are present both in the *copy of control channel* and the *copy of data channel* nodes since stealing the content of both channels is needed to perform a successful attack.

Moreover, accessing a gateway or replacing a gateway are two actions that cannot be performed on the same patterns as in most Internet communications: in fact, those actions must not interfere with the existing connections on the gateways, otherwise the VoIP call can be influenced, with the obvious consequence that it will drop. The reason is that VoIP calls are real-time, streaming connections, that is, any loss or detour has an high probability to conclude the communication, thus destroying what a malicious user wants to observe.

When the hacker wants to copy the content of the control or the media channel, it has to decide whether to violate a gateway or to substitute it, thus performing an “access gateway” or a “replace gateway” action. The choice

- Goal:** wire tapping
1. Decode media type
 - 1.1 Copy of control channel (AND)
 - 1.1.1 Copy traffic
 - 1.1.1.1 Access gateway
 - 1.1.1.2 Replace gateway
 - 1.1.2 Select and copy control channel
 - 1.1.2.1 Access gateway
 - 1.1.2.2 Replace gateway
 - 1.2 Copy of data channel (AND)
 - 1.2.1 Copy all traffic
 - 1.2.1.1 Access gateway
 - 1.2.1.2 Replace gateway
 - 1.2.2 Select and copy data channel
 - 1.2.2.1 Access gateway
 - 1.2.2.2 Replace gateway
 - 1.3 Decode data channel
 - 1.3.1 Establish coding algorithm (AND)
 - 1.3.1.1 Guess coding algorithm
 - 1.3.1.2 Read control channel
 - 1.3.2 Establish encryption key

Fig. 2. The master attack tree

depends both on the specific attack technique, and on the position of the gateway in the network; in fact, according to our hypotheses, it is impossible to replace a border gateway, since no access to the private networks is permitted; on the contrary, both the access and the replace actions are possible on internal gateways, since the data traffic can be deviated, allowing the replacement of a gateway, and the device can be attacked, since it is exposed on the network.

When the attack to the gateway is successful, among the connections traversing the gateway, the attacker can choose the interesting ones, namely the control channel and the media channel of the VoIP call. To single out the media channel, that uses no well-known service, one has to decode the control channel (hence goal 1.1 in the attack tree).

When a copy of the media channel has been duplicated, the malicious user has to decode its content in order to understand the human voice encrypted in the channel. Therefore, the malicious user has to understand how the media channel codes the human voice, and, if the coding is a real encryption, the user has to gain the knowledge of the coding key. The first step, knowing the encryption algorithm is simple, since this information has been transmitted in the control channel, thus the hacker can access it having a copy of that channel. If data is encrypted, its decoding is a *standard* attack to data already available to the malicious user, hence we do not explicit the corresponding attack tree, that could be reconstructed from literature, e.g. [19].

The concrete attack tree for the subgoal “access gateway” is depicted in Figure 3; the one for the subgoal “replace gateway” is shown in Figure 4. A brief description of the attacks are in order: the “access gateway” goal is obtained by identifying the gateway we want to access, by trying to access it by a standard protocol used to administer these devices, e.g., `telnet`¹, and finally, by forcing the access with a weak or stolen password; the “replace gateway” attack is similar, starting with the identification of the device to replace with a malicious substitute, then forcing the traffic from a border gateway to the identified gateway to be diverted to the malicious gateway.

The most interesting technical aspects of these attacks, that enables us to identify the vulnerabilities of the system, are:

- the first vulnerability, referred as V_1 , is a weak authentication in a gateway, e.g., a blank or a default password, or a easily guessable password, by means of a dictionary attack. We observe that a weak authentication in a border gateway acts differently than in an internal gateway, thus we will speak of V_{1F} to identify a weak

¹Here and elsewhere we concentrate on particular attack path to better exemplify the general technique.

Goal: access gateway

1. Identify gateway address (AND)
 - 1.1 Assumption in the case of a border gateway (V_3)
 - 1.2 Identify internal gateway address
 - 1.2.2 Traceroute to a border gateway address (AND) (V_4)
 - 1.2.3 Choose a weak gateway close to the border gateway
2. Access the gateway (AND)
 - 2.1 Telnet; if it fails, we can try another gateway or replace the gateway
3. Identify gateway password
 - 3.1 Default or weak password (V_1)
 - 3.2 Guess
 - 3.3 Sniff a link connected to the gateway (V_2)

Fig. 3. Attack tree for the subgoal “access gateway”

Goal: replace gateway

1. Identify gateway address (AND)
 - 1.1 Traceroute to a border gateway address (AND) (V_4)
 - 1.2 Choose a weak gateway close to the border gateway
2. Poison the route between a border gateway and the identified gateway
 - 2.1 Announce a false OSPF bandwidth

Fig. 4. Attack tree for the subgoal “replace gateway”

authentication vulnerability on a border gateway (V_1 on the *frontier*), and we use V_{1I} for a weak authentication on an internal gateway.

- sniffing a link that connects a gateway to the network is another vulnerability, V_2 , that may enable us to gain the password to access the gateway: if an administrator tries to access it, we can observe the password by looking at the content of the telnet transmission. Moreover, sniffing is difficult, since it requires the ability to put a suitable device exactly on the observed link that must be directly attached to the gateway we want to violate.
- identifying the address of a gateway is fundamental in order to attack it, either to access or to replace. We assumed to know the address of the border gateways of the LANs where IP phones are located. This assumption is not trivial: usually, we know something about the IP phones that enables us to identify the networks where they are located. We do not want to analyse how this information is gained, but we remark that a fundamental requirement to exploit it, is a weak setting in the IP phones, e.g., giving the IP phones the names of their owners and publishing the list. We identify the weak setting vulnerability as V_3 .
- a fundamental step in our attack strategy is to know, to some extent, the route the call follows from an IP phone to the other one. The idea is that, if we know the route a call follows, then we can either access or replace a gateway on that route, thus having the possibility to observe the call itself. We suggested that the knowledge of that route can be acquired by means of the `traceroute` service, that calculates the route between our host and a target node of the Internet, reporting also an estimation of the round trip time. By using the source routing option of IP [17], we can force `traceroute` from our malicious host to one of the border gateway to follow a route through the other border gateway. In this way, we see the optimal route between the two border gateways, and we can calculate the round trip times between every pair of gateways in that route. The knowledge of times enables us to poison the dynamic routing information on the network, i.e., to obtain the next goal in our attack tree, when we want to replace a gateway. This attack is possible if and only if the border gateways respect the source routing option. Therefore, the real vulnerability V_4 that enables the depicted attack pattern is the availability of the source routing option in the border gateways.

B. Step 2: vulnerability dependence graph

Summarising, we have identified five vulnerabilities. It is evident that they are not independent; in particular, V_{1I} , the weak authentication in an internal gateway, helps sniffing on every link connected to that gateway, i.e., V_2 .

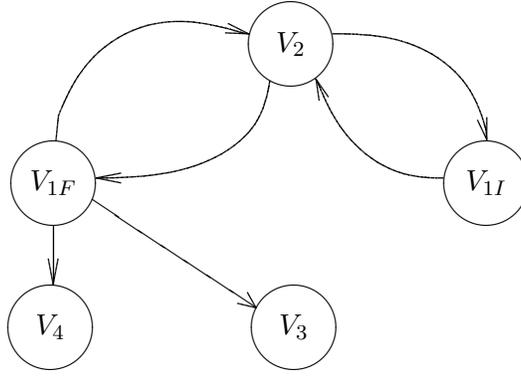


Fig. 5. The vulnerability dependence graph

On the contrary, whenever we are able to sniff the link, it becomes easier to gain control over the gateways, since we may benefit from additional information that weakens the gateway’s authentication. Obviously, the same holds for a border gateway, thus for V_{1F} , but remembering that sniffing cannot happen on the private networks.

It is also evident that a weak authentication on a border gateway (V_{1F}) eventually simplifies the weak setting of IP phones (V_3) as well as enabling source routing on the gateway (V_4) since, when one gains access to the gateway, he can manipulate its settings so to weaken the security posture of the whole private network.

There are no other significant relations between vulnerabilities, since internal gateways and Internet links cannot directly influence IP phones’ settings or a configuration option in a border gateway. In fact, in both cases the Internet component has to gain access to a border gateway, thus it has to exploit vulnerability V_{1F} .

Therefore, we are able to draw the vulnerability dependence graph, as shown in Figure 5.

We want fix a reference system to measure the risk connected to the various vulnerabilities: we use the range $0, \dots, 10$ where 0 means “impossible to violate” and 10 means “completely straightforward”.

The risks connected to every vulnerability are measured according to our metric and to the previously discussed qualitative weighting:

- the vulnerability V_{1I} , considered by itself, has a low risk thus we assign it the value of 2, then $E_0(V_{1I}) = 2$;
- similarly, V_{1F} has a low risk, $E_0(V_{1F}) = 2$;
- instead, V_2 is easy if it is possible to access the link we want to sniff; for this reason, in the wild Internet, we assume $E_0(V_2) = 5$;
- usually, IP phone address are almost public and few cares are taken to ensure that names or addresses of devices are difficult to guess, therefore we assume an high risk, $E_0(V_3) = 8$,
- finally, a gateway may or may not honour the source routing option in an incoming packet, depending on its configuration. Therefore, we set $E_0(V_4) = 6$ since we assume not to know what is the default behaviour when a gateway receives a packet with the source routing option.

The dependence between vulnerabilities is measured on the same metric, where the value represents how easier becomes to attack a vulnerability, broken the preceding one in the dependence graph.

The values are as follows, and the reader is invited to check that they strictly follows the qualitative judgement we have already discussed:

- the exploitability of V_{1I} when V_2 has been achieved is in the average, $E(V_{1I}|V_2) = 4$;
- the opposite is immediate, thus $E(V_2|V_{1I}) = 9$;
- the case of a border gateway is slightly more difficult, since the gateway is accessible only on the public side, hence $E(V_2|V_{1F}) = 9$ and $E(V_{1F}|V_2) = 3$;
- whenever we gain the control of a border gateway, we are able to simplify the attack to V_3 and V_4 , although not in a completely immediate way, thus $E(V_3|V_{1F}) = 8$ and $E(V_4|V_{1F}) = 7$.

The final result is shown in Figure 6.

C. Step 3: iteration and equilibrium condition

Propagating the dependencies as discussed in Section III-A on the graph, after a couple of iterations, yields the following fixed point: $E(V_{1I}) = 4$, $E(V_2) = 5$, $E(V_{1F}) = 3$, $E(V_3) = 8$, $E(V_4) = 6$.

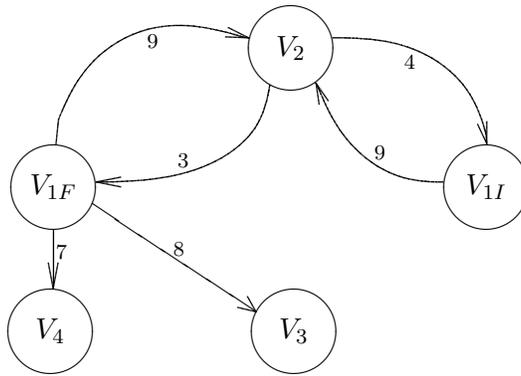


Fig. 6. The weighted vulnerability dependence graph

Given the depicted situation, a security administrator can propose different solutions in order to mitigate the total risk of the system. Since security solutions are usually expensive, it is very important to choose the most effective one, that reduces the risk to a minimum and, at the same time, has the minimal economical impact. To this aim, the possibility to quickly simulate and explore the impact of multiple actions allows the user to choose the best solution.

Vulnerability	Initially	$E(V_3) = 0$	$E(V_2) = 3$	$E(V_{1F}) = 0$
V_{1I}	4	4	3	4
V_{1F}	3	3	3	3
V_2	5	5	3	5
V_3	8	3	8	8
V_4	6	6	6	6

TABLE I

A COMPARISON OF DIFFERENT COUNTERMEASURES

Table I illustrates a variety of trials, simulating the effects of various countermeasures to mitigate the vulnerabilities.

The rationale is as follows:

- firstly, we use the heuristics *strengthen the weakest point*; in our setting, the weakest point corresponds to V_3 , and we suppose to completely eliminate this single vulnerability ($E_0(V_3) = 0$). The propagation of dependencies yields to a local improvement in the overall risk.
- another possibility is to apply the heuristics *strengthen the central point*; in our system, the central point is V_2 in the dependency graph, and we suppose to use encryption, e.g., using SSL, to mitigate the effects of sniffing ($E_0(V_2) = 3$). The propagation of dependencies shows an improvement, thus the countermeasure is convenient.
- the last trial consists in the impossibility to access a border gateway, that is $E_0(V_{1F}) = 0$. Unfortunately, the propagation of dependencies shows how this popular countermeasure is completely ineffective.

It is evident that operating on V_2 , V_3 and V_4 is the most convenient choice to reduce the risk measures; what is not immediately clear is that, because of the way the vulnerabilities influence one another, these are the *only* reasonable improvements to the overall security we can make. In fact, any other single or combined action that do not radically modify the picture, will lead to the optimal risk vector $E(V_{1I}) = 3$, $E(V_{1F}) = 3$, $E(V_2) = 3$, $E(V_3) = 3$, $E(V_4) = 3$, obtained, e.g., by posing $E(V_2) = 3$, $E(V_3) = 3$ and $E(V_4) = 3$.

The described scenario had a double purpose: to make evident that, without changing the existing VoIP protocols, it is difficult, but not impossible, to intercept phone calls. Although one can judge in a different way some of our initial measures in order to evaluate the risk of an external attack to a phone call based on standard protocols and devices, the final result is qualitatively the same: an optimal non-zero risk vector can be deduced. The conclusion after our risk analysis is that intercepting a phone call from “outside”, that is, in the wild Internet, with no access

to the protected networks where the phones reside, is difficult. Although this difficulty is not a surprise, since, as already remarked, VoIP calls are peculiar communications because they are real-time phenomena, thus imposing serious limitations to the range of possible attacking techniques, the surprising fact is that few countermeasures are really effective.

V. RELATED WORK

Voice over IP, convergence and real-time communication are concepts that introduce a revolution inside the ICT market. In literature there are many works [9], [14], [7] that show the advantages of this new way of communication.

On the contrary, the scientific community agrees that the real diffusion of VoIP is limited by security problems. For example, NIST [5] asserts that one of the main source of confusion for those new to VoIP is the natural assumption that because the digitised voice travels in packets, just like other data, existing network architecture and tools can be used without change.

Unfortunately, VoIP adds a number of complications to existing network technology. Thus, it is important to study an ad hoc security solution for VoIP system. In particular, NIST [5] explains the challenge of VoIP security and outline steps needed to help secure a VoIP network; M. Tanase [20] points out the principal threats and the consequential countermeasures; Bruschi et al. [3] analyse voice performance over IPsec. In literature there are many other works, e.g., [8], [13], [4] offering an overview about general threats and countermeasures.

On a slightly different line is [10], where the VoIP performance is evaluated when traditional security solutions (firewall, encryption, etc.) are adopted.

Our approach is different because, in agreement with the ones who consider security as a process characterised by ordered phases, we evaluate in a quantitative way the risk using a methodology defined in prior works. In order to specialise an abstract approach to a concrete case study, we chose to evaluate the risk of wire tapping.

Although it is evident that the chosen system is worth a risk assessment analysis, the reader may wonder why the methodology described in Section III is adequate. In general, risk, trust, security requirements mapping, and component interdependence are concepts that are linked together and have been widely discussed in literature: for example, Baskerville [2] describes the evolution of different methods to measure risk that sometimes could be used together to improve the result accuracy. Even though software security is extensively discussed in risk management methodologies [18], [12], [1], among information security experts there appears to be no agreement regarding the best or the most appropriate method to assess the probability of computer incidents [15].

In our opinion, the chosen method combines three components that are perfectly suited to the risk assessment analysis of VoIP systems:

- it is clean, i.e., an expert in the risk assessment field has to produce a well-understood set of security judgements, using a natural metric;
 - it is simple and efficient, i.e., it is based on a simple model of the system that closely resembles the global view of an expert, and it has an efficient algorithm that calculates the exploitability values in presence of dependencies among vulnerabilities;
 - it is mathematically well-founded, although we have not discussed these aspects in this work for conciseness.
- We plan to publish an in-depth analysis of the mathematical formalisation of the risk assessment methodology in the near future.

Therefore, the chosen risk assessment procedure produces the best result when it is applied in an environment where the experts can easily give a *relative* judgement on the severity of potential vulnerabilities, where the dependencies among vulnerabilities are locally clear, but globally obscure because of the complexity of the architectural level. Both requirements are completely fulfilled by the VoIP environment.

VI. CONCLUSION

This paper has discussed the problem of risk assessment in the case of a VoIP system. The analysis has been conducted by means of a case study: the interception of a VoIP phone call.

As a result we have shown both a risk assessment methodology applied to the case study and an analysis of the wire tapping risk.

As in any quantitative analysis based on experts' judgements, the exact values of the presented evaluations of the risks associated to each vulnerability are, to some extent, subjective. But the most interesting result of our analysis

is of a qualitative kind, and does not depend on the values we have used to compute: the result is the existence of an optimal risk vector different from the zero vector. The meaning of this result is that there is an intrinsic insecurity in the VoIP service under analysis.

We believe that the analysis of the discovered intrinsic insecurity deserves further study and a deep understanding of its architectural origin.

REFERENCES

- [1] C. Alberts, A. Dorofee, J. Stevens, and C. Woody. Introduction to the OCTAVE approach. *CERT Coordination Center*. Available at http://www.cert.org/octave/approach_intro.pdf, 2003.
- [2] R. Baskerville. Information system security design methods: Implications for information systems development. *ACM Computing Surveys*, 25(4):375–414, 1993.
- [3] D. Bruschi, R. Barbieri and E. Rosti. Voice over IPsec: Analysis and solutions. *Proceedings of the 18th Annual Computer Security Applications Conference*, 2002.
- [4] M. Evans, J.C. Straley, J. Larson and T. Dawson. Defending VoIP networks from DDoS attacks. *GlobeCom 2004 VoIP Security Workshop*, 2004.
- [5] S. Fries, D.R. Kuhn, T.J. Walsh. Security considerations of Voice over IP systems. *Computer Security Division Information Technology Laboratory National Institute of Standards and Technology (NIST)*, Gaithersburg, MD 20899-8930, Jan. 2005.
- [6] S.L. Garfinkel. VoIP and Skype Security, Jan. 2005.
- [7] B. Goode. Voice over Internet Protocol (VoIP). *Proc. of IEEE*, 90:1495–1517, Sep. 2002.
- [8] J. Halpern. IP telephony security in depth. White Paper, *Cisco Systems*, 2002.
- [9] W.C. Hardy. VoIP service quality: Measuring and evaluating packet-switched voice. *McGraw-Hill*, 2003.
- [10] P. Hochmuth and T. Greene. Firewall limits vex VoIP users. *Network World Fusion*, Jul. 2002.
- [11] M. Howard and D. Leblanc. Writing secure code. *Microsoft Press*, 2003.
- [12] B. Jenkins. Risk analysis helps establish a good security posture; Risk management keeps it that way. Whitepaper, *Countermeasures Inc.*, 1998.
- [13] M. Marjalaakso. Security requirements and constraints of VoIP. *Department of Electrical Engineering and Telecommunications, Helsinki University of Technology*, 2001.
- [14] P. Mehta and S. Udani. Overview of Voice over IP. Technical Report MS-CIS-01-31, *Department of Computer Information Science, University of Pennsylvania*, Feb. 2001.
- [15] G.P. Sharp, P.H. Enslow, S.B. Navathe and F. Farhmand. Managing vulnerabilities of information system to security incidents. *ACM International Conference: 5th international conference on Electronic commerce*, 2003.
- [16] S. Sicari, D. Balzarotti and M. Monga. Assessing the risk of using vulnerable components. *Quality of Protection. Security Measurements and Metrics*. Springer, 2006.
- [17] Internet Protocol. RFC 791, 1981.
- [18] T. Siu. Risk-eye for IT security guy. *Gsec*, 2004.
- [19] W. Stallings. Cryptography and network security: Principles and practices. Pearson Education, 2003.
- [20] M. Tanase. Voice over IP security. *Security Focus*, 2004.