The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0968-5227.htm

Expanding topological vulnerability analysis to intrusion detection through the incident response intelligence system

Dimitrios Patsos, Sarandis Mitropoulos and Christos Douligeris
Department of Informatics, University of Piraeus, Piraeus, Greece

Received 23 December 2009
Revised 10 May 2010
Accepted 10 May 2010

Abstract

Purpose – The paper proposes looking at the automation of the incident response (IR) process, through formal, systematic and standardized methods for collection, normalization and correlation of security data (i.e. vulnerability, exploit and intrusion detection information).

Design/methodology/approach – The paper proposes the incident response intelligence system (IRIS) that models the context of discovered vulnerabilities, calculates their significance, finds and analyzes potential exploit code and defines the necessary intrusion detection signatures that combat possible attacks, using standardized techniques. It presents the IRIS architecture and operations, as well as the implementation issues.

Findings – The paper presents detailed evaluation results obtained from real-world application scenarios, including a survey of the users' experience, to highlight IRIS contribution in the area of IR.

Originality/value – The paper introduces the IRIS, a system that provides detailed security information during the entire lifecycle of a security incident, facilitates decision support through the provision of possible attack and response paths, while deciding on the significance and magnitude of an attack with a standardized method.

Keywords Data security, Computer crime, Risk management

Paper type Conceptual paper

1. Introduction

A weakness in a system, whether in software, hardware or firmware, that allows an attacker to violate the integrity, the confidentiality, the access control, the availability, the consistency, the audit mechanism of the system, the applications it hosts as well as the data it manages is described by the term vulnerability. A more formal definition states that "a vulnerability is not lack or want, but defenselessness, insecurity, and exposure to risk, shocks and stress" (Chambers, 2006). There have been important attempts to measure and/or estimate the importance and magnitude of a vulnerability, like those described in Hansman and Hunt (2005), Killourhy *et al.* (2004), Kim *et al.* (2008), which concluded that the actual significance of each vulnerability can be estimated only in the context of the information environment this particular vulnerability resides within.

For example, a system offering Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP) and Simple Mail Transfer (SMTP) services may be subject to a number of vulnerabilities that could allow for remote code execution. In the case where this system is protected by a properly configured firewall if everything but the HTTP traffic has been blocked, the vulnerabilities on SMTP and FTP protocols have an entirely different security significance; the existence of this firewall mechanism allows us to positively assume that both SMTP and FTP services cannot be remotely exploited, since no traffic

Information Management & Computer Security
Vol. 18 No. 4, 2010
pp. 291-309
© Emerald Group Publishing Limited
0968-5227
DOI 10.1108/09685221011079207

IPR2025-01397
CrowdStrike EX1006 Page 1

is permitted for SMTP and FTP communications. However, the use of a security mechanism (such as a firewall) does not imply that the risk is eliminated. A notable example is the attacks destined to web-based applications, such as cross-site scripting and Structured Query Language injection (SANS, 2009). In these attacks, a malicious user can access backend systems (such as application and database servers usually residing inside the network perimeter defined by the firewall) by crafting an attack on the HTTP protocol (usually permitted by most firewalls; Kieyzun *et al.*, 2009).

The systematic examination of a system to identify the infrastructure or related components that may be at risk from attacks and the determination of appropriate procedures that can be implemented to reduce this level of risk is known as vulnerability assessment (VA). VA is the process through which vulnerabilities are discovered and (in some cases treated) at operating system, network and application level within a given infrastructure. However, it is not only an exploit that could realize a particular vulnerability and the possible security countermeasures that are trying to prevent this from happening; it is also the combination of various vulnerabilities that form attack paths, especially in multi-staged and blended attacks.

Topological vulnerability analysis (TVA), as defined in Jajodia and Noel (2009), is the process that analyzes vulnerability dependencies and produces all the possible attack paths in a network. An attack path is defined as the exploitation of one or more vulnerabilities in a predefined order, i.e. the road an attacker follows to reach high-level security objectives by using low-level attacking techniques. For example, in the last years, the blended attacks concurrently exploit a combination of two, three or perhaps more vulnerabilities that may exist in a networked environment with a predefined order (Chien and Szor, 2002; Conklin, 2008), forming different attack paths. Until recently, most of the commercial or open-source automated vulnerability scanners that are used in the industry lacked the ability to examine the interdependency of a series of vulnerabilities and the ability to provide a cross-reference of known vulnerabilities to known exploits, and the ability to cooperate with security monitoring/response policies (Debar *et al.*, 2007). Recent advances in a very limited set of commercial automated VA tools include links to exploit code of a discovered vulnerability. However, none of these tools, at least up to now, correlate vulnerabilities with exploit code, or provide intrusion detection and prevention (IDP) signatures required to detect and/or block those threats. The latter still remains a human-oriented task, usually performed by high-skilled security analysts.

On the other hand, exploits (or exploit patterns) are known to IDP systems that make use of signatures. Signatures combat known-attacks using statistical and/or behavioral analysis, which make them capable of detecting slight variations of known-attacks as well (also referred to as unknown attacks in the security industry). In most cases, an IDP signature includes the exact piece of code of an exploit, or a pattern of this exploit. This code (or most commonly a hash of it) is compared by the IDP system against the payload of every packet in transit within the policy domain this signature is effective. When a certain threshold of similarity (according to what the particular security policy indicates) is reached, an alarm is triggered. Using these techniques IDP systems can detect violations of a system's security policy (Srilatha *et al.*, 2004). Moreover, IDP systems can also verify, itemize and characterize threats from both outside and inside a corporate network, assisting in the decision-making process of allocating computer security resources (Scarfone and Mell, 2008). Nevertheless, IDP are known to suffer

IPR2025-01397
CrowdStrike EX1006 Page 2

from large false positive ratios that, if not treated accordingly by continuous fine-tuning procedures, may turn these systems to produce useless security information, also known as security noise (Aberdeen Group, 2003; Mitropoulos *et al.*, 2007; Tian *et al.*, 2008).

2. Overview

In this paper, we propose the incident response intelligence system (IRIS) that works towards the elimination of the gap between automatic vulnerability identification; exploit verification and intrusion detection (Jajodia and Noel, 2009). IRIS is a security-management system that correlates vulnerability, exploit and IDP information in real time, leveraging the policy effectiveness of VA tools and IDP systems since it reduces false negatives and false positives, respectively. Beyond this, the IRIS operations can provide the IR specialists with more accurate information regarding the scope and magnitude of an attack, by significantly reducing the volume of data that need to be examined by security experts in order to reach more rapid and concrete conclusions (Adelstein, 2006), as well as by providing a numerical value on an attack's significance and impact. Finally, IRIS can contribute towards the automatic creation of IDP policies.

We present IRIS through the formal definition of attack and response paths in the context of TVA (Jajodia *et al.*, 2006). We then introduce the basic operations of IRIS by presenting its workflow of operation. IRIS automatically correlates vulnerability, exploit and intrusion detection information, thus constructing attack and response paths. A response path contains the appropriate policy definitions to IDP systems. We also discuss the benefits of IRIS on IR and digital forensics.

A detailed discussion of the architectural and implementation issues addressed in the development of the current version of IRIS is provided. The system modules, along with the interrelationships between these modules, the functions every module contributes to the overall operation, as well as the limitations of the current version are given. Moreover, the operation of IRIS is presented through a case study. Finally, the use of IRIS is evaluated through a user survey.

3. Related work

Templeton and Levitt (2000) describe a flexible extensible model for computer attacks, including a language for specifying this model, along with proposing applications of this model in vulnerability analysis and intrusion detection. Sheyner *et al.* (2002) provide a tool for the assessment of security attributes and vulnerabilities in computer networks which also correlates attack graphs with the most exploitable components of the system configuration. Swiler *et al.* (2001) also provide a tool that generates and analyzes attack path information based upon symbolic model checking algorithms. Ammann *et al.* (2002) provide a more compact and scalable representation of attack graphs, relying on an explicit assumption of monotonicity, which, in essence, states that the precondition of a given exploit is never invalidated by the successful application of another exploit. This approach focuses on chaining together the vulnerabilities uncovered by VA tools in order to reveal end-to-end attack scenarios.

The works discussed above primarily focus on modeling network and computer-based attacks and on producing attack paths and/or graphs rather than on providing a system that uses this information in order to implement matching scenarios with intrusion detection policies. A series of works have recently attempted to address the latter.

IPR2025-01397
CrowdStrike EX1006 Page 3

For example, Gula (2009) has illustrated a number of scenarios that could be used in correlating intrusion detection logs with vulnerability information. This approach aims to reduce the false positive information provided by the IDP, but lacks "filtering" mechanisms on the information used as input in the VA tools and in the IDP systems (i.e. topological information), as well as it lacks the use of exploit information. Ning and Xu (2003) have presented techniques to automatically learn attack strategies from correlated intrusion alerts by a model that represents an attack strategy as a graph of attacks with constraints on the attack attributes and as a temporal order among these attacks. This work presents a methodology to correlate intrusion detection information with static VA input, without modeling network security conditions and exploits or analyzing the exploit sequences (attack paths) that lead to specific attack goals. The latter is described by Jajodia *et al.* (2006) where the relationships of vulnerabilities and exploits with IDP signatures are not given specific consideration.

Papadaki and Furnell (2006) have proposed an architecture of a flexible and intelligent-automated response system that is able to adapt response decisions based on the context of a detected incident. Response decisions are based on expert knowledge, but no vulnerability and signature correlation is included in the approach.

Debar *et al.* (2007) propose an architecture that allows the dynamic and automatic deployment of a generic security policy in specific policy instances by taking into account the threat level that has been identified by the available IDP.

Finally, O'Hare *et al.* (2008) developed a visualization tool for TVA, which is based on a database of known vulnerabilities and exploits. The tool represents, in a graph form, the possible attack combinations on a predefined networked infrastructure. This system does not include, though, intrusion detection data.

4. Defining topological security information

A vulnerability is a potential hazard that may exist in nearly every piece of software. There have been a number of research efforts that try to classify or model vulnerabilities, such as Killourhy *et al.* (2004), Hansman and Hunt (2005), but to the best of our knowledge no formal or standard method exists today. Moreover, the vulnerability reporting community has not yet defined a standardized language to describe vulnerability semantics even though there exist interesting approaches, like Jajodia *et al.* (2006), Yegneswaran *et al.* (2005), Brumley *et al.* (2006, 2007).

On the other hand, the security community provides a large set of reference vulnerability lists like CERT/CC (US-CERT, 2009), Mitre's Corporation common vulnerabilities and exposures (CVE; MITRE, 2009), SecurityFocus (2009) and BugTraq (2009), and major vendors like Microsoft (2009) and Cisco (2009) maintain and publish their own vulnerability data. Moreover, various IR teams and analysis centers around the world provide their own vulnerability information. Nevertheless, there is a debate regarding the uniqueness of the information they provide (Schultz, 2004). Most open-source and commercial VA products like Nessus (Tenable, 2009), Foundstone (McAfee, 2009) and Internet Security Systems (IBM, 2009) either provide some sort of proprietary vulnerability description or point to some of the aforementioned lists. For the rest of our discussion, Mitre Corporation's CVE reference list will be used as a de-facto standard, following the wide adoption this list has gained by the security research community during the last years.

IPR2025-01397
CrowdStrike EX1006 Page 4

A piece of code that realizes the existence of a particular vulnerability and takes some action is called an exploit. An attack path can be seen as the series of consecutive vulnerability exploits that allow an attacker to reach his objectives.

We formally define an *attack path* a_i as the unique sequence of successful vulnerability exploits[1], that is $a_i = \{(e_l|v_m), \ldots, (e_j|v_k)\}, \forall a \in A, \forall e \in E, \forall \in v\ V, i, j, k, l, m, n \in N$, where A is the set of *attack paths* a_i, E the set of *exploits* e_j and V is the set of *vulnerabilities* v_k.

Every exploit corresponds to one or more vulnerabilities and one vulnerability can be realized by more than one exploits (i.e. there exists a many-to-many relationship between vulnerabilities and exploits). Figure 1 shows an example attack path, including the relationships between v_k and e_j, where the dependency of $e_1, e_2, \ldots, e_j$ with $v_1, v_2, v_3, \ldots, v_k$ is depicted. In this example, v_1 can be exploited by e_1, v_2 by e_1 and e_j, v_3 by e_2 and e_j, while v_k can be realized by e_2 and e_j. Therefore, the attack path a_1 is the series of v_k exploited by an attacker using e_j. For more complicated attack scenarios, the attack path is expected to be significantly more complex. For a more coherent attack path presentation, the reader is referred to Krasser *et al.* (2005).

Information regarding exploits can be found either directly in a large number of underground web sources (for similar purposes) or indirectly, as part of a security application, like in signatures in antivirus programs or in IDP systems.

TVA (Jajodia and Noel, 2009), via the construction of the associated attack paths, is of major importance when the corresponding response paths can be also dynamically constructed. We define a response path as the set of IDP signatures that comprise the IDP policy for a particular attack path. Formally, the response path r_i that corresponds to attack path a_i, is defined as $r_i = \{(e_k / s_k), \ldots, (e_i / s_i), \forall r_i \in \mathrm{R}, \forall a_i \in \mathrm{A}, \forall s_i \in \mathrm{S}, \forall i, k \in \mathrm{N}$, where R is the set of response paths r_i and S the set of IDP signatures s_i.

Typically, signatures s_1, s_2 that correspond to exploits $e_1, e_2, \ldots, e_j$ (therefore preventing the realization of vulnerabilities v_1, v_2, v_3, v_k) form the response path depicted as r_1 in Figure 2.

5. The incident response intelligence system

5.1 Introduction

The IRIS is a security management system, with applications to network vulnerability analysis, IR and the tuning of intrusion detection systems. The system collects security information related to vulnerabilities, exploits and IDP signatures from a large number

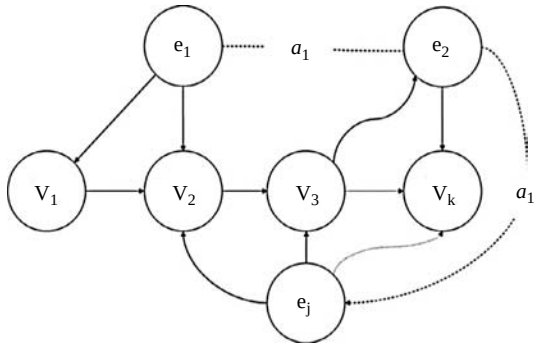

Figure 1.
A sample attack path (a_1) and the related vulnerabilities (v_k) and exploits (e_j)

IPR2025-01397
CrowdStrike EX1006 Page 5

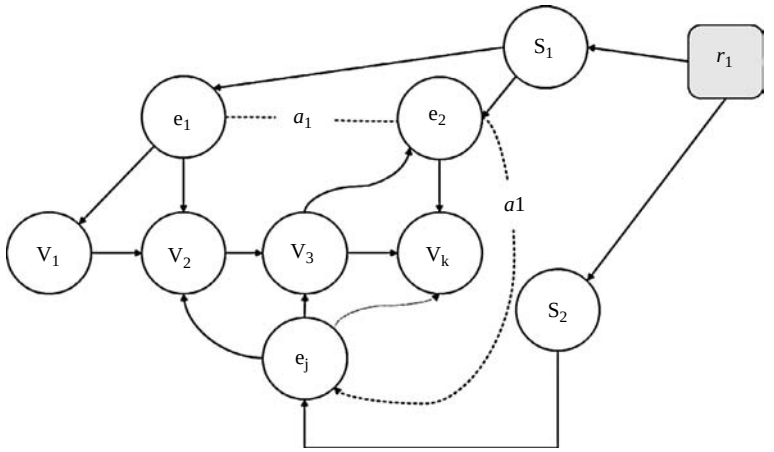

Figure 2.
A sample response path (r1)

of sources (web, mailing lists, hacking web sites, security systems, vulnerability scanning reports, etc.) and stores them in a number of local databases. This vulnerability information is then normalized (i.e. redundant data are de-duplicated) and correlated with relative exploit code and IDP signatures. When IRIS parses a report from a VA tool, it can construct the attack and response paths for the identified vulnerabilities using TVA techniques (Jajodia and Noel, 2009), score every identified vulnerability using CVSSv2 (Mell *et al.*, 2006) and provide the IDP signature(s) required to combat this particular attack.

The system can be either used as a proactive security mechanism (i.e. during the vulnerability analysis of a network infrastructure) or as a response security mechanism (i.e. when an attack is in progress), as described in "Section 4.3". Moreover, as shown in "Section 5", the system can be used for efficiently and effectively tuning the IDP systems according to the unique security requirements of the network infrastructure.

In the following section, we present the system architecture as well as the security services the system provides.

5.2 System architecture and operations

An overview of the IRIS modules is presented in Figure 3, while the system modules and operations are analyzed in the following sections.

5.2.1 Collection of security information. The system collects and stores well-documented vulnerability information from a large number of sources. We use the Mitre Corporation's CVE list as a primary source for vulnerabilities. Other recognized lists, including vulnerability lists maintained by major vendors (e.g. Microsoft, Cisco Systems, Check Point Software Technologies, etc.), analysis and response centers (i.e. US-CERT, FIRST, etc.) and/or independent security researchers are also consulted. These information sources are frequently maintained and updated by their respected owners. IRIS frequently updates its vulnerability database accordingly.

Information related to exploits (and actual exploit code) can be obtained from hacking web sites, from alternative sources (such as security tools) as well as from specific mailing lists. Although IRIS does not rely on the validity or the accuracy of this

IPR2025-01397
CrowdStrike EX1006 Page 6

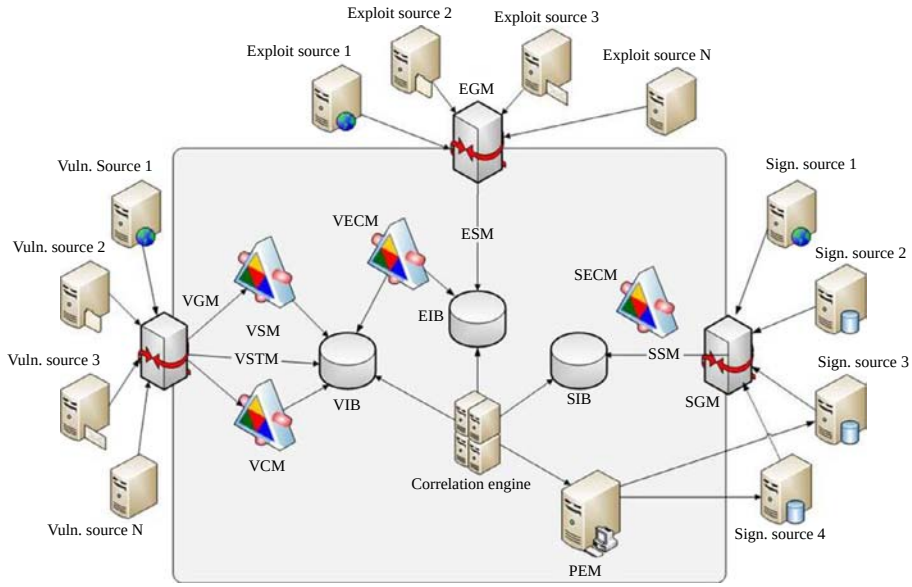

Figure 3.
An overview of IRIS modules

information (since hacking web sites provide "volatile" information for a limited period of time), this information is useful for further vulnerability analysis as well as for vulnerability scoring (described in following section).

Information related to IDP signatures can be obtained from IDP systems, IDP vendor web sites, associated mailing lists as well as the open source research community.

By the time of this writing, IRIS collects information (through the vulnerability gathering module – VGM) from 18 vendor web sites, 12 coordination centers and 12 independent lists, therefore maintaining more than 33,000 vulnerabilities in the vulnerability information base (VIB). The signature-gathering module (SGM) collects information from two IDP systems, maintaining more than 16,000 different IDP signatures in the signature information base (SIB). The exploit-gathering module (EGM) is gathering information from three different sources, maintaining more than 2,000 exploits in the exploit information base (EIB). These numbers increase frequently, through the updating of the respective databases.

In IRIS, these operations are performed by a large set of processes and their corresponding modules. The vulnerability gathering process from various sources (e.g. the web, mailing lists and VA tools) is performed by VGM, while the secure storage process to the VIB is performed by the vulnerability storage module. The exploit information gathering process, from various sources, is performed by EGM, while the secure storage process to the EIB is performed by exploit storage module. Finally, the signature gathering process from the IDP signature database is performed by SGM, while the secure storage process to the SIB is performed by the signature storage module.

5.2.2 Normalization and correlation of security information. The normalization and the correlation of the collected vulnerability information is of major significance, since the security community describes a particular vulnerability in many different ways

IPR2025-01397
CrowdStrike EX1006 Page 7

and various conventional names. For example, the vulnerability described as 2000-0246 by the CVE list is described as MS00-019 by Microsoft, while the CVE-2003-0100 vulnerability corresponds to what is described as 20030507 by Cisco Systems. IRIS also provides a hyperlink to the aforementioned sources, mainly for on-demand real-time validation of the IRIS local databases with the information provided by the information sources. The CVE identifier is used as a "primary" key for all the vulnerabilities that are maintained in the system databases.

In IRIS, these operations are performed by the following processes and corresponding modules:

- the vulnerability correlation process performed by the vulnerability correlation module (VCM);
- the correlation of exploits to vulnerability identifiers is performed by the vulnerability exploit correlation module (VECM); and
- the correlation of signature information to exploits is performed by the signature exploit correlation module (SECM).

The correlation modules (VCM, VECM and SECM) are parts of the system's correlation engine. This engine expands the TVA presented in the previous sections, by providing a full correlation between all attack components, i.e. the vulnerabilities, the exploits that can potentially realize these vulnerabilities and the associated countermeasures (IDP signatures) that prevent this from happening. This is achieved not only through the construction of the attack path that describes the steps of an attack but also through the construction of the response path that describes the associated responses.

5.2.3 Vulnerability scoring. The system calculates the significance of all the vulnerabilities stored and maintained in the IRIS databases. The objective is to provide an initial metric for the significance of a particular vulnerability, in order to facilitate appropriate and timely response. Vulnerability scoring is a non-trivial task since it is heavily dependent on the security context this vulnerability has been discovered. In other words, vulnerability scoring is calculated using base, temporal and environmental metrics, through an open and easily understandable method. For these reasons, IRIS uses the National Infrastructure Advisory Council's CVSSv2 system, which is supported by major organizations and internet vendors, like CERT/CC, Cisco Systems, Department of Homeland Security, Mitre Corporation, eBay, IBM Internet Security Systems, Microsoft, Qualys, Symantec and others. CVSS is currently maintained by the forum of IR teams (FIRST, 2007).

The base metrics describe the native vulnerability characteristics that do not change over time, such as the access complexity, the authentication levels, the impact on confidentiality/integrity/availability, etc. The temporal metrics describe the vulnerability characteristics that change during the vulnerability lifecycle, such as the exploitability, the remediation level and the report confidence.

Finally, the environmental metrics describe the characteristics of a vulnerability that are affected by the environment a vulnerability resides, such as the collateral damage potential and the target distribution as well as the confidentiality, integrity and availability requirements of this particular environment. CVSSv2 uses a set of predefined values for these characteristics, while the calculation of the base, temporal and environmental vulnerability score is the result of an extremely complex operation. The overall vulnerability score is the environmental score, when such

IPR2025-01397
CrowdStrike EX1006 Page 8

information is provided. When this is not the case, the overall vulnerability score is the temporal score, when such information is provided. Finally, if neither environmental nor temporal metrics are provided, the overall vulnerability score is the base score of a vulnerability (the latter is provided by some automated VA tools). IRIS automatically expands this capability by automatically calculating the base and temporal scores using the correlated information maintained in the system databases, while providing an interface to the end-user to provide the environmental metrics (where necessary). This human interaction with the system, through the provision of the environmental metrics, provides the overall vulnerability score.

In IRIS, these operations are performed by the vulnerability scoring module. Figure 4 shows a snapshot of the system's capabilities to collect, normalize and correlate security information, including the scoring of a particular vulnerability. In brief, section "Host identified" presents the lists of hosts and the corresponding vulnerabilities (after automatically parsing a VA report; Tenable, 2009).

Moreover, by clicking on a particular vulnerability, the IRIS administrator can find the associated exploits ("Exploit Information" section) as well as the applicable signatures for this particular vulnerability ("Applicable Signatures" section). In this example, the Snort signatures were used as IDP identifiers. Moreover, the system provides a direct link to the actual exploit and the IDP signature code.

Finally, the "Vulnerability Information" section presents a series of various vulnerability information indices (such as CVE ID, severity, status, date of publication, date of last modification, etc.) as well as the overall CVSSv2 score, based on the base and temporal metrics. The temporal metrics automatically provide the overall vulnerability score. In other words, IRIS automatically uses the normalized and correlated information related to this particular vulnerability to calculate the base and temporal score, in order to provide the system administrator with an initial estimation of the severity and magnitude of the vulnerability. Moreover, the system also provides

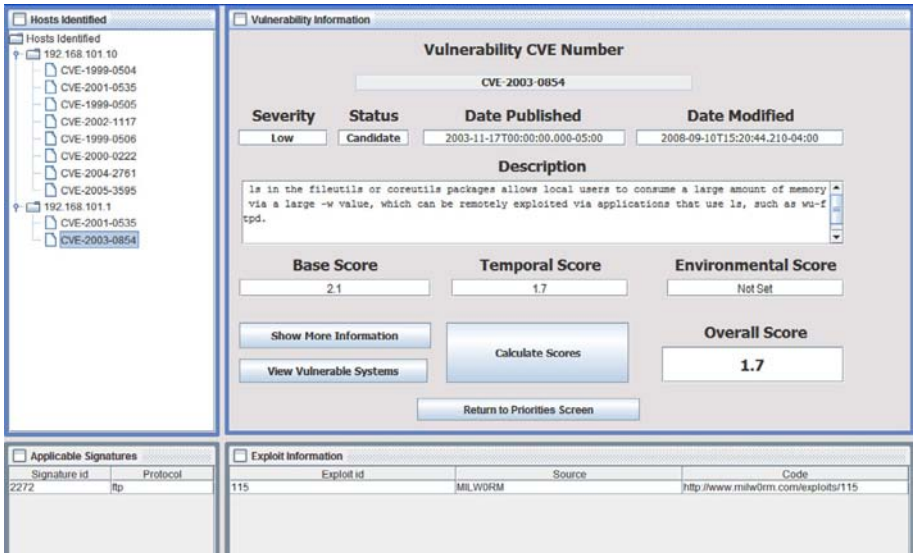

Figure 4.
Correlated vulnerability information and vulnerability scoring in IRIS

IPR2025-01397
CrowdStrike EX1006 Page 9

the IRIS administrator with an assessment of the environmental metrics, to facilitate the estimation of the potential damage. Through this process, the overall significance of a vulnerability is calculated using a combination of automatic system functions, while can be verified, for a particular infrastructure, by the IRIS administrator, using simple risk-analysis functions.

5.2.4 Construction of attack and response paths. When the VA report has been successfully parsed by the system, the correlation engine assumes the responsibility to correlate the information stored in all the databases (SIB, EIG and VIB) and to provide a mapping between vulnerabilities, exploits and IDP signatures. This output includes not only the relationship between vulnerabilities and exploits, but also the relationship of IDP signatures with exploits and, in turn, with vulnerabilities. The latter is depicted in Figure 5, which provides a snapshot of the IRIS ability to construct and associate attack and response paths.

The main objective of this process is to enhance the TVA by providing a complete mapping between vulnerabilities, exploits and signatures that should describe not only the attack path an attacker can use (by exploiting a series of vulnerabilities that are present in different system boundaries, i.e. in the perimeter and in the internal systems) but also a response path that should describe the corresponding response policies that have to be enforced in order to detect – and perhaps block – these actions.

A response path (r_i), as described before, contains the necessary IDP signatures that correspond to an attack path (a_i). It is expected that for every a_i at least one IDP signature will be used. In simple cases, one can positively assume that a single signature combating a particular exploit on the attack sequence could effectively terminate this attack (since the attack cannot be continued from this point onwards). Although this could be a highly possible scenario, the generalization of this concept cannot be effectively applied to multistage and blended attacks, where the exploitation of a particular vulnerability in an attack path initiates subsequent attack paths (Mathew *et al.*, 2005).

It is expected that a 1-1 relationship should exist between attack and response paths, and an entire attack path should correspond to a number of different IDP signatures that address it, as shown in Figure 5. The set of IDP signatures (s_i) used for the construction of r_i is the IDP policy for this attack scenario (a_i).

Figure 5 presents the attack and response paths for the vulnerabilities discovered in a predefined set of hosts. The "CVE Number" column lists all the vulnerabilities identified for this particular network infrastructure. The "Attack Path" column presents the successful exploitation (i.e. privilege escalation, system access, etc.) of these

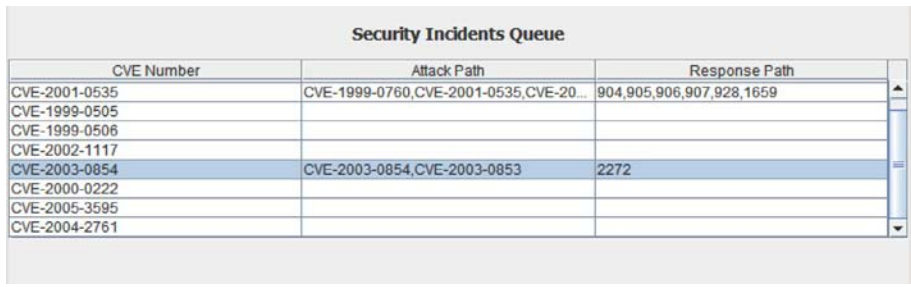

Figure 5.
Construction of attack and response paths

IPR2025-01397
CrowdStrike EX1006 Page 10

vulnerabilities with a predefined order. The "Response Path" column presents the associated response path (the set of IDP signature identifiers).

5.2.5 Policy enforcement and IDP re-configuration. The policy enforcement module (PEM) is responsible for analyzing the set of signatures an IDP policy is using, along with the structuring and enforcing policies after the attack and response paths have been identified, thus optimizing the IDP configuration.

PEM identifies and selects the minimal set of signatures (from the signature database of the IDP) that are needed to detect and/or prevent a specific attack path. Currently, the activation of this minimal set of signatures on the IDP system is performed manually, since the automatic activation of IDP signatures (especially to block traffic) has to be done with great confidence to avoid blocking legitimate traffic. We are currently working on automating this process for specific IDP systems.

This feature allows for interesting configuration scenarios in IDP systems, especially during times where prompt responses are required (or changes to the IDP policy are necessary) when an attack not addressed by the IDP is in progress.

5.3 The use of IRIS in IR

IRIS can be effectively used in corporate IR programs, since the main services of the system are the collection, normalization and correlation of security information related to an incident's lifecycle. In this context, IRIS can be used as a reference system where information regarding the risk analysis of an incident and the associated responses are required.

Many modern corporate IR methodologies are comprised of six distinct phases, referred to as preparation, identification, containment, eradication, recovery and follow-up (NIST, 2004; Mitropoulos *et al.*, 2006). The use of IRIS during the preparation phase can assist system administrators in tuning their VA tools and IDP systems so that false negatives and false positives are minimized. During the identification phase, the IRIS administrator can refer to the system in order to find information related to the root cause of an attack (i.e. the unadressed by the IDP systems vulnerabilities that were exploited), using information contained in VIB, EIB and SIB. Moreover, he or she can estimate the magnitude and the severity of this incident by invoking the vulnerability scoring funtionality provided by the tool, verify the actual situation (i.e. confirmed vulnerability, proof of exploit, use of IDP singatures, etc.) and interact with the system in order to provide the environmental factors (in a risk assessment manner) to calculate the overall score.

The latter is of crucial significance, since most of the response actions are driven by the conclusions derived during the identification phase. Moreover, during the containment and eradication phase of a security incident, the IRIS administrators can quantify the effectiveness of the IDP systems policy by analyzing attack and response paths. In brief, the IRIS administrator can use the response path which contains the entire set of IDP signatures needed to combat a particular attack.

For the recovery and follow-up phase, IRIS can provide correlated information that can be used for the documentation of the attack characteristics, including the significance and the magnitude of this specific incident, the IDP signatures activated, etc.

5.4 Limitations

IRIS is a management tool that does not intend to replace VA and IDP functionalities. On the contrary, the system's functionality is affected by the effectiveness of these

IPR2025-01397
CrowdStrike EX1006 Page 11

mechanisms (VA and IDP). For example, in case an IDP system's signature database has not been updated for a long time, IRIS is unlikely to provide any response paths for vulnerabilities related to these particular signatures. Although there is no deterministic way to bypass this obstacle, we are working on developing information freshness and liveness mechanisms that could potentially feed our system with the missing data. Moreover, we are also working in integrating our system with automated signature generation mechanisms, as those described in Newsome and Song (2005), Brumley *et al.* (2006) and Cui *et al.* (2007).

IRIS can construct attack and response paths using the CVE identifier of vulnerabilities as a primary key. In case of an unknown (or undocumented) vulnerability in terms of the CVE list, our system cannot provide any kind of useful information. However, taking into account the rather vast adoption of the CVE list from both the security industry and the community, this specific issue does not pose a significant threat to our system, at least for known vulnerabilities.

6. System evaluation and results

We evaluated IRIS using a series of experimental and real world data. In the latter case, we tested and evaluated IRIS against an internet infrastructure (a total of four hosts), a server infrastructure (a total of 22 hosts) and a subset of the desktop infrastructure (a total of 18 hosts) of a major private Greek Bank.

We used the existing IDP configuration of this particular infrastructure, enumerating and documenting the number of different IDP signatures that were used before the use of IRIS. Then, we performed three VA scans in three different subnets (using Nessus), and imported the scan report in IRIS, which calculated a new set of signatures, as depicted in Table I, using the TVA functions we described in "Section 5.2"). Finally, we activated the new set of signatures in those subnets and re-scanned the infrastructure, cross-correlating our attained results with the traffic detected by the IDP systems and the alerts created.

6.1 Real world results

Before evaluating IRIS, the Bank's administrators identified the absolute minimum of networking protocols and services offered by every one of the aforementioned systems and fine-tuned their IDP systems by activating the entire set of (Snort IDP) signatures for these services. Without deactivating any security mechanism in place, we scanned every distinct subnet (internet, server and desktop, respectively) using the most-updated version of Nessus. For every subnet scanned for vulnerabilities, we tuned Nessus accordingly, using a combination of policies, credentials, plugins and settings to make the scan as accurate as possible, and to eliminate unnecessary and unsafe checks. We deliberately omitted from our scanning procedure all

Subnet name	Number of hosts	Number of offered services	Number of IDP signatures (before IRIS)	Number of IDP signatures (after IRIS)	Percentage of the initial signatures (after IRIS)
Internet	4	6	401	4	1
Server	22	7	937	136	15
Desktop	18	10	1,147	1	0.1

Table I.
Real world results

IPR2025-01397
CrowdStrike EX1006 Page 12

the vulnerabilities related to the host operating system (in particular those addressed by host-based IDP systems, since Snort is a network-based IDP system and cannot detect such malicious traffic).

The objective was to identify all the possible attack paths for malicious traffic in every subnet and to construct the corresponding response paths, and subsequently to compare the number of the necessary IDP signatures provided by IRIS to the number of IDP signatures used by the Bank.

Results presented in Table I proved that the use of IRIS greatly reduced the number of necessary IDP signatures for a particular infrastructure. The main reason for this is the correlation of identified vulnerabilities using TVA techniques, along with the construction of the respective attack and response paths. Before the use of IRIS, in this particular infrastructure, the entire set of Snort signatures for the services running on this infrastructure was activated, in order to detect malicious traffic on those services.

In the tests performed, Nessus was the main tool for the collection of vulnerability information. The underlying security infrastructure of the organization (e.g. network separation, firewalls and IDP systems) greatly affected the scanning results. In other words, there were particular systems where Nessus could not discover any potential vulnerability in order to allow IRIS constructing attack and response paths. This is explained by the fact that the security mechanisms in place were agnostic to the scanning tool during the VA process. The presence of a properly configured security mechanism, such as a firewall, significantly reduced the possibility of a security incident on the limited set of permitted services. The latter is more obvious on the server and desktop subnets as presented in Table I. Finally, we also observed that the minimal set of IRIS-proposed IDP signatures also reduced the false positive ratio of IDP signatures dramatically, a fact which was not – however – among our evaluation objectives.

6.2 Users experience

The functionality of IRIS was also evaluated by 20 security professionals of the Greek information security arena, directly related to the security functions of their academic, banking, government and private sector organizations. The size of those organizations varied from 20 to 15,000 hosts. After the functionality evaluation by the experts, we ran a survey to evaluate the major features of IRIS, including development and performance issues. The most important findings of our survey are presented in Figures 6-9.

The majority of the security professionals responded with a rather positive opinion for the IRIS interface, as shown in Figure 6, overcoming one of our main concerns for the proper presentation of a vast amount of information managed by the system. The current version of the system uses a Java client as the interface with the IRIS administrator, which led many of the professionals to propose the use of a web interface. However, and despite the use of Java, the vast majority of the professionals commented very positively on the overall performance of the system, as shown in Figure 7. Moreover, taking into account that virtually all the security professionals participated in this survey by executing and evaluating IRIS in complex corporate environments (e.g. different security mechanisms and policies used, different network bandwidth capabilities) the results on the IRIS performance are more than encouraging.

Regarding the overall IRIS capabilities as an IR system, most of the security professionals responded positively, as Figure 8 shows. Taking into account the demographics of our survey (i.e. experience with IR and relative areas,

IPR2025-01397
CrowdStrike EX1006 Page 13

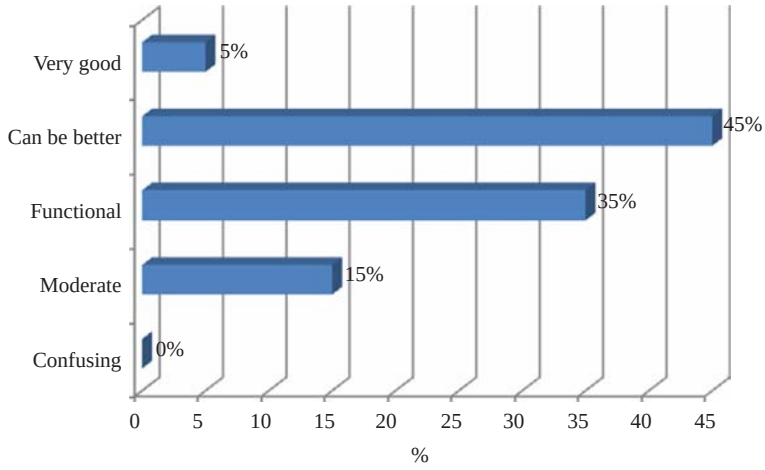

Figure 6.
IRIS interface

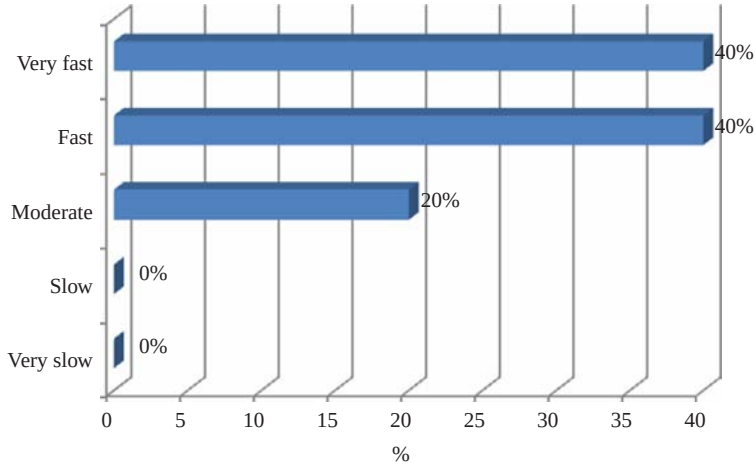

Figure 7.
IRIS performance

size of organizations, etc.) one could conclude that our proposed system could easily fit within a corporate IR program.

Finally, regarding the system's capabilities of collection, normalization, correlation and scoring of vulnerability, exploit and IDP information in the context of corporate IR, the vast majority of the professionals found the system very useful, as shown in Figure 9. The latter, combined again with the demographics of our survey, provides us with confidence that IRIS addresses the main issues of IR with a solid, well-established and structured approach.

7. Conclusions and future work

Within our next immediate steps is the development of a role-based access control framework, for the IR contacts, as the latter are defined in Mitropoulos *et al.* (2006), which would contribute to the management aspect of IR. Moreover, we also work

IPR2025-01397
CrowdStrike EX1006 Page 14

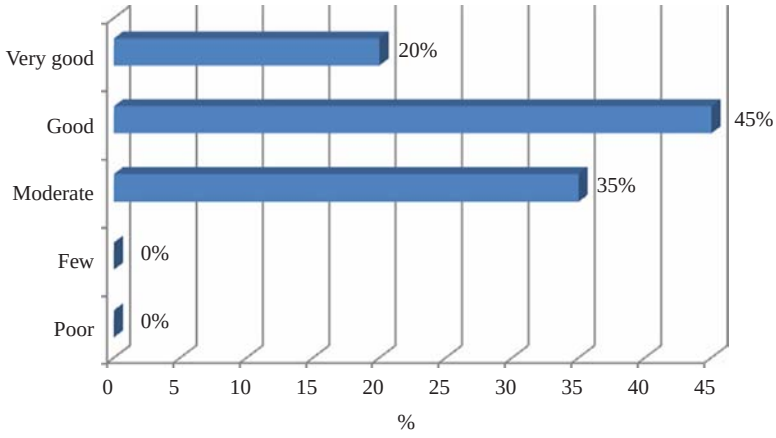

Figure 8.
Overall IRIS capabilities as an IR system

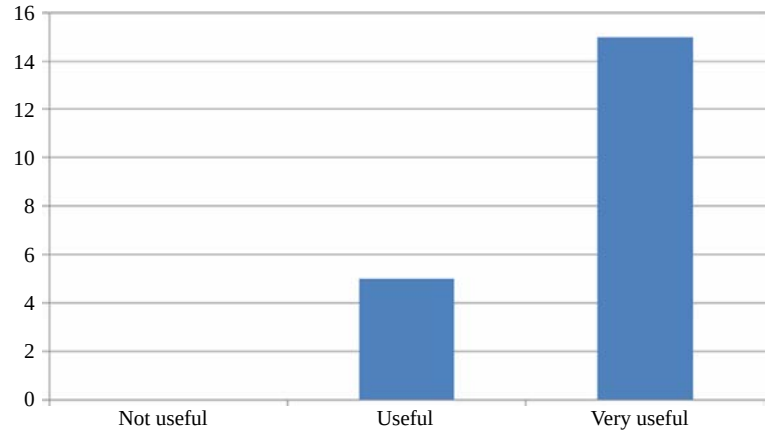

Figure 9.
IRIS capabilities of collection, normalization, correlation and scoring of vulnerability, exploit and IDP information

towards integrating our system with more VA tools and IDP systems (by the time of writing our proposed system is compatible with three VA tools and two IDP systems), while we also develop a web-based version to facilitate access from heterogeneous platforms (such as Personal Digital Assistant, netbooks, mobile phones, etc.).

Moreover, we are also considering adding a series of new functions to our system, using the common platform enumeration CPE (2009), the common weaknesses enumeration CWE (2009), the common configuration enumeration CCE (2009) and the common attack pattern enumeration and classification CAPEC (2009) which provide large and structured sets of measurable security data. The latter is expected to extend the IRIS applicability to the application security area.

Note

1. We assume that an attack path is formed only when vulnerabilities are successfully exploited in a certain order. For simplicity and without loss of generality, we may assume that a different order may not result in the same attack path, if it even results to any attack path at all. The proof of this concept, though, is beyond the scope of this paper.

IPR2025-01397
CrowdStrike EX1006 Page 15

References

Aberdeen Group (2003), *Turning IT Security into Effective Business Risk Management*, Computer Associates S.A., Bridgeton, NJ.

Adelstein, F. (2006), "Live forensics: diagnosing your system without killing it first", *Communications of the ACM*, Vol. 49 No. 2, pp. 63-6.

Ammann, P., Wijesekera, D. and Kaushik, S. (2002), "Scalable, graph-based network vulnerability analysis", *Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM Press, Washington, DC, USA*, pp. 217-24.

Brumley, D., Newsome, J., Song, D., Wang, H. and Jha, S. (2006), "Towards automatic generation of vulnerability-based signatures", *IEEE Symposium on Security and Privacy, Oakland, CA*, pp. 2-16.

Brumley, D., Newsome, J., Song, D., Wang, H. and Jha, S. (2007), *Theory and Techniques for Automatic Generation of Vulnerability-based Signatures*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

BugTraq (2009), "BugTraq", SecurityFocus, available at: www.securityfocus.com/archive/1 (accessed October 8, 2009).

CAPEC (2009), "Common attack pattern enumeration and classification, a community knowledge resource for building secure software", available at: http://capec.mitre.org/ (accessed October 25, 2009).

CCE (2009), "Unique identifiers for common system configuration issues", Common configuration enumeration, available at: http://cce.mitre.org/ (accessed October 25, 2009).

CPE (2009), "A structured naming scheme for IT systems, platforms and packages", Common platform enumeration, available at: http://cpe.mitre.org/ (accessed October 25, 2009).

CWE (2009), "A community-developed dictionary of software weakness types", Common weaknesses enumeration, available at: http://cwe.mitre.org/ (accessed October 25, 2009).

Chambers, R. (2006), "Vulnerability, coping and policy", *IDS Bulletin*, Vol. 37 No. 4, pp. 24-31.

Chien, E. and Szor, P. (2002), "Blended attacks exploits, vulnerabilities and buffer-overflow techniques in computer viruses", *Virus Bulletin Conference*, Virus Bulletin, New Orleans, LA, pp. 1-35.

Cisco (2009), "Products and services security advisories, Cisco security advisories and notices, from Cisco Systems", available at: www.cisco.com/web/go/psirt (accessed October 8, 2009).

Conklin, A.D. (2008), "Systems theory model for information security", paper presented at: 41st Annual International Conference on System Sciences, Big Island, HI.

Cui, W., Peinado, M., Wang, H. and Locasto, M. (2007), "ShieldGen: automatic data patch generation for unknown vulnerabilities with informed probing", *Proceedings of IEEE Symposium on Security and Privacy, Oakland, CA*, pp. 252-66.

Debar, H., Thomas, Y., Cuppens, F. and Cuppens-Boulahia, N. (2007), "Enabling automated threat response through the use of a dynamic security policy", *Journal of Computer Virology*, Vol. 3, pp. 192-5.

FIRST (2007), "Common vulnerability scoring system (CVSS-SIG)", available at: www.first.org/cvss/ (accessed May 7, 2010).

Gula, R. (2009), "Correlating IDS alerts with vulnerability information", Tenable Network Security, available at: www.nessus.org (accessed October 8, 2009).

Hansman, S. and Hunt, R. (2005), "A taxonomy of network and computer attacks", *Computers and Security*, Vol. 42 No. 1, pp. 31-43.

IPR2025-01397
CrowdStrike EX1006 Page 16

IBM (2009), "Proventia network enterprise scanner", IBM Corporation, available at: www.935.ibm.com/services/us/index.wss/offering/iss/a1027216 (accessed October 8, 2009).

Jajodia, S. and Noel, S. (2009), "Topological vulnerability analysis", in Jajodia, S., Wang, C., Swarup, V. and Liu, P. (Eds), paper presented at Army Research Office Cyber Situational Awareness Workshop.

Jajodia, S., Noel, S. and O'Berry, B. (2006), "Topological analysis of network attack vulnerability", in Kumar, V., Srivastava, J. and Lazarevic, A. (Eds), *Managing Cyber Threats: Issues, Approaches, and Challenges*, Springer, Heidelberg, pp. 247-66.

Kieyzun, A., Guo, P.J., Jayaraman, K. and Ernst, M.D. (2009), "Automatic creation of SQL injection and cross-site scripting attacks", *Proceedings of the 2009 IEEE 31st International Conference on Software Engineering, May 16-24, Vancouver, Canada*, pp. 199-209.

Killourhy, K., Maxion, R. and Tan, K. (2004), "A defense-centric taxonomy based on attack manifestations", *Proceedings of the International Conference on Dependable Systems & Networks, 28 June-1 July, Florence, Italy*.

Kim, D.W., Choi, Y., Kim, I.K., Oh, J.T. and Oh, J.T. (2008), Patent No. US20080083034A1, USA.

Krasser, S., Conti, G., Grizzard, J. and Gribschaw, J. (2005), "Real-time and forensic network data analysis using animated and coordinated visualization", *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY*, pp. 42-9.

McAfee (2009), "McAfee vulnerability manager (formerly foundstone)", McAfee, available at: www.mcafee.com/ (accessed October 8, 2009).

Mathew, S., Britt, D., Giomundo, R. and Upadhyaya, S. (2005), "Real-time multistage attack awareness through enhanced intrusion alert clustering", *Proceedings of Military Communications Conference, MILCOM, Atlantic City, NJ*, pp. 1801-6.

Mell, P., Scarfone, K. and Romanosky, S. (2006), "Common vulnerability scoring system", *IEEE Security and Privacy*, Vol. 4 No. 6, pp. 85-9.

Microsoft (2009), *Microsoft Security Response Center*, Microsoft Corporation, available at: www.microsoft.com/security/msrc/default.mspx (accessed October 8, 2009).

MITRE (2009), "Common vulnerabliities and exposures, the standard for information vulnerability names", available at: http://cve.mitre.org/ (accessed October 8, 2009).

Mitropoulos, S., Patsos, D. and Douligeris, C. (2006), "On incident handling and response: a state of the art approach", *Computers and Security*, Vol. 25 No. 5, pp. 351-70.

Mitropoulos, S., Patsos, D. and Douligeris, C. (2007), "Incident response requirements for distributed security information management", *Journal of Information Management & Computer Security*, Vol. 15 No. 3, pp. 226-40.

NIST (2004), "Computer security incident handling guide", NIST Special Publication 800-61, National Institute of Standards and Technology, Gaithersburg, MD.

Newsome, J. and Song, D. (2005), "Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software", *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA*.

Ning, P. and Xu, D. (2003), "Learning attack strategies from intrusion alerts", *Proceedings of the 10th ACM Conference on Computer and Communications Security* (*CCS '03*), *Washington, DC, October*, pp. 200-9.

O'Hare, S., Noel, S. and Prole, K. (2008), "A graph-theoretic visualization approach to network risk analysis", in Goodall, J.R., Conti, G. and Ma, K.-L. (Eds), VizSec 2008, LNCS, Vol. 5210, Springer, Berlin, pp. 60-7.

IPR2025-01397
CrowdStrike EX1006 Page 17

Papadaki, M. and Furnell, S.M. (2006), "Achieving automated intrusion response: a prototype implementation", *Information Management & Computer Security*, Vol. 14 No. 3, pp. 235-51.

SANS (2009), "The top cyber security risks, two risks dwarf all others, but organizations fail to mitigate them", available at: http://sans.org (accessed April 24, 2010).

Scarfone, K. and Mell, P. (2008), *Guide to Intrusion Detection and Prevention Systems (IDPS), Recommendations of the National Institute of Standards and Technology*, SP 800-94, NIST, Gaithersburg, MD.

Schultz, E. (2004), "Incident response teams need to change", *Computers and Security Journal*, Vol. 4, pp. 87-8.

SecurityFocus (2009), *Vulnerabilities*, Security Focus, available at: www.securityfocus.com/vulnerabilities (accessed October 8, 2009).

Sheyner, O., Haines, J., Jha, S. and Lippmann, R. (2002), "Automated generation and analysis of attack graphs", *Proceedings of IEEE Symposium on Security and Privacy, Oakland, CA*, pp. 273-84.

Srilatha, C., Ajith, A. and Johnson, P.T. (2004), "Feature deduction and ensemble design of intrusion detection systems", *Computers & Security*, Vol. 24 No. 4, pp. 295-307.

Swiler, L., Phillips, C., Ellis, D. and Chaker, M. (2001), "Computer-attack graph generation tool", *DISCEXII Proceedings, DARPA's Information Survivability Conference and Exposition*, Vol. 2, pp. 307-21.

Templeton, S. and Levitt, K. (2000), "A requires/provides model for computer attacks", *Proceedings of the New Security Paradigms Workshop, Ballycotton, Ireland*, pp. 31-8.

Tenable (2009), "The network vulnerability scanner", Tenable Network Security, available at: www.nessus.org/nessus/ (accessed October 8, 2009).

Tian, Z., Zhang, W., Ye, Z., Yu, A. and Zhang, H. (2008), "Reduction of false positives in intrusion detection via adaptive alert classifier", *Proceedings of the International Conference on Information and Automation (ICIA 2008), 20-3 June, Changsha*, pp. 1599-602.

US-CERT (2009), *Vulnerability Notes Database Field Descriptions*, United States Computer Emergency Readiness Team (accessed October 8, 2009).

Yegneswaran, V., Giffin, J., Barford, P. and Jha, S. (2005), "An architecture for generating semantics-aware signatures", *Proceedings of the 14th USENIX Security Symposium, Baltimore, MD*.

About the authors

Dimitrios Patsos received a PhD in Information Security from the Department of Informatics of the University of Piraeus, Greece. Prior to this, he has received a BSc in Informatics from the Department of Informatics of the Athens University of Economics and Business (AUEB) and an MSc with Distinction in Information Security from the Department of Mathematics of the Royal Holloway University of London, UK. He has served as a services manager, security consultant and security architect for leading Greek PKI & IT security companies operating in Greece, Southeastern Europe and the Middle East. His main research interests include IR systems, automated vulnerability management, systems survivability, cryptography, PKI and electronic crime. His published work includes original research works in international scientific journals (including Elsevier), international conference proceedings (including IEEE, PCI), and invited talks. He is a holder of a large number of professional certifications, and a permanent member of the Greek Computer Society (EPY). Dimitrios Patsos is the corresponding author and can be contacted at: dpatsos@gmail.com

Sarandis Mitropoulos has been a Visiting Lecturer and a Senior Researcher at the Department of Informatics of the University of Piraeus, Greece, since 2004, and a System Analyst at a Bank

IPR2025-01397
CrowdStrike EX1006 Page 18

supervised by the Ministry of Economics of Greece since 2002. He recently received (September 2007) the Master degree in Business Administration for Executives (EMBA) from AUEB. He completed his PhD at the Department of Electrical and Computing Engineering of the National Technical University of Athens (NTUA) in 1994. His PhD dissertation focused on Distributed System and Network Management. He received his degree in Informatics and Computing Engineering for the same department of NTUA in 1990. He has been working in technical and project management and business development in European R&TD and integrated solution projects, in the areas of system and network management, of advanced telecommunication/telematic services, and of management information systems, as well as of system and network security. He taught in NTUA from 1991 to 1994. He is a senior member of IEEE (*Communications and Computer Societies*) and member of CNOM, the Greek Computer Society and Technical Chamber of Greece.

Christos Douligeris received the Diploma in the Electrical Engineering from the NTUA in 1984 and the MS, MPhil and PhD degrees from the Columbia University in 1985, 1987 and 1990, respectively. He has held positions with the Department of Electrical and Computer Engineering at the University of Miami, where he reached the rank of associate professor and was the associate director for engineering of the Ocean Pollution Research Center. He is currently teaching at the Department of Informatics of the University of Piraeus, Greece. He has served in technical program committees of several conferences. His main technical interests lie in the areas of performance evaluation and security of high speed networks, neurocomputing in networking, resource allocation in wireless networks and information management, risk assessment and evaluation for emergency response operations. He was the guest editor of a special issue of the *IEEE Communications Magazine* on *Security for Telecommunications Networks* and he is the co-editor of a book on *Network Security* published by IEEE Press/Wiley. He is an editor of the *IEEE Communications Letters*, a technical editor of *IEEE Network* and a technical editor of *Computer Networks*.

To purchase reprints of this article please e-mail: **reprints@emeraldinsight.com**
Or visit our web site for further details: **www.emeraldinsight.com/reprints**

IPR2025-01397
CrowdStrike EX1006 Page 19

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.