

# Fortification of IT security by automatic security advisory processing

Stefan Fenz  
Secure Business Austria  
Vienna, Austria  
Email: sfenz@securityresearch.at

Andreas Ekelhart  
Secure Business Austria  
Vienna, Austria  
Email: aekelhart@securityresearch.at

Edgar Weippl  
Vienna University of Technology  
Vienna, Austria  
Email: weippl@ifs.tuwien.ac.at

**Abstract**—The past years have seen the rapid increase of security related incidents in the field of information technology. IT infrastructures in the commercial as well as in the governmental sector are becoming evermore heterogeneous which increases the complexity of handling and maintaining an adequate security level. Especially organizations which are hosting and processing highly sensitive data are obligated to establish a holistic company-wide security approach. We propose a novel security concept to reduce this complexity by automatic assessment of security advisories. A central entity collects vulnerability information from various sources, converts it into a standardized and machine-readable format and distributes it to its subscribers. The subscribers are then able to automatically map the vulnerability information to the ontological stored infrastructure data to visualize newly-discovered software vulnerabilities. The automatic analysis of vulnerabilities decreases response times and permits precise response to new threats and vulnerabilities, thus decreasing the administration complexity and increasing the IT security level.

## I. INTRODUCTION

Over the past years there has been a dramatic increase in information technology related incidents regarding newly-discovered software vulnerabilities. According to the CERT Coordination Center [1] the number of reported incidents doubled in the last two years (3780 reported incidents in the year 2004 and 8064 reported incidents in the year 2006) and especially organizations, which run business-critical applications, processing sensitive data, have to pay attention to this increase [2]. In the course of time, most IT environments have become heterogeneous due to several technology changes. While the administration of such environments is complex in nature, an increasing amount of IT related incidents makes the maintenance of a proper security level more difficult [3] [4]. Administrators are overwhelmed with an increasing amount of security advisories, thus they are not always able to filter relevant information for their organization [5] to patch their systems appropriately [6]. According to the CSIRT (Computer Security Incident Response Team) Organizational Survey [7], 93% of the CSIRT constituents receive their incident information via email and 79% also via phone. The majority of these reports are not structured for automatic processing and human beings have to interpret and process the reports manually to filter information which is relevant to their IT infrastructure.

According to the European Network and Information Security Agency (ENISA) [8], a CSIRT is a team of IT security

experts whose main business is to respond to computer security incidents. It provides the necessary services to handle them and to support their constituents to recover from breaches. To mitigate risks and minimize the number of required responses, most CSIRTs do not only provide reactive functions but also take a proactive role, providing security awareness training, security consulting, configuration maintenance, and producing technical documents and advisories [7]. In this paper, we focus on CSIRTs that provide such advisories on vulnerabilities (spanning hard- and software) and reports on exploits that take advantage of these flaws.

CSIRTs propagate security advisories via various channels such as public websites, member areas on websites, mailing lists, email, phone/fax, SMS, paper letters, or periodic reports to their subscribers (e.g., governmental institutions, companies, or internet service providers) [8]. Due to the permanently increasing amount of security advisories, CSIRT subscribers are often overwhelmed by processing warnings and are unable to react in time to take the necessary steps, such as patching vulnerable systems. One of the main reasons is that among the numerous security advisories only a fractional amount is relevant for a specific organization.

We propose a model which is able to process security advisories automatically to ensure faster reaction times for newly-discovered vulnerabilities. The model consists of the Security Ontology concept [9], [10], which is able to capture the IT infrastructure in a semi-automatic way, and the ATCSIRT (Austrian Computer Security Incident Response Team) concept that creates and distributes semantic and machine-readable messages to its subscribers. By mapping these messages to the ontologically stored IT infrastructure information it is possible to efficiently visualize the impact of newly-discovered vulnerabilities and to react semi-automatically or automatically. Our aim is to support humans in the process of interpreting security advisories as well as during the mitigation process. This vision strongly contributes to one of the purposes of a CSIRT, namely, to improve the overall efficiency of an organization's response processes [11].

## II. THE ARCHITECTURE

Figure 1 depicts the high-level architecture of the ATCSIRT concept, which is segmented into a publisher and subscriber

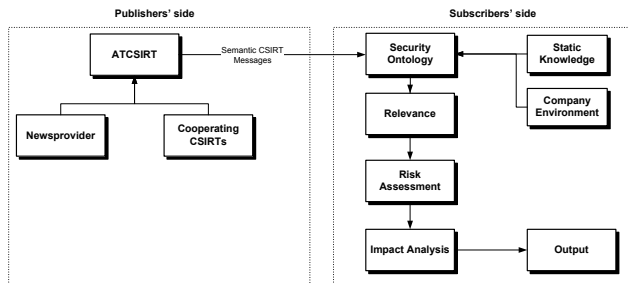


Fig. 1. The architecture

side. The publisher side represents the security advisory generating entity which provides the subscriber with semantic meaningful and machine-readable vulnerability information. This information comes from news-providers (e.g., mailing lists, websites, or vendor bulletins), cooperating CSIRTs or the publisher itself. The task of the publisher is to convert this information into a semantic meaningful and machine-readable message format to enable automatic security advisory processing at the subscriber.

Compared to text-based security advisory systems, the subscriber has two options to benefit from the semantic security advisories: First, precise automated filtering is possible because ATCSIRT security advisories are semantically structured on a highly detailed level. The subscriber is able to define rules (e.g., “Show only security advisories which are relevant to Microsoft Windows XP Systems with Service Pack 2”) to ease the information overload for the individuals who need to filter incoming security advisories.

The second option is the combination with the Security Ontology concept, which unifies static knowledge and knowledge about the concrete company environment (e.g., concrete computer servers and installed software including information about the patch level). The static knowledge is mainly segmented into threats (e.g., malware), vulnerabilities (e.g., software incidents), and countermeasures (e.g., software patches). With this ontology, an engine is able to map security advisories to the existing infrastructure of the organization and thus the impact of newly-discovered vulnerabilities can be visualized immediately. This enables the organization to react more precisely and faster to these vulnerabilities. The following sections describe the architecture and concept of the ATCSIRT in more detail.

### III. PUBLISHERS' SIDE

According to Killcrece et al. [7], the ATCSIRT is structured as an analysis center which focuses on synthesizing data from various sources to provide its subscribers with a unified and processable security advisory format. Figure 2 shows the information flow on the publishers' side which represents a modification of the information process flow according to ENISA [8].

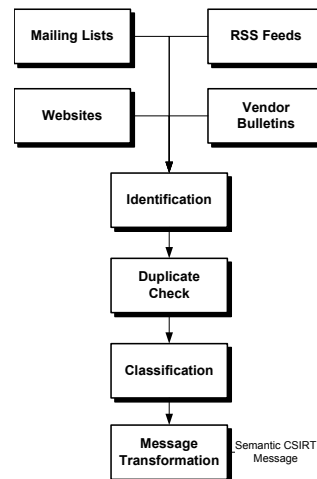


Fig. 2. Publishers' side - information flow

#### A. Information sources

Data sources such as mailing lists, RSS feeds, websites, and specific vendor bulletins are used by the ATCSIRT to provide its subscribers with the most comprehensive set of security advisories. In the most cases these sources provide the reader with information about the vulnerability regarding the following aspects: (1) general vulnerability description, (2) affected system description, (3) severity rating, and (4) solution description.

#### B. Informations' trustworthiness

Due to the wide range of data sources the identification of trustworthy security advisories is crucial for providing the subscriber with valid information. The ATCSIRT Identification module rates the information mainly automated based on (1) the general trustworthiness of the organization from which the data is drawn, (2) the correctness of the PGP-signature in the case of e-mails, and (3) the correctness of browser certificates in the case of HTTPS sources. If there are any doubts in the automatic identification phase, the information is verified by manual means such as cross-checking the information with other security advisory sources.

#### C. Redundant information

After ensuring the trustworthiness of the information the Duplicate Check module has to perform a lookup in the security advisory database to avoid that the same advisory information is processed for more than one time. This check is mainly done automatically by checking unique attributes such as CVE-reference numbers [12]. Although attributes such as advisory titles, the description of the affected systems and the URL of the recommended software patch are not unique, the ATCSIRT concept is using them to identify a potential duplicate.

#### D. Information classification and severity rating

Since not all data sources are classified as public the Classification module determines non-public information (e.g., incoming incident information from a subscriber) to ensure that the sending engine is able to distinguish between different classification levels. Since the ATCSIRT is designed as a CSIRT, which provides more than one subscriber, it is crucial that incident information, which comes from a certain organization, is not distributed to the other subscribers. Especially published security advisories regarding legacy systems can seriously affect the security of these systems.

Besides the information classification the module also provides a mapping to a standardized severity rating such as the EISPP (European Information Security Promotion Programme) vulnerability rating [13]. Because of the wide range of data sources we defined mappings from vendor-specific severity rating systems, such as the Microsoft Security Response Center Security Bulletin Severity Rating System [14], to the EISPP severity rating system.

Although standards such as the common vulnerability scoring system [15] emerge, they are not widely accepted yet. As each provider of security advisories uses its own severity rating system the ATCSIRT concept has to take different types of these systems into account to provide the subscribers with consistent vulnerability ratings.

#### E. Message transformation

At this point the vulnerability information, coming from various data sources, is converted into a standardized and machine-readable message format to enable automatic processing at the subscriber. Because mailing list and website content is mainly structured in a non-semantic form (e.g., HTML code), the ATCSIRT implements, in its Message Transformation module, conversion rules for each data source. These rules enable the transformation engine to convert the information into the ATCSIRT EISPP profile message format, which was developed as an extension to the EISPP advisory message standard [13]. A full and detailed description of the ATCSIRT EISPP profile message format goes beyond the scope of this paper and so only the most relevant parts are explained to the reader (for a full description of the ATCSIRT EISPP profile message format see Ekelhart et al. [16]).

Listing 1 shows how affected systems are described in the ATCSIRT EISPP profile: the document-unique system ID can be used as reference in the solutions section and the system information is split into *vendor*, *product*, *version*, and *patchlevel* which results in both machine-readability and easier interpretation for humans.

```
<system id="system1">
  <system_part type="platform">
    <instance tag="w2k">
      <attribute_value tag="UNSPSC">
        <value>43233004</value>
      </attribute_value>
      <attribute_value tag="vendor">
        <value>MS</value>
      </attribute_value>
    </instance>
  </system_part>
</system>
```

```
</attribute_value>
<attribute_value tag="product">
  <value>Windows</value>
</attribute_value>
<attribute_value tag="version">
  <value>2000</value>
</attribute_value>
<attribute_value tag="patchlevel">
  <value>4</value>
</attribute_value>
</instance>
</system_part>
</system>
```

Listing 1. ATCSIRT EISPP message - system information

A solution section for the system depicted above is shown in Listing 2. The *systemId* reference, the *rebootRequired* attribute and the download link with a defined file size (kilobytes) are extensions to the EISPP standard and represent essential information in automatic message processing.

```
<sol_section type="code_fix" systemId="system1"
  rebootRequired="yes">
  ...
  <reference ref_type="code_fix" issuer="MS">
    <uri size="93">
      http://www.microsoft.com/downloads/details.aspx
      ?familyid=b3599afb-7673-4ef6-a2b1-d77e39fd782c
    </uri>
  </reference>
</sol_section>
```

Listing 2. ATCSIRT EISPP message - solution section

With such a standardized and machine-readable message format it is possible to process the messages more efficient on the subscribers' side.

As mentioned above, not all security advisories are classified as *public* information and so the confidentiality as well as the integrity of the information have to be protected when it is in transmission. Therefore, the ATCSIRT holds the public keys of its subscribers to encrypt the information separately for each subscriber. Since the integrity of the information is crucial for the subscriber, the ATCSIRT digitally signs the information. Then the subscriber is able to check the digital signature of the message with the ATCSIRT's public key and to decrypt the information with its private key.

The security advisories are transmitted via the Internet (e.g., RSS feed) or via alternative channels, such as governmental or corporate Intranets, to the subscribers.

As long as publishers of vulnerability information are not using standardized message formats, it is necessary to convert the information into a standardized and semantic meaningful security advisory format such as the ATCSIRT EISPP profile. The most efficient way of distributing and processing security advisories would be the usage of a standardized and semantic meaningful security advisory format by all relevant entities such as software vendors, CSIRTs, and CSIRT subscribers. In this case, the ATCSIRT still has to perform the steps of collecting relevant information and checking its trustworthiness to provide its subscribers a single point of service regarding

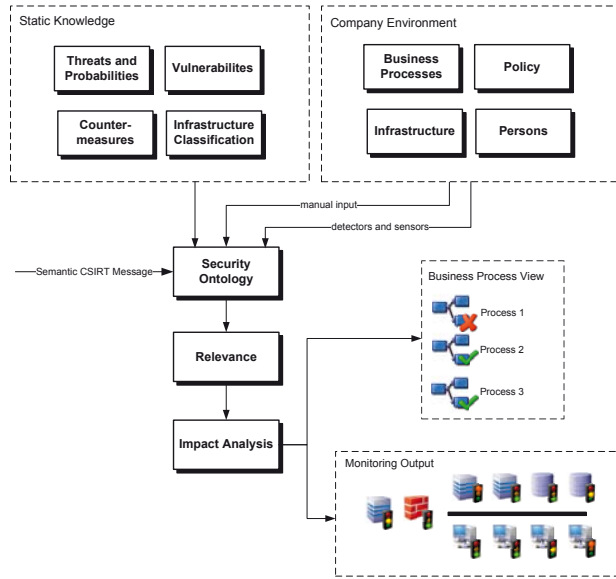


Fig. 3. Subscribers' side

security advisories.

#### IV. SUBSCRIBERS' SIDE

Figure 3 depicts the information flow on the subscriber's side. Based on the reference processes provided in ENISA CSIRT Guide [8], we have adapted the information process flow and framework to reflect our concept of automatic security advisory filtering and processing.

##### A. Trustworthiness of the information

As a first step, the subscriber has to determine the trustworthiness of the source to avoid disturbance and security breaches caused by bogus messages. For this purpose the subscriber checks the included publisher signature (from the ATCSIRT) and subsequently (if the message is encrypted) uses his private key to decrypt the received message. If the signature is not valid the message is discarded, otherwise the next process step is initialized.

##### B. Relevance and impact analysis

Prior to evaluating the relevance, the organization's installation base must be assessed. The classical approach, described in [8], [17], advises any CSIRT to gather an overview of the subscriber's IT infrastructure as a first step, using, for instances, spreadsheet templates or web forms. On the basis of these inventory lists CSIRTs can filter information before redistribution and thereby provide their subscribers only with information that is relevant for their infrastructure. We identified the following shortcomings in this approach:

- 1) The IT infrastructure gathered from the subscriber provides a static snapshot in time and is very likely to be outdated soon (e.g., a new computer could be present in the network, or new software was installed). As a

consequence essential information might not reach a subscriber in the worst case.

- 2) Keeping this system up-to-date requires an elaborate update process and time for manual processing.
- 3) Regarding security, releasing detailed information on installed systems poses as a threat, if it is released to the wrong people.
- 4) While messages are filtered for subscribers by the CSIRT, the IT administration on the subscriber side still has to evaluate the relevance of each arriving message.

The ATCSIRT approach tries to eliminate the aforementioned flaws. In comparison to the classical framework it does not depend on the storage of IT infrastructure information at the CSIRT side. As the ATCSIRT publishes semantically enriched security advisories, the automation of the filtering process can be easily achieved. While in the classical method, analyzing a message and understanding its relevance is a cumbersome process, first conducted at the CSIRT side and then at the subscriber side, semantic messages offer this information explicitly. For instance, the proposed ATCSIRT EISPP extension in Ekelhart et al. [16] has sections for affected systems, i.e. combinations of platform and software product descriptions. To unambiguously identify products, product lists such as the CSMI [18] or the product classification from the Security Ontology [16] can be used. If security advisories are represented in the ATCSIRT EISPP format, either originally created or parsed (see Section III-E), filtering messages comes to a simple field comparison. In the following three possible scenarios, including semantic security advisories regarding the relevance evaluation step, are described:

- *Manual filtering with advanced presentation:* ATCSIRT EISPP messages are well structured, consistent and contain semantic information and thus can be presented more efficiently (e.g., showing only the effected systems in a list or highlighting the severity ratings). These advanced representation possibilities make it easier for the responsible employee to select incoming information and to avoid errors during the filtering. XPath queries or complete software solutions could be offered for these purposes.
- *Semi-automatic filtering with predefined lists:* EISPP messages are machine readable due to their XML format and the meaning of the message content is given by well defined semantics in the specification. An organization can provide its IT infrastructure in compliance with these specifications and thus a reasoning engine can identify affected systems by simple matchmaking. This method takes the laborious and time-consuming work of affected system comparison from the IT administration and thus countering item 4 of the listed flaws.
- *Automatic filtering:* The automatic filtering approach is what we want to propagate in this paper, which is similar to the semi-automatic scenario but without the need of any human interaction on the subscriber's side concerning the filtering process. Inventory the IT infrastructure

can be accomplished through various software tools and guarantees up-to-date and correct information on the IT infrastructure and thereby also mitigating item 1 and 2 of the listed flaws.

As we have seen, the filtering process with semantic security advisories as source can be automated and therefore, the pre-filtering process on the CSIRT side can be omitted. Instead we emphasize on moving the whole filtering process to the subscriber side, allowing not only for up-to-date IT infrastructure information but also increasing the level of security by keeping critical infrastructure information in-house (see item 3 of the listed flaws).

The Security Ontology (see Figure 3) offers concepts to model a company in detail, comprising its infrastructure, persons, workflow processes and policies. Modeling the technical infrastructure is required for the ATCSIRT automation approach and is supported by network discovery tools (see Section V-B). Furthermore, information on threats, vulnerabilities and countermeasures is integrated into this framework. Incoming IT vulnerabilities, taken from ATCSIRT EISPP messages, can be stored as instances in the ontology. Note that not only the vulnerability with its affected systems and description is stored, but also solutions (e.g., links to update patches) (for more information see [16]). At this point, all the necessary information for automatic relevance determination is set up.

In the relevance phase the security advisories attributes are compared to the stored infrastructure instances. E.g., a vulnerability message MS07-022 lists Windows 2000 Service Pack 4 operating systems as vulnerable. These specifications are compared to the organization's IT infrastructure information stored in the ontology, resulting in a list of potentially vulnerable systems.

### C. Impact report and visualization

In the next step, the vulnerable systems are presented to the organization's IT administration. As Figure 3 illustrates, one effective way for presentation are network plans including signs to visualize problems. We have chosen traffic lights for our initial prototype as the colors red, yellow and green are intuitively understood. Systems flashing red draw attention and signalize danger, yellow denotes an uncertain state with potential problems and has to be investigated and green indicates a healthy system. At one glance the overall organization's IT network can be monitored, which is also an attractive option for management purposes. Passing over the system symbols in the graphic calls up information on the reason for the set state (e.g., the vulnerability description in case of a red state). Of course detailed reports comprising data of the relevance check and information on the effected systems can be generated from the knowledge base. Rule based notification to administration personnel is a further option.

Besides notification and visualization of identified vulnerabilities in the organization's system, the idea of the Security

Ontology concept goes beyond. As shown in [9], the in-depth knowledge on a company's infrastructure and processes allows for automated risk analysis. In the same way, the impact of vulnerable systems on the organization can be calculated. E.g., affects on business processes can be visualized.

## V. PROOF OF CONCEPT

In this section we provide a simplified example of how an organization would use the aforementioned ATCSIRT to secure its IT environment in an efficient way.

### A. Subscribing at the ATCSIRT

The subscriber registers its institution by entering contact information, its institution profile and its digital certificate by the registration form of the ATCSIRT portal. After a successful validity check of the certificate (the given contact information is compared to the contact information of the digital certificate) the subscriber receives the credentials for its personal area on the ATCSIRT portal by an out-of-band communication such as mail or fax. The aforementioned security advisories which are not classified as public are distributed encrypted with the subscriber's public key over a personalized RSS feed to the corresponding subscriber.

After a successful registration the software necessary for the ATCSIRT concept is set up, including the following components: (1) Inventory module (implemented in Python), which inventors the IT infrastructure of the subscriber, (2) Import module (implemented in Java), which listens for feeds coming from the ATCSIRT and for XML data coming from the Inventory module, provides functions for encryption and integrity checks, and imports the security advisory and infrastructure information into the Security Ontology, (3) Risk Assessment module (implemented in Java), which compares security advisory content to the stored inventory data and thus identifies threatened systems, and (4) Impact Analysis module (implemented in Java), which visualizes the results and generates textual status reports.

### B. Inventory at the subscribers' side

After the subscription step the subscriber inventors its IT infrastructure by the ATCSIRT Inventory module (see Figure 4). The software installation is guided by the ATCSIRT team. The Inventory module is implemented in Python and incorporates over a plug-in mechanism several third-party products such as Nmap [19], OCS [20] and WSUS [21] to ensure a platform-independent collection of IT infrastructure data.

The Inventory module scans at frequent intervals (e.g., each hour) the accessible network segment by utilizing the configured third-party products. The following scanning procedure was used for the proof of concept: (1) *nmap-list* to check the reverse DNS entries, (2) *nmap-ping* to check which hosts are replying to an ICMP echo, (3) *nmap-scan* to scan a defined IP address range, (4) *nmap-os* to determine the operating system of active hosts in a defined IP address range, (5) tap the OCS inventory database if available, (6) tap the WSUS inventory database if available, and (7) run SNMP scan to gather data

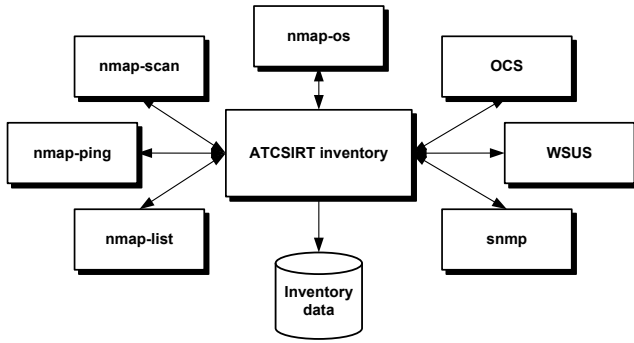


Fig. 4. ATCSIRT inventory module

from devices which use the SNMP protocol. Further plug-ins will include support for the Solaris Patch Manager, the Red Hat Network, and the Debian Packaging System.

The scan conducted by the inventory module in the test environment results in the following infrastructure data (for enhanced readability we only focus on a part of the network infrastructure): ten workstations (*WS0* - *WS9*) and a firewall *FW1*. For each infrastructure element the inventory module also determines its current software configuration (e.g., Microsoft Windows 2000 SP4 is installed on *WS8*).

The results of each scan are stored in a XML file located on a network share. The file watcher of the Import module recognizes the changes and imports the content into the Security Ontology as depicted in Figure 3.

### C. Message handling at the publishers' side

1) *Collecting Vulnerability Information:* In the current stage of development the ATCSIRT prototype collects vulnerability information based on the information published by RSS feeds such as the Microsoft Bulletin RSS feed from the Microsoft TechNet<sup>1</sup>. This RSS feed announces a list of short descriptions of security vulnerabilities concerning Microsoft software products. Each of the entries contains a URL to a specific website that describes the entire vulnerability information more precisely on a specific website. To get this additional information the prototype downloads the content of each of these websites and converts it to the ATCSIRT EISPP profile format as described in the next subsection. More sophisticated versions of the prototype will include the support of mailing lists and websites with no RSS functionality as source. While mailing lists deliver the information by push mechanisms to the ATCSIRT, websites without any RSS functionality require a watcher (e.g., *Website watcher* [22] or *Watch that page* [23]) which indicates changes on a website to enable the collection of new vulnerability information by the ATCSIRT.

For the proof of concept we used a typical Microsoft Security Bulletin, which contains vulnerability information such as title, creation date, impact type, and severity rating. Since this information is represented in HTML code with no

<sup>1</sup>Microsoft Bulletin RSS feed: <http://www.microsoft.com/technet/security/bulletin/secrss.aspx>, last access: 8 June 2007

semantic information included, the ATCSIRT has to transform this information into a semantic and machine-readable format such as the ATCSIRT EISPP profile as shown in the following step.

2) *Evaluation of the information and assessment of the risk:* Since the ATCSIRT consumes vulnerability data from a wide range of sources it has to determine the trustworthiness of every source. In the current stage of development we implemented a white list of allowed domains (e.g., mitre.org, microsoft.com, and adobe.com) which allows the ATCSIRT to distinguish between trusted and non-trusted sources. In more sophisticated versions, the ATCSIRT will additionally check the correctness of the PGP-signature in the case of e-mails, and the correctness of browser certificates in the case of HTTPS sources to ensure the integrity of the received security advisory.

After confirming the trustworthiness of the received information the prototype parses it to the ATCSIRT EISPP profile format to ensure that the information can be processed automatically in the subsequent steps.

Since the vulnerability information coming from Microsoft does not use a severity rating which is compatible to the EISPP standard we transform the rating of MS07-022 (Microsoft rating *important*) to the EISPP rating *high* as described in Section III-D. Because Section III-E already presents parts of the generated ATCSIRT EISPP profile message the following listing shows only the identification and vulnerability data of the transformed Microsoft Security Bulletin MS07-022:

```
<EISPP-Advisory issuer="MS" version="1.0" xml:lang="en"
date="April 10, 2007">
<Id_Data>
<ref_num>MS07-022</ref_num>
<title>
<FreeText>Micros... </FreeText>
</title>
<abstract>
<FreeText>This upda... </FreeText>
</abstract>
</Id_Data>
<Vulnerability_Class>
<vulnerabilities>
<vulnerability>
<vuln_ids>
<vuln_id issuer="CVE" ref_num="CVE-2007-1206"/>
</vuln_ids>
<confidence_level type="official_and_tested"/>
<immediacy rating="high"/>
</vulnerability>
</vulnerabilities>
</Vulnerability_Class>
...
```

Listing 3. ATCSIRT EISPP message - identification data

Using this message format enables actors to process security advisories on a very granular level as described in Section V-D.

3) *Distribution:* After the message transformation the security advisory is transmitted to the relevant subscribers, where a public and a private RSS feed are used for the distribution. All security advisories which are classified as

public are transmitted over a RSS feed which is accessible by all subscribers. Each entry in the RSS feed provides a link to a XML file located on the ATCSIRT portal which contains the security advisory in the ATCSIRT EISPP profile format. In the case of non-public security advisories the distribution is done separately for the relevant subscribers over their private RSS feed. Because non-public security advisories are encrypted with the public key of the subscriber, each subscriber needs its own copy available on its personal area on the ATCSIRT portal. The URL to the encrypted security advisory is provided by the private RSS feed.

#### D. Message handling at the subscribers' side

If a new vulnerability message is available for a subscriber, as a first step the publisher's signature is checked by the Import module—in case the issuer is not the ATCSIRT the message is declined. Encrypted messages, not classified as public, are in the second step decrypted with the subscriber's private key.

1) *Relevance and impact analysis*: Now, when the trustworthiness of a new vulnerability message has been confirmed and the message is available in clear text, the message's information must be stored locally by the Import module. For this purpose we use the Security Ontology which comprises concepts for vulnerabilities and solutions. The *Software Vulnerability* concept has the necessary properties to store all aspects of the ATCSIRT EISPP message format. Connections between a vulnerability and affected systems are modeled by taking advantage of product categories from the Security Ontology.

Mitigation of this vulnerability on the specific operating system Microsoft Windows 2000 can be achieved by the software patch *Windows2000-KB931784-x86*. Further information on the software patch, including file size and the download URL, could be investigated by inspecting this instance.

As explained in Section V-B, the organization's IT infrastructure has been inventoried (by the Inventory module) and completely stored in the ontological knowledge base (by the Import module). After a new message arrives, a vulnerability check is immediately conducted automatically by searching for affected systems in the organization's infrastructure by the Risk Assessment module. Of course, as not only new vulnerability messages appear, but also the infrastructure and configurations change, all existing vulnerabilities have to be tested against the changed infrastructure. Our prototype uses SPARQL [24], a promising RDF query language from the W3C RDF Data Access Working Group, to find affected systems. Listing 4 shows a SPARQL query to identify vulnerable systems which are already patched (*ent:computerHasSoftwarePatch ?vMitigation*). Resulting systems from this query would be marked as unaffected (denoting as green).

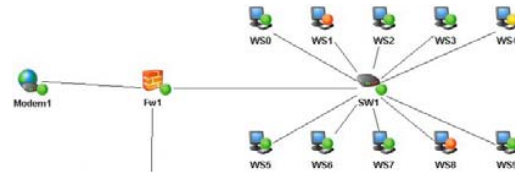


Fig. 5. ATCSIRT Management Cockpit excerpt

```

SELECT ?computer
WHERE {
  ?computer a ?x .
  ?x rdfs:subClassOf ent:Computer .

  ?vul a atcert:SoftwareVulnerability .
  ?vul atcert:softwareVulnerabilityAffectsOS ?vOs .
  ?vul atcert:softwareVulnerabilityAffectsOSPatchLevel
    ?vOSPatchLevel .
  ?vul atcert:softwareVulnerabilityMitigatedBySoftwarePatch
    ?vMitigation .

  ?computer ent:computerHasOS ?vOs .
  ?computer ent:computerHasSoftwarePatch ?vOSPatchLevel .
  ?computer ent:computerHasSoftwarePatch ?vMitigation .
}

```

Listing 4. SPARQL query to determine patched systems

To identify affected systems—computer systems which match the vulnerability description but do not have the patch installed—the difference of affected systems and affected systems with the patch installed has to be calculated. This returns two endangered systems, Workstation *WS1* and *WS8* (cf. Section V-B). *WS4* is marked as yellow by reason that the system has been found in the inventory step, but no information on installed software could not be extracted.

2) *Impact report and visualization*: At this point a notification is transmitted to the administration personnel via email or SMS by the Impact Analysis module, including the vulnerability description and the affected systems, taken from the knowledge base. Within seconds after the receipt of a new vulnerability message, the administrator knows vulnerable systems in his organization and can react immediately to the incident. By inspecting the logging output of the Impact Analysis module or opening the Management Cockpit (see Figure 5), the administrator gains additional information, as well as solution information. The Management Cockpit view reflects the organization's network, gathered from the most recent inventory, and follows the visualization system explained in Section IV-C to draw attention on vulnerable systems. JGraph, a Java Open Source graph drawing component, is used to generate the visualization output. Only the components' identifiers are shown in the cockpit view for clarity, by hovering over a component a tooltip with further information appears (e.g., the IP address).

As a further extension notifications can be triggered depending on the vulnerability rating or the importance of the endangered system. For instance, if affected systems are found but the severity rating of a message is only *low*, the administrator should not be thrown out of bed by an automated phone call; if, however, one of the systems is highly business

critical maybe even several specific people should be alerted. A rule based decision system will be developed to follow these ideas and improve the reaction times and efficiency.

## VI. CONCLUSION AND OUTLOOK

In the current paper we proposed a novel security concept which helps to maintain an adequate IT security level by the automatic assessment of security advisories. Organizations are faced with heterogeneous IT infrastructures and an increasing number of incidents. Therefore, we proposed a twofold concept to counteract these threats: On the one hand a central entity collects vulnerability information from various vendor-independent data sources and converts it into a machine-readable and semantic meaningful data format. On the other hand subscribers consume this information to map it to their ontological stored IT infrastructure data which was gathered in an inventory phase. A detailed description of each IT infrastructure element, including its operating system and software configuration, in combination with semantic security advisories enables faster reaction times and precise response to new threats and vulnerabilities at the subscribers' side.

Further research will address the monetary assessment of the potential damage caused by newly-discovered vulnerabilities by taking the value of IT infrastructure elements into account. The ontology offers detailed information on connections between systems and thus allows for sophisticated reasoning on effects. Furthermore, the integration of a business process view should indicate via the IT infrastructure elements which business processes are endangered by newly-reported incidents. Further research activities will also address the possibility for a centralized, vendor independent, automatic patching mechanism, to ensure that vulnerable systems are patched in a timely manner after a software vulnerability is reported.

## ACKNOWLEDGEMENTS

This work was performed at Secure Business Austria, a competence center that is funded by the Austrian Federal Ministry of Economics and Labor (BMWA) and by the City of Vienna.

## REFERENCES

- [1] CERT/CC, "Cert/cc statistics 1988–2006," <http://www.cert.org/stats/>, January 2007.
- [2] R. Holbein and T. Gaugler, "It-security in electronic commerce: From cost to value driver," in *DEXA '99: Proceedings of the 10th International Workshop on Database & Expert Systems Applications*. Washington, DC, USA: IEEE Computer Society, 1999, p. 816.
- [3] R. Bhaskar, "State and local law enforcement is not ready for a cyber katrina," *Commun. ACM*, vol. 49, no. 2, pp. 81–83, 2006.
- [4] M. Franz, "Containing the ultimate trojan horse," *IEEE Security and Privacy*, vol. 5, no. 4, pp. 52–56, 2007.
- [5] D. Lekkas and D. Spinellis, "Handling and reporting security advisories: A scorecard approach," *IEEE Security and Privacy*, vol. 3, no. 4, pp. 32–41, July/August 2005. [Online]. Available: <http://www.spinellis.gr/pubs/jrnl/2005-CS-SecAdvisory/html/LS05.htm>
- [6] M. Carvalho, T. Cowin, N. Suri, M. Breedy, and K. Ford, "Using mobile agents as roaming security guards to test and improve security of hosts and networks," in *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*. New York, NY, USA: ACM Press, 2004, pp. 87–93.

- [7] G. Killcrece, K.-P. Kossakowski, R. Ruefle, and M. Zajicek, "State of the practice of computer security incident response teams (csirts)," Carnegie Mellon Software Engineering Institute, Tech. Rep. CMU/SEI-2003-TR-001, October 2003.
- [8] European Network and Information Security Agency (ENISA), "A step-by-step approach on how to set up a csirt," European Network and Information Security Agency (ENISA), Tech. Rep. Deliverable WP2006/5.1(CERT-D1/D2), 2006.
- [9] A. Ekelhart, S. Fenz, M. Klemen, and E. Weippl, "Security ontologies: Improving quantitative risk analysis," in *40th Hawaii International Conference on System Sciences (HICSS'07)*. Los Alamitos, CA, USA: IEEE Computer Society, Jan 2007, pp. 156–162.
- [10] A. Ekelhart, S. Fenz, Markus D. Klemen, A. Tjoa, and E. Weippl, "Ontology-based business knowledge for simulating threats to corporate assets," in *Practical Aspects of Knowledge Management (PAKM'06)*, ser. Lecture Notes in Artificial Intelligence, U. Reimer and D. Karagiannis, Eds., vol. 4333. Vienna, Austria: Springer, Dec 2006, pp. 37–48.
- [11] V. Masurkar, "Responding to a customer's security incidents - part 1: Establishing teams and a policy," Sun Microsystems, Inc., Santa Clara, CA 95045 U.S.A., Tech. Rep. 817-1795-10, March 2003.
- [12] MITRE, "Common vulnerabilities and exposures - the standard for information security vulnerability names," <http://cve.mitre.org/>, May 2007.
- [13] EISPP Consortium, "Eispp common advisory format description," [http://www.eispp.org/commonformat\\_2\\_0.pdf](http://www.eispp.org/commonformat_2_0.pdf), May 2004.
- [14] Microsoft Corporation, "Microsoft security response center security bulletin severity rating," <http://www.microsoft.com/technet/security/bulletin/rating.mspx>, November 2002.
- [15] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security & Privacy*, vol. 4, no. 6, pp. 85–89, Nov 2006.
- [16] A. Ekelhart, S. Fenz, and D. Karollus, "Semantic enrichment for the efficient handling of incident messages," Secure Business Austria, Tech. Rep., May 2007.
- [17] Moira J. West-Brown, Don Stikvoort, Klaus-Peter Kossakowski, Georgia Killcrece, Robin Ruefle, and Mark Zajicek, "Handbook for computer security incident response teams (csirts)," Carnegie Mellon, Software Engineering Institute, Pittsburgh, PA 15213-3890, Tech. Rep. CMU/SEI-2003-HB-002, April 2003.
- [18] B. Grobauer, "Cve, cme, ... cmsi? standardizing system information," in *17th Annual FIRST Conference on Computer Security Incident Handling*. Singapore, Singapore: Forum of Incident Response and Security Teams, June 2005.
- [19] I. LLC, "Nmap security scanner," <http://insecure.org/nmap/>, June 2007.
- [20] OCS Inventory Development Team, "Ocs next generation inventory," <http://www.ocsinventory-ng.org/>, June 2007.
- [21] Microsoft Corporation, "Windows server update services," <http://www.microsoft.com/germany/windowsserver2003/technologien/updateservices/default.mspx>, June 2007.
- [22] M. Aignesberger, "Website watcher," <http://www.aignes.com/index.htm>, June 2007.
- [23] A. Consulting, "Watch that page," <http://www.watchthatpage.com/>, June 2007.
- [24] "Sparql query language for rdf," <http://www.w3.org/TR/rdf-sparql-query/>, 2006.