Software Vulnerability Management Techniques

Ehab Qadah

Seminar: Selected Topics in IT-Security

University of Bonn

Motivation

- In 2015, according to Symantec, 5,585 new vulnerabilities were found.

- According the US Department of Homeland Security (DHS), 90% of security incidents result from exploits against defects in software.

- Thus, finding the vulnerable software inside an organization is important to prevent cyber-attacks.

Outline

- Introduction

- Software Vulnerability Management (SVM) Standards
    - Security Content Automation Protocol (SCAP)
    - Common Platform Enumeration (CPE)
    - Common Vulnerabilities and Exposures (CVE)
    - Common Vulnerability Scoring System (CVSS)

- Software Vulnerability Management Approaches
    - SVM Using open Standards
    - SVM Using Social Networks Information

- Discussion

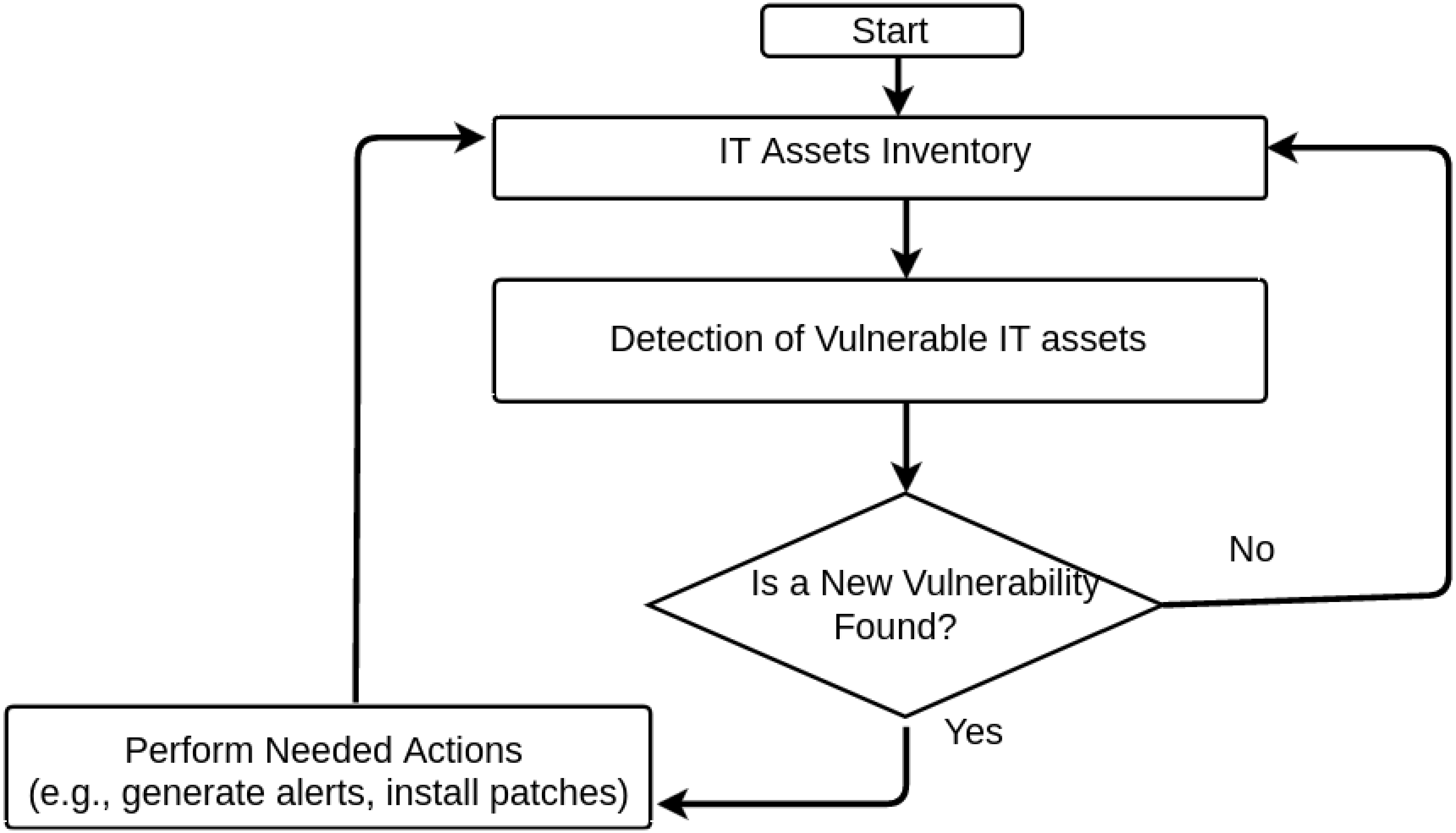
- Conclusion

Introduction

- **Vulnerability**: is a defect or weakness in a system that leads to a security incident or unauthorized access to information or services by attackers.

- **Software Vulnerability Management (SVM)**: the continuous process that identifies vulnerabilities in a software product that is installed inside an organization.

Introduction


Start
IT Assets Inventory
Detection of Vulnerable IT assets
Is a New Vulnerability Found?
No
Yes
Perform Needed Actions
(e.g., generate alerts, install patches)

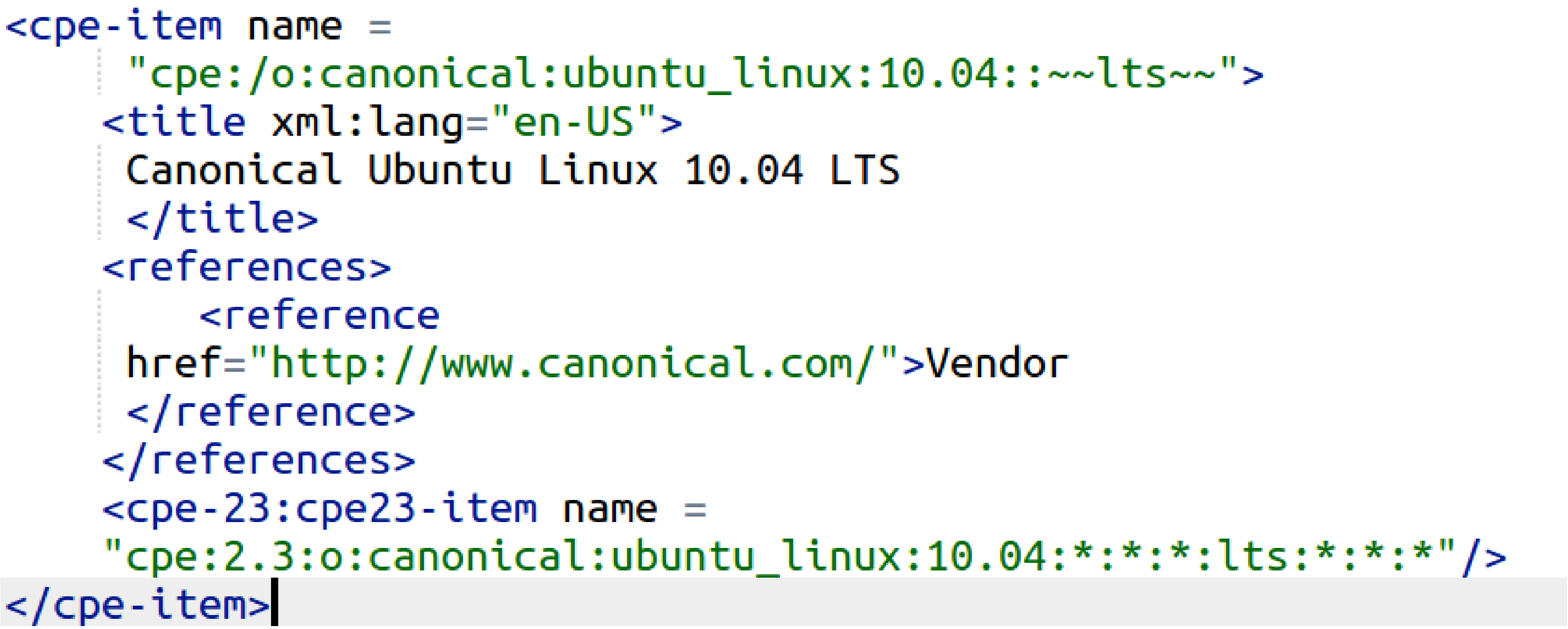
Process flow of the Software Vulnerability Management System.

SVM Standards

- Security Content Automation Protocol (**SCAP**)
  - Collection of open standards to identify security flaws and config issues.
  - Supports automated vulnerability management, patch checking, and security measurement.
  - Maintained by the National Institute of Standards and Technology (NIST).
  - Content is accessible via the National Vulnerability Database (NVD).
  - Used by SVM systems to automate the process of identification of vulnerable software.

Common Platform Enumeration (CPE)

- A dictionary that identifies software products and applications, operating systems, and HW devices.



```
<cpe-item name =
     "cpe:/o:canonical:ubuntu_linux:10.04::~~lts~~">
    <title xml:lang="en-US">
     Canonical Ubuntu Linux 10.04 LTS
     </title>
    <references>
        <reference
     href="http://www.canonical.com/">Vendor
     </reference>
    </references>
    <cpe-23:cpe23-item name =
    "cpe:2.3:o:canonical:ubuntu_linux:10.04:*:*:*:lts:*:*:*"/>
</cpe-item>
```

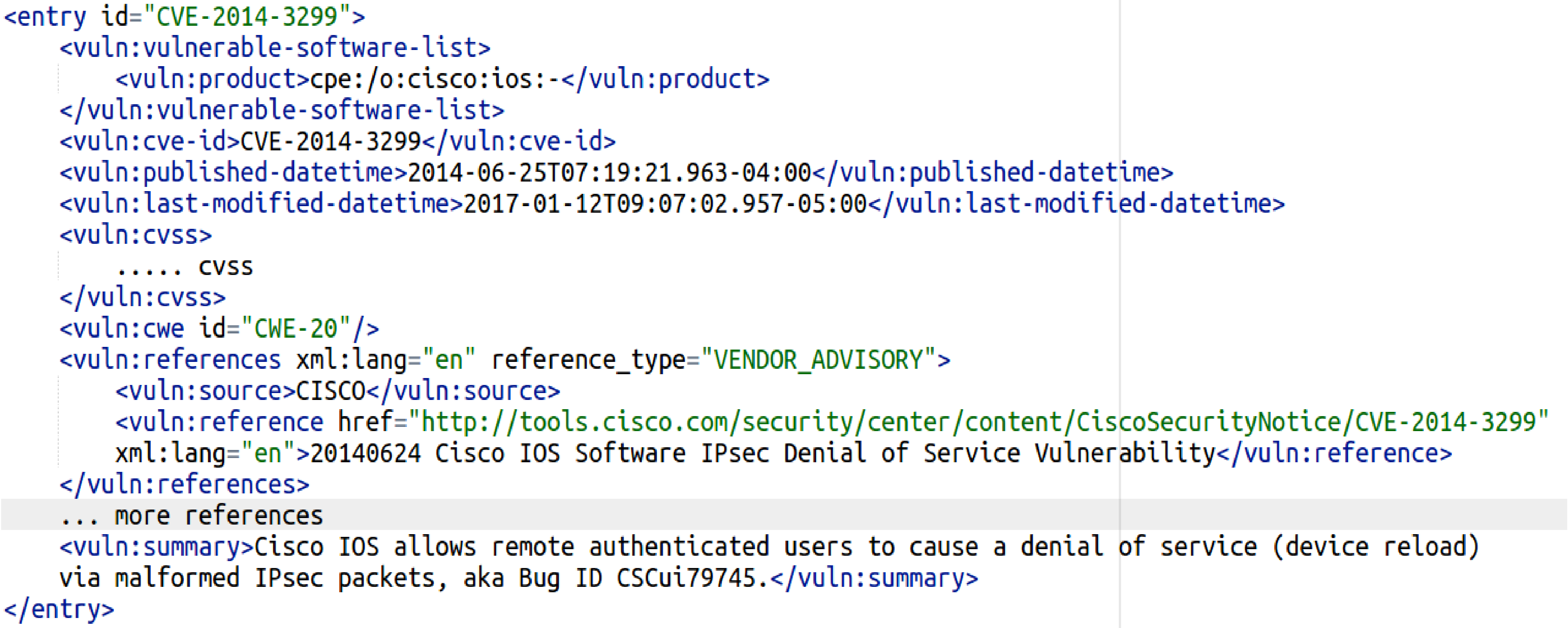
An example of the Official CPE Dictionary entry provided By NVD.

Common Vulnerabilities and Exposures (CVE)

- List of publicly known vulnerabilities.
- Unique ID for each vulnerability is assigned.
- e.g.: **CVE-2017-0001**, where the CVE ID is CVE prefix + year + sequence number.
- Allow to share data between separate security tools.

Common Vulnerabilities and Exposures (CVE)

- NVD provides an XML feed for the vulnerabilities listed in CVE.

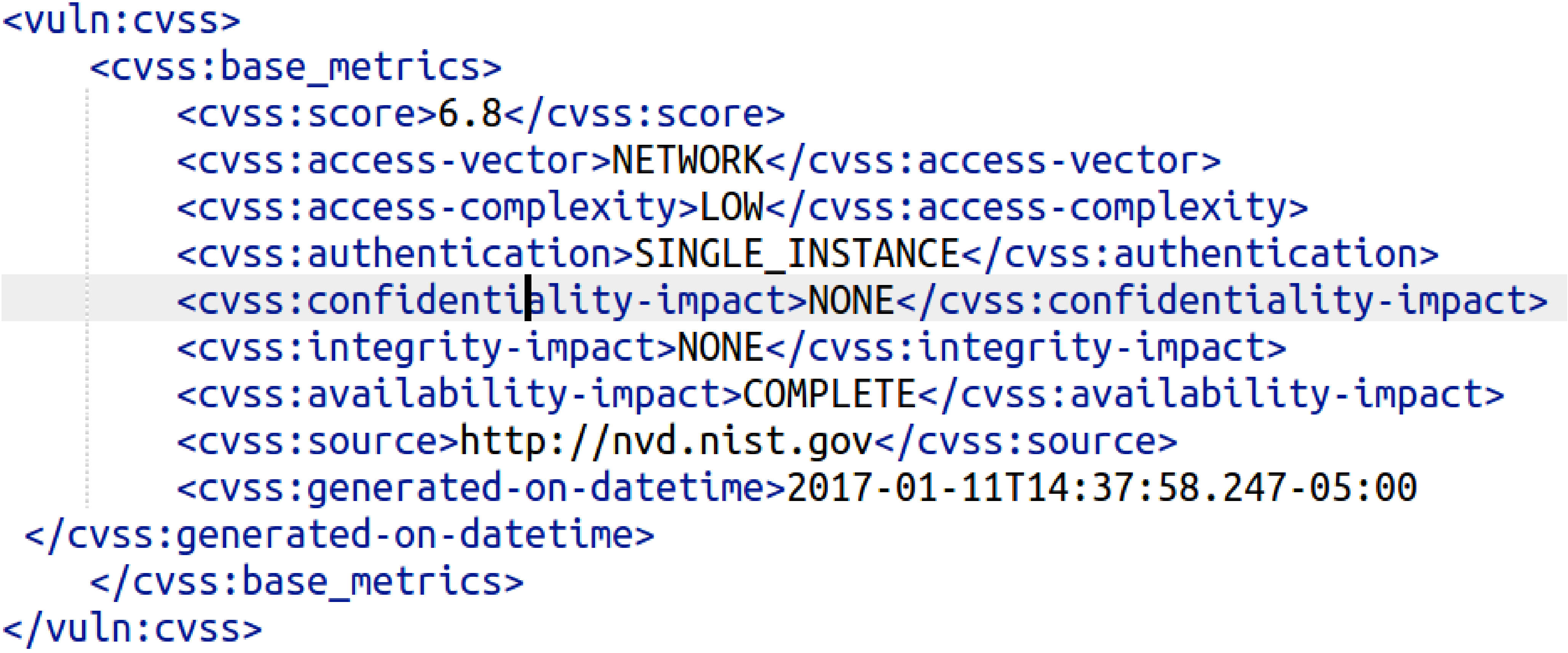


```
<entry id="CVE-2014-3299">
    <vuln:vulnerable-software-list>
        <vuln:product>cpe:/o:cisco:ios:-</vuln:product>
    </vuln:vulnerable-software-list>
    <vuln:cve-id>CVE-2014-3299</vuln:cve-id>
    <vuln:published-datetime>2014-06-25T07:19:21.963-04:00</vuln:published-datetime>
    <vuln:last-modified-datetime>2017-01-12T09:07:02.957-05:00</vuln:last-modified-datetime>
    <vuln:cvss>
        ..... cvss
    </vuln:cvss>
    <vuln:cwe id="CWE-20"/>
    <vuln:references xml:lang="en" reference_type="VENDOR_ADVISORY">
        <vuln:source>CISCO</vuln:source>
        <vuln:reference href="http://tools.cisco.com/security/center/content/CiscoSecurityNotice/CVE-2014-3299"
        xml:lang="en">20140624 Cisco IOS Software IPsec Denial of Service Vulnerability</vuln:reference>
    </vuln:references>
    ... more references
    <vuln:summary>Cisco IOS allows remote authenticated users to cause a denial of service (device reload)
    via malformed IPsec packets, aka Bug ID CSCui79745.</vuln:summary>
</entry>
```

“CVE-2014-3299” vulnerability entry by NVD.

Common Vulnerability Scoring System (CVSS)

- Scoring system which provides the characteristics and relative severity of a vulnerability.



```
<vuln:cvss>
    <cvss:base_metrics>
        <cvss:score>6.8</cvss:score>
        <cvss:access-vector>NETWORK</cvss:access-vector>
        <cvss:access-complexity>LOW</cvss:access-complexity>
        <cvss:authentication>SINGLE_INSTANCE</cvss:authentication>
        <cvss:confidentiality-impact>NONE</cvss:confidentiality-impact>
        <cvss:integrity-impact>NONE</cvss:integrity-impact>
        <cvss:availability-impact>COMPLETE</cvss:availability-impact>
        <cvss:source>http://nvd.nist.gov</cvss:source>
        <cvss:generated-on-datetime>2017-01-11T14:37:58.247-05:00
 </cvss:generated-on-datetime>
    </cvss:base_metrics>
</vuln:cvss>
```

CVSS score and metrics of the "CVE-2014-3299" vulnerability.

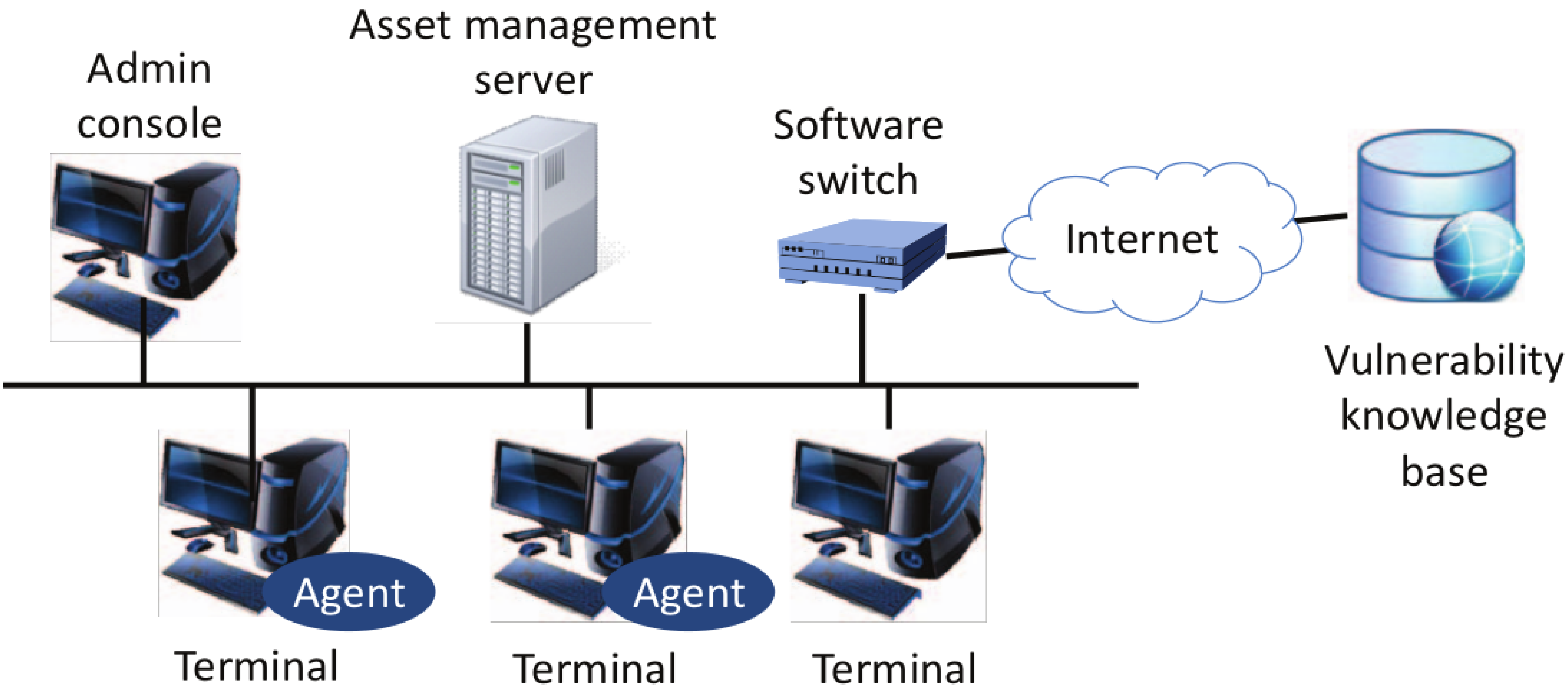
Software Vulnerability Management Approaches

SVM Using Open Standards

- Proposed by Takahashi et al.[1] to automatically monitor vulnerabilities in IT assets inside an organization.
- Open standards and information sources are used to develop a system that can be used by a wide range of organizations.

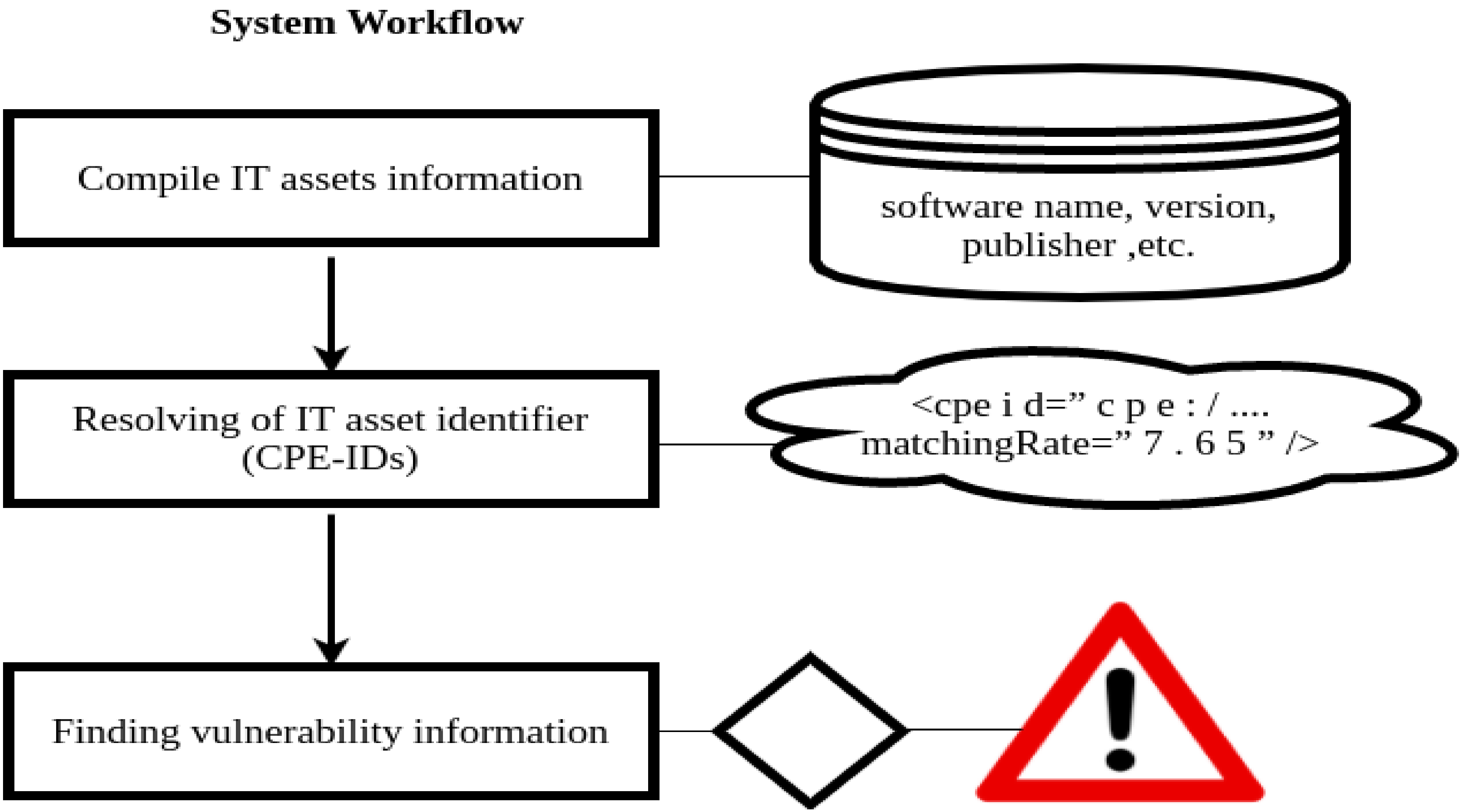
[1] Takahashi et al. ”Toward automated vulnerability monitoring using open information and standardized tools.”

SVM Using Open Standards



Source: Takahashi et al. ”Toward automated vulnerability monitoring using open information and standardized tools.”

SVM Using Open Standards


System Workflow
Compile IT assets information
software name, version, publisher ,etc.
Resolving of IT asset identifier (CPE-IDs)
<cpe i d=" c p e : / ....
matchingRate=" 7 . 6 5 " />
Finding vulnerability information

SVM using Social Networks Information

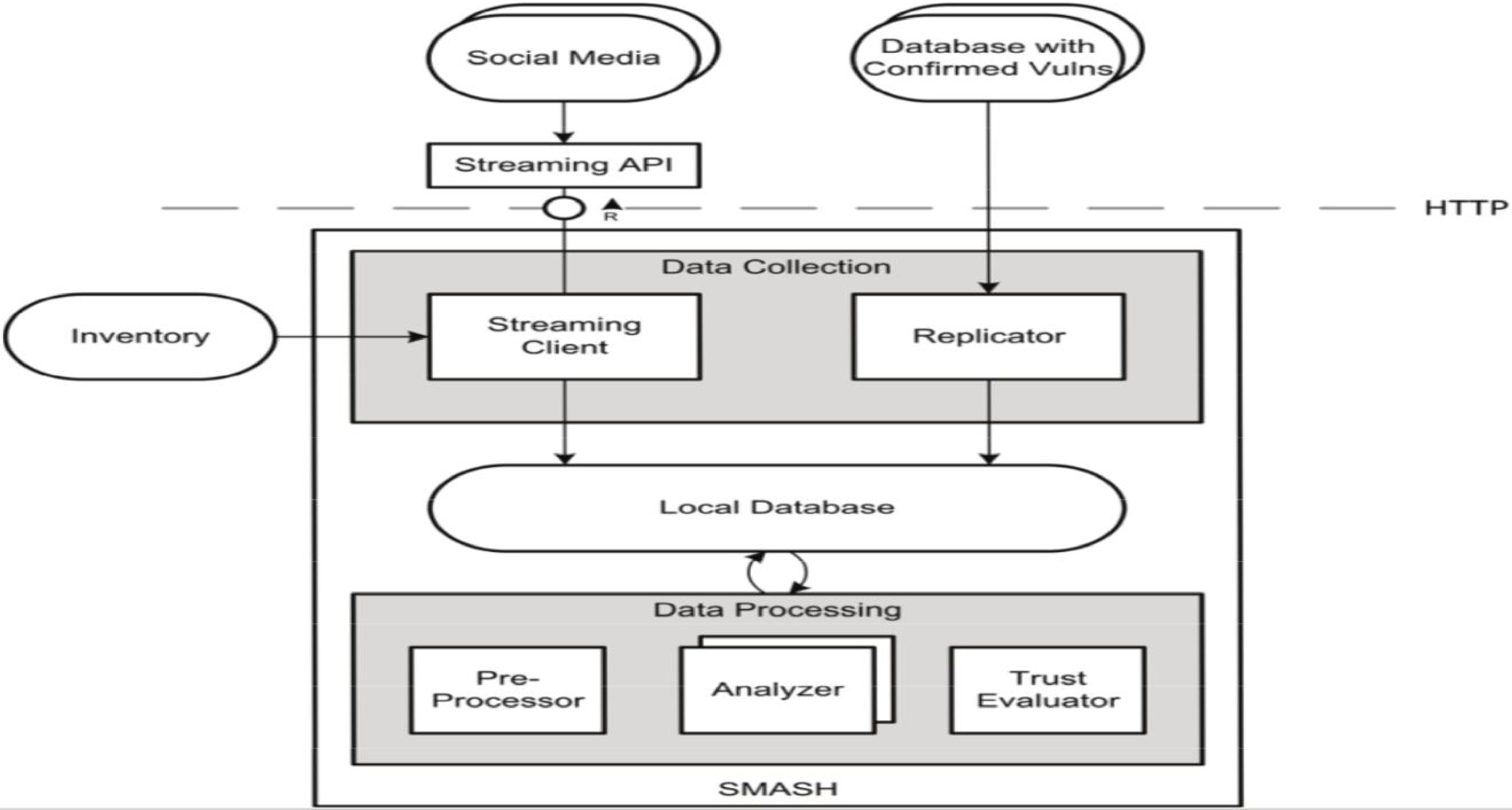
- Getting informed about zero-day vulnerabilities is difficult, due to the delay of vulnerability information publication.

- Trabelsi et al.[1] proposed an approach to detect software vulnerabilities based on security information collected from Twitter.

- Identify a vulnerable software earlier than the normal vulnerabilities repositories.

[1] Trabelsi, Slim, et al. ”Mining social networks for software vulnerabilities monitoring.”

SVM using Social Networks Information



Source: Trabelsi, Slim, et al. ”Mining social networks for software vulnerabilities monitoring.”

Discussion

- The first system is not fully automated, it just sends an alert when a new vulnerability is found.
- The first system does not validate the computed CPE-ID.
- The accuracy of the second system’s output is fundamentally affected by the strength of the user trust model.

Conclusion

- Organizations must continually monitor software vulnerabilities to prevent possible attacks.
- SVM systems help organizations to automatically monitor the software vulnerabilities.
- SCAP Standards (e.g., CVE) help to automate the process of SVM and the exchange of security information.
- Issue: the availability and reliability of vulnerabilities information.

THANK YOU FOR YOUR ATTENTION