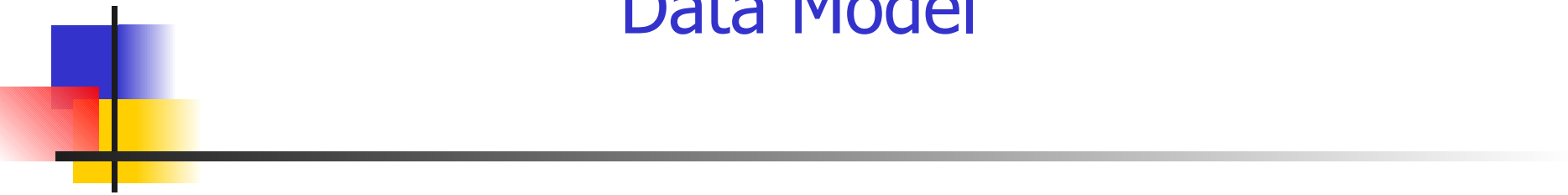
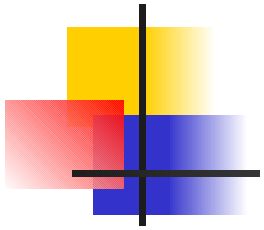
Vulnerability Assessment Report Format Data Model



Dr.D.Polemi

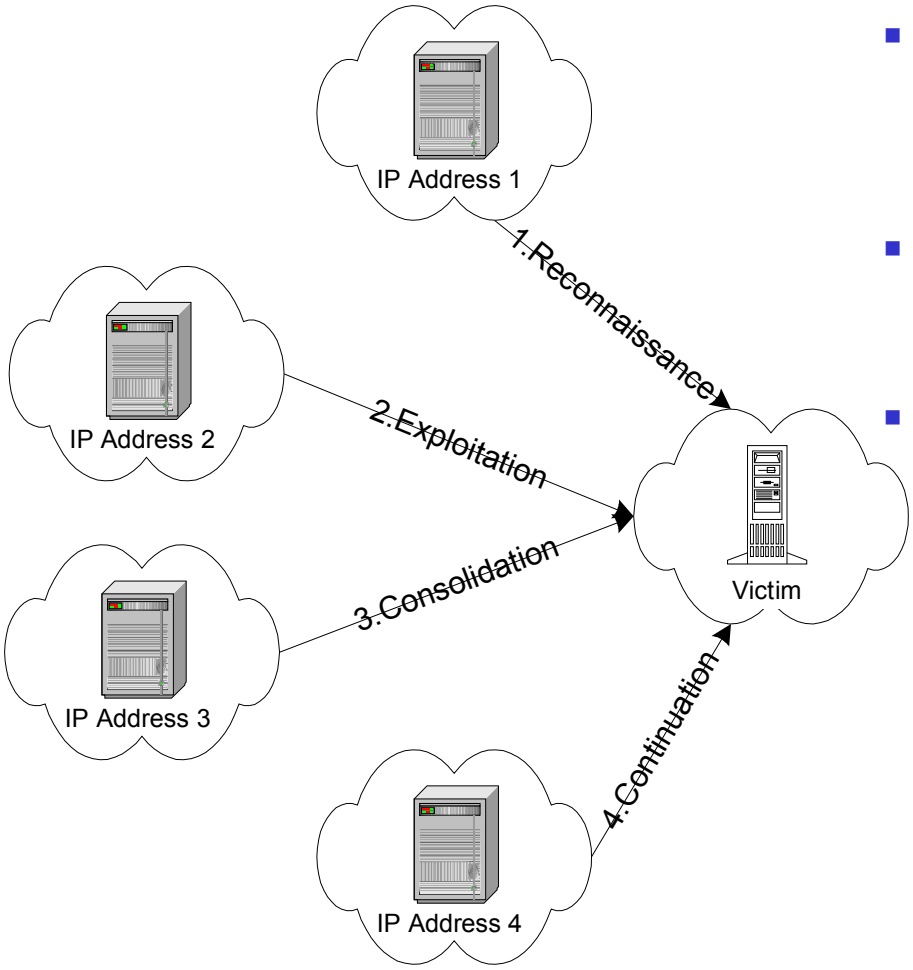
G.Valvis



Issues

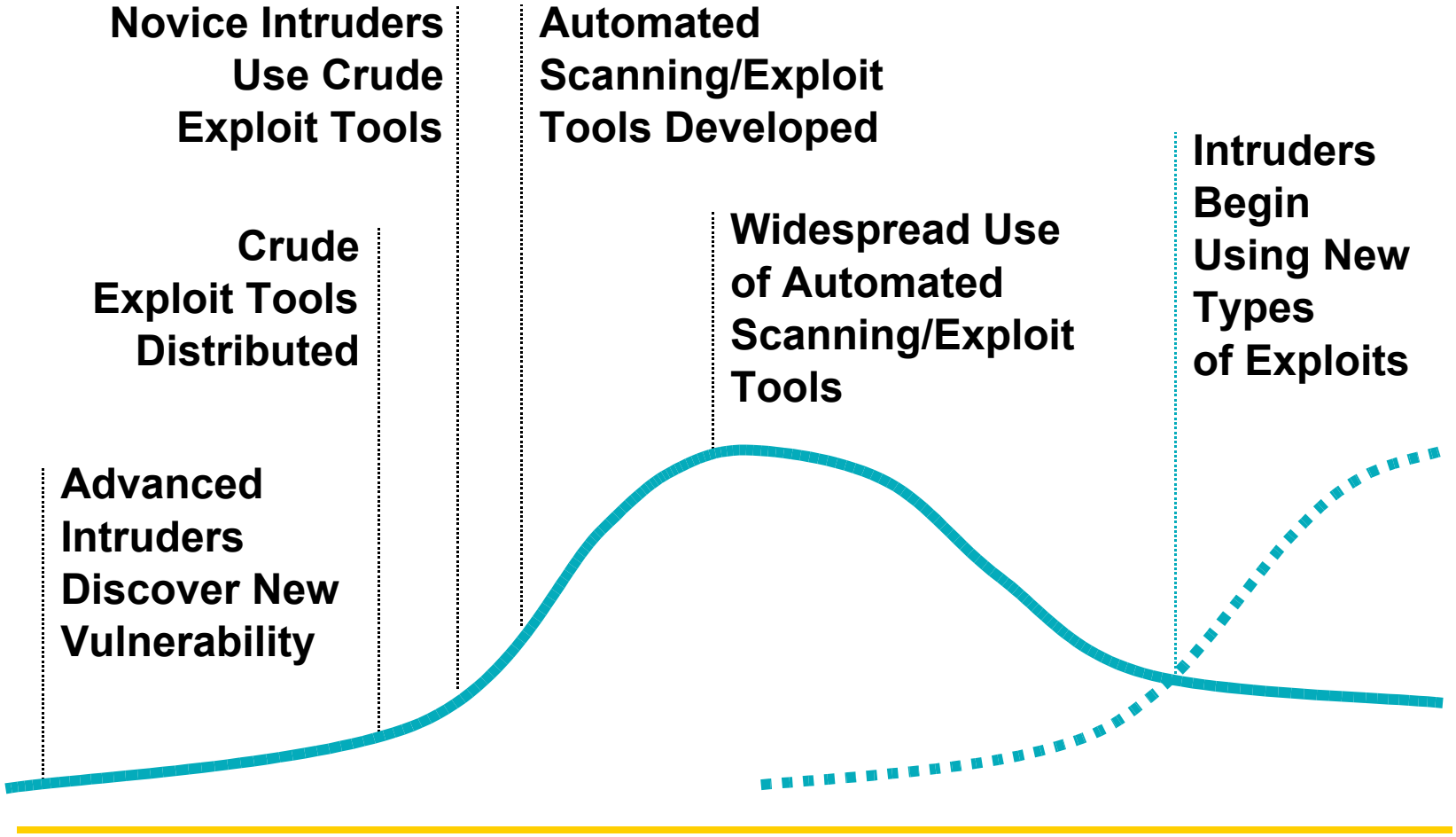
- Attack paradigm
- Vulnerability exploit life cycle
- Vulnerability assessment process
- Challenges in vulnerability assessment process
- VARF data model
- Vulnerability diagram
- Conclusions

Attack Paradigm

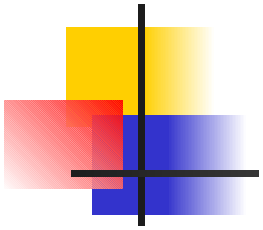


- Information gathering
  - Determination of the characteristics of the target network such as network topology, host OS type, listening services
- Exploitation
  - Compromise of a vulnerable host on the target network
- Metastasis
  - Consolidation
    - Remove any evidence of the exploitation phase, and to ensure that remote access is available to the attacker
  - Continuation
    - Utilize ‘passive’ as well as ‘active’ attack methods to deepen the penetration

Vulnerability Exploit Cycle

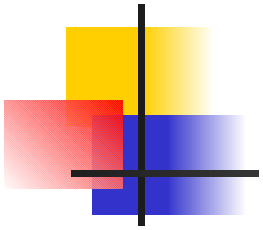


Source : CERT/CC (http://www.cert.org)



The vulnerability assessment process A.I.D.A.

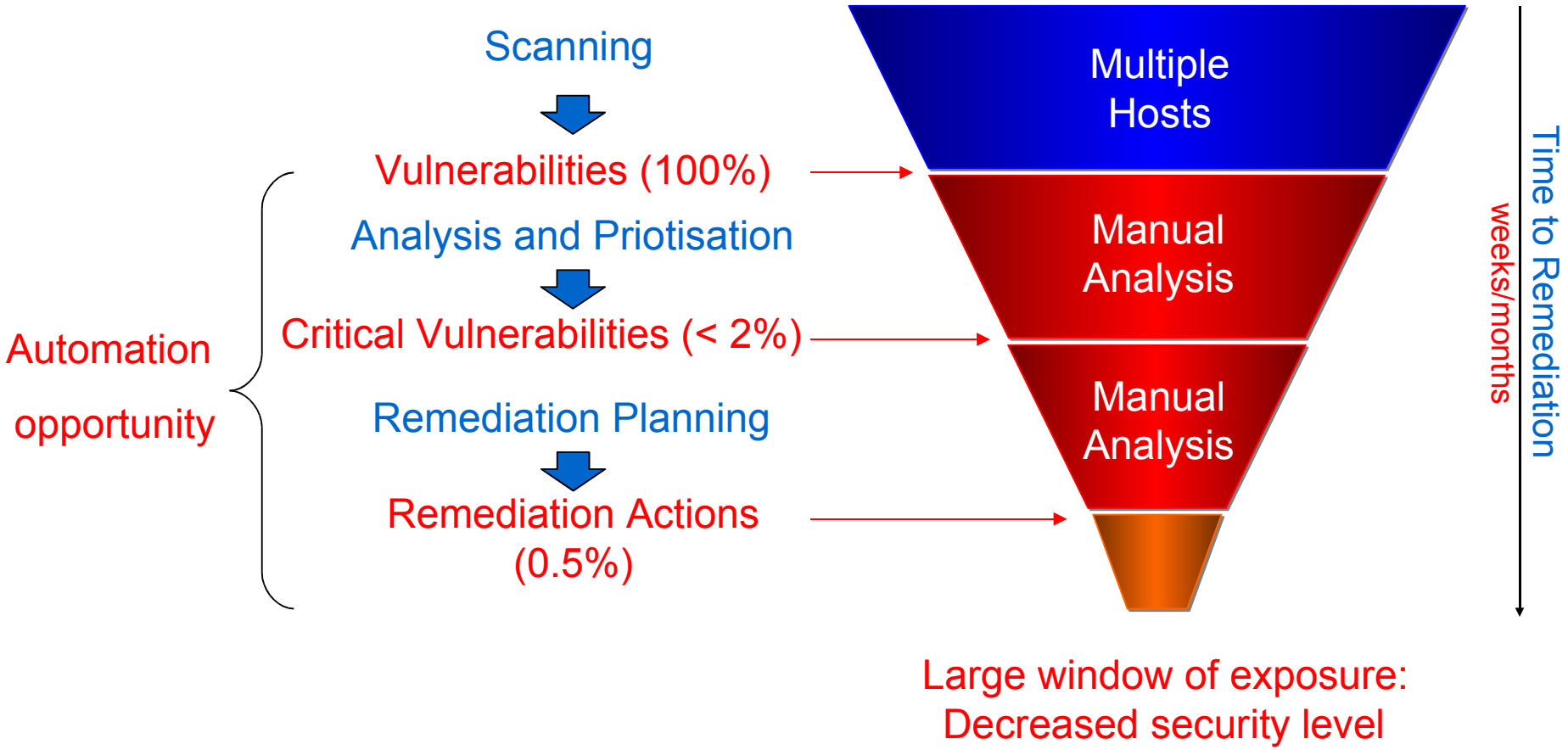
- Attention: Do we pay attention to our weak points ?
  - We find them by scanning our assets
    - Use vulnerability assessment tools for efficiency
    - In large networks different tools are deployed for more complete coverage
- Interest: How do we focus on the most interesting issues ?
  - Analysis and prioritization
    - A large number of vulnerabilities are of low risk or irrelevant to the specific environment
    - Critical vulnerabilities need to be dealt with priority
- Decision: Remediation planning
- Action: Patch management, etc.



Challenges in vulnerability assessment process

- For a complex IT environment most of the analysis work must be done by human
- Generate large volume of data
- Different vulnerability assessment tools provide heterogeneous output
- Effective communication between existing tools suffers by a lack of common ground
- Area of potential improvement

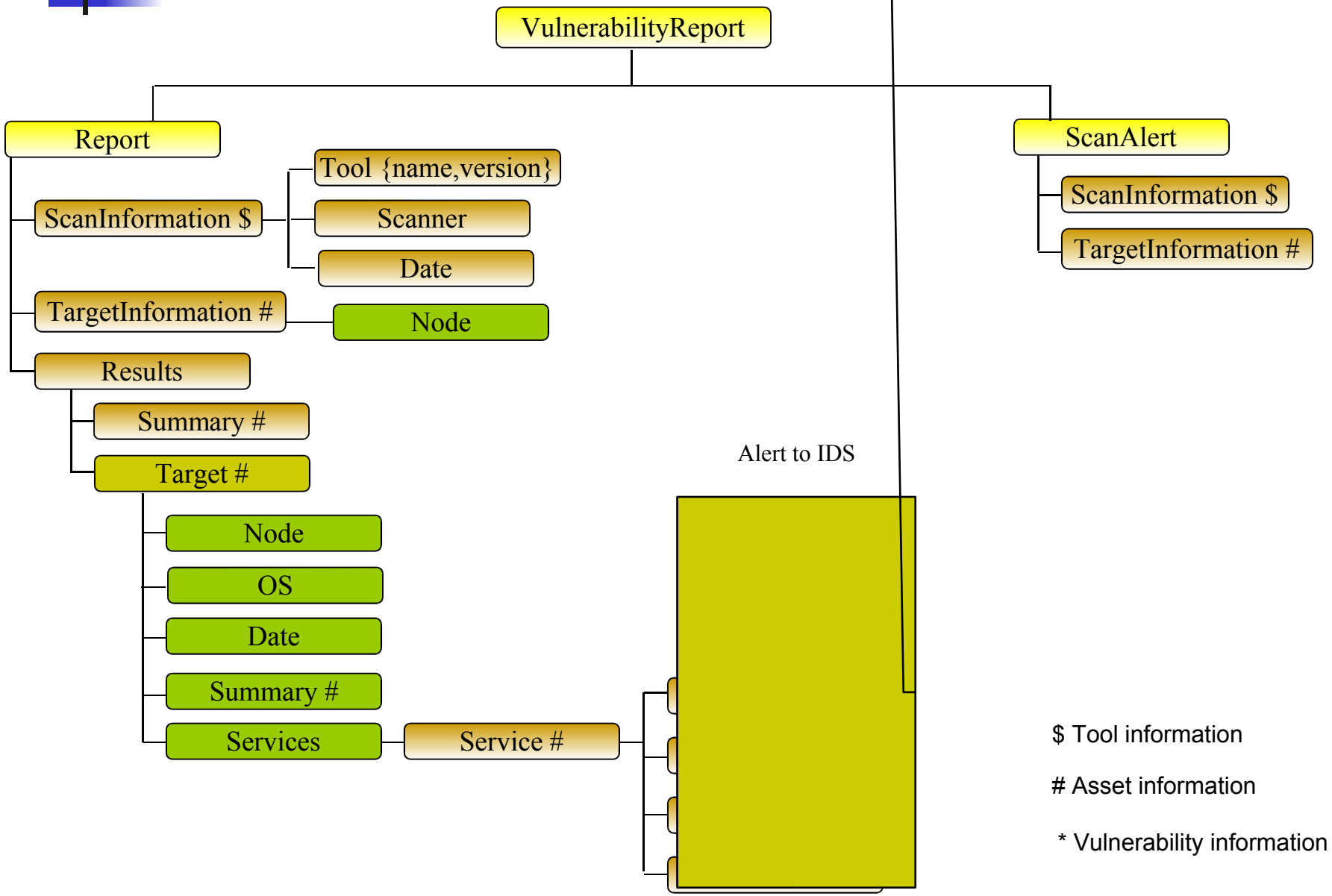
Challenges in Vulnerability assessment (cont.)



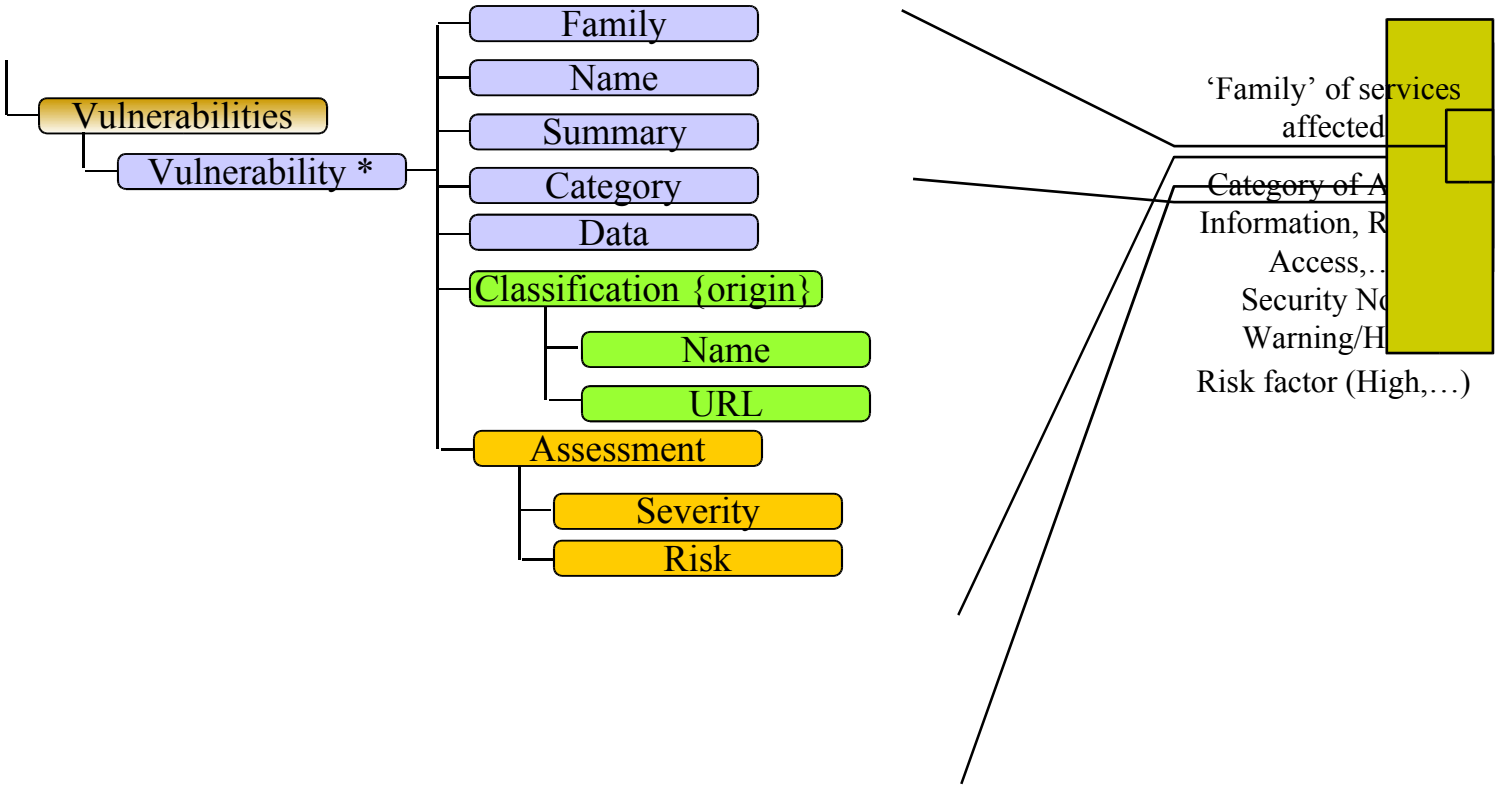
VARF: An attempt to address those issues

- The focus of the models is to facilitate the analysis and prioritization stage
- This model is based on a comparison of:
  - Latest versions of Nessus XML reports and SARA™ and
  - The latest Intrusion Detection Message Exchange Format (IDMEF) and Incident Object Description and Exchange Format (IODEF) drafts
- There was effort to reuse IDMEF elements
  - Either directly or by sub-classing them to add functionality
- The Vulnerability XML report is structured in order to
  - extract the asset information and
  - group the associated vulnerabilities
- The two main elements provided are the ScanAlert and Report

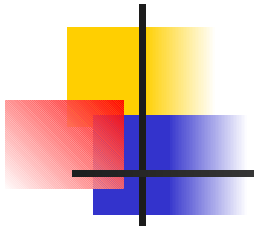
Vulnerability report model (cont.)



Vulnerability report model (cont.)



* Vulnerability information



<ScanAlert> Class

- **<ScanAlert >**
  - It is modeled on the IODEF IncidentAlert
  - Provides a different type of functionality
    - The IncidentAlert is used to simply alert someone/something to the occurrence of an incident and provide relevant information (such as raw IDMEF messages)
  - ScanAlert alerts an intrusion detection management system or other management system that a scan is going to be performed
  - As part of this alert, the scanner would provide ScanInformation and TargetInformation (detailed next)

<ScanAlert> Class (cont.)

- **<ScanInformation>**
  - It encapsulates information such as
  - the tool that is performing the scan, version of the tool
  - Information about the node that is being used to launch the scan,
  - Time information for documenting scan and a general description

- **<TargetInformation>**
  - This element documents the targets of the scan and contain the following items:
  - Address, name

Major <Report> classes

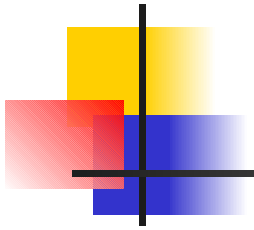
- **<Results>**
  - This element is meant to take the place of Nessus Results and SARA Details
  - It is closely tied to the IODEF Attack class, which in turn shares structure with IDMEF Alerts
- **<Target>**
  - Use of the IDMEF/IODEF Target class to achieve a standard format for representing the 'host' specific information
  - It includes
  - the <Node> class which contains address and name elements
  - <OS> element (type of operating system), <date> element
- **<Service>**
  - This class generically describes network services
  - A network service is defined by name and port
  - It includes the <vulnerabilities> class, since one service may have multiple vulnerabilities

<Vulnerability> Class

- **<Vulnerability>**
  - This class describes vulnerability by
  - Name
  - Family of services affected (e.g. FTP)
  - Category of attack (e.g. Information, Access, etc.)
  - It includes the <Classification> and <Assessment> classes and additional data

- **<Classification>**
  - Allows the manager who receives the Report messages to be able to obtain additional information
  - Origin (CVE, Bugtruq) of the source, name and URL are included

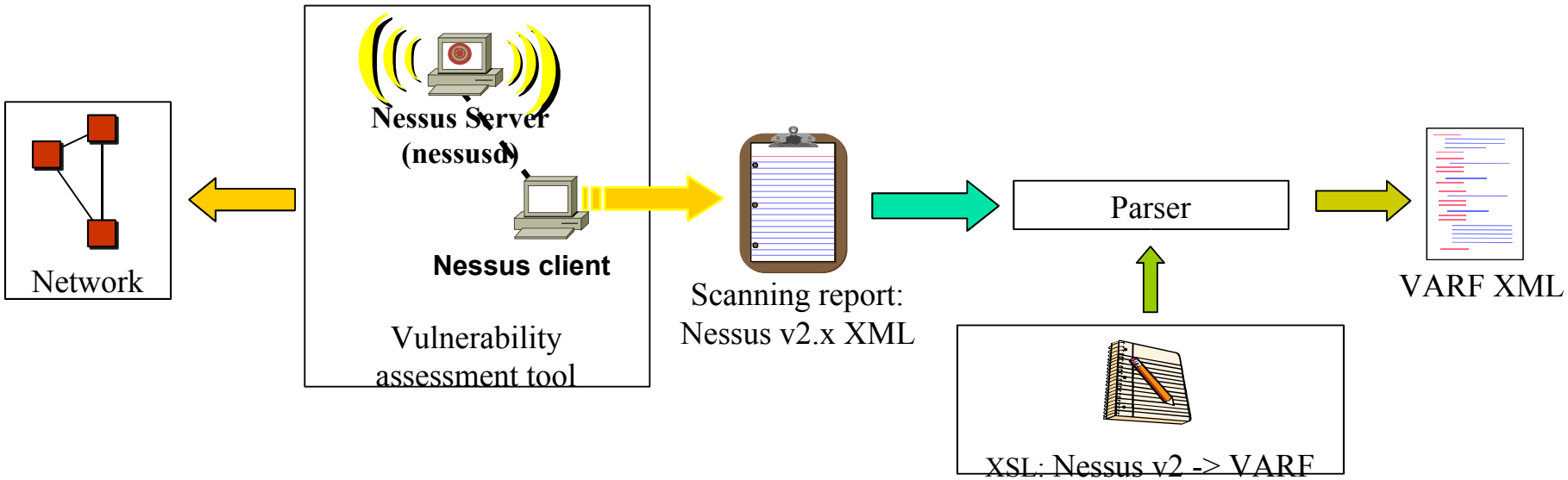
- **<Assessment>**
  - It provides information related to the scanner's assessment of the vulnerability
  - Includes the elements <Risk> and <Severity>



XSL transformations

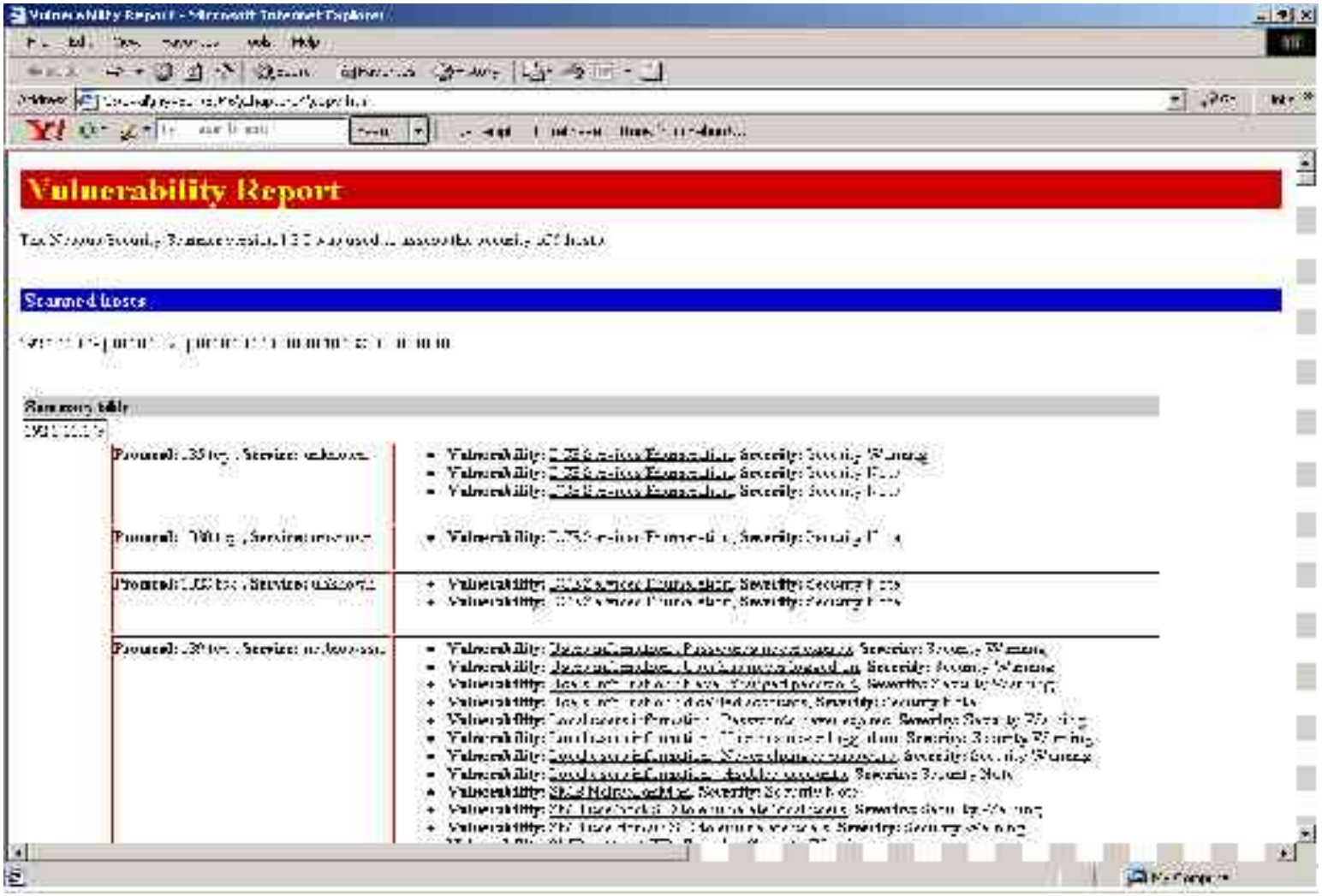
- Generate VARF XML
- HTML presentation
- Creation of vulnerability diagram: visual representation of association between assets and vulnerabilities

XSL Generate transformations

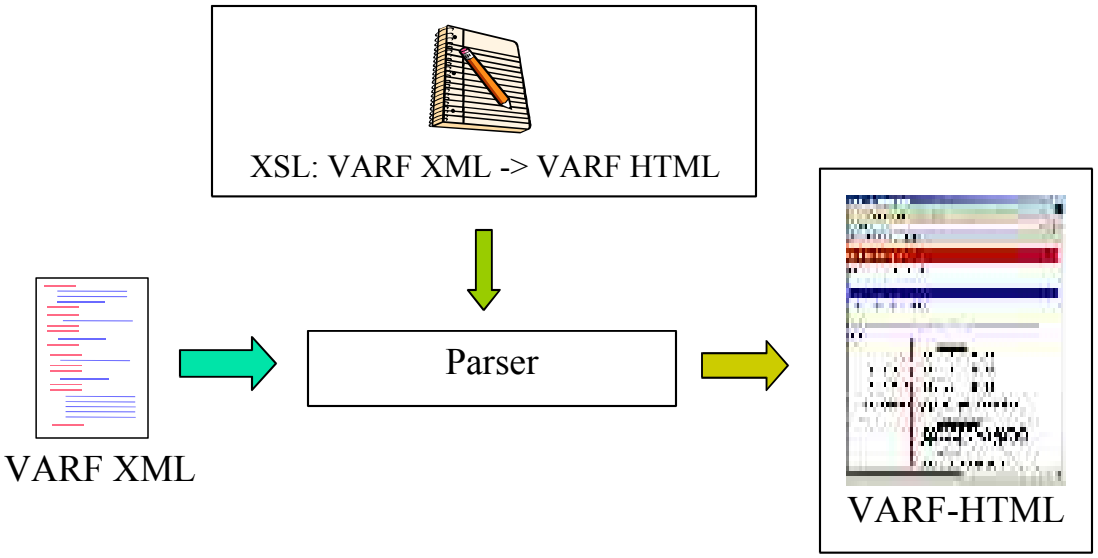


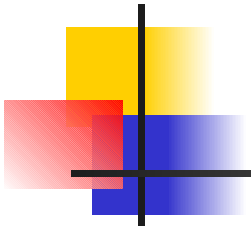
HTML presentation

Dynamic XSLT (client side XSLT transformations)



HTML presentation (cont.)

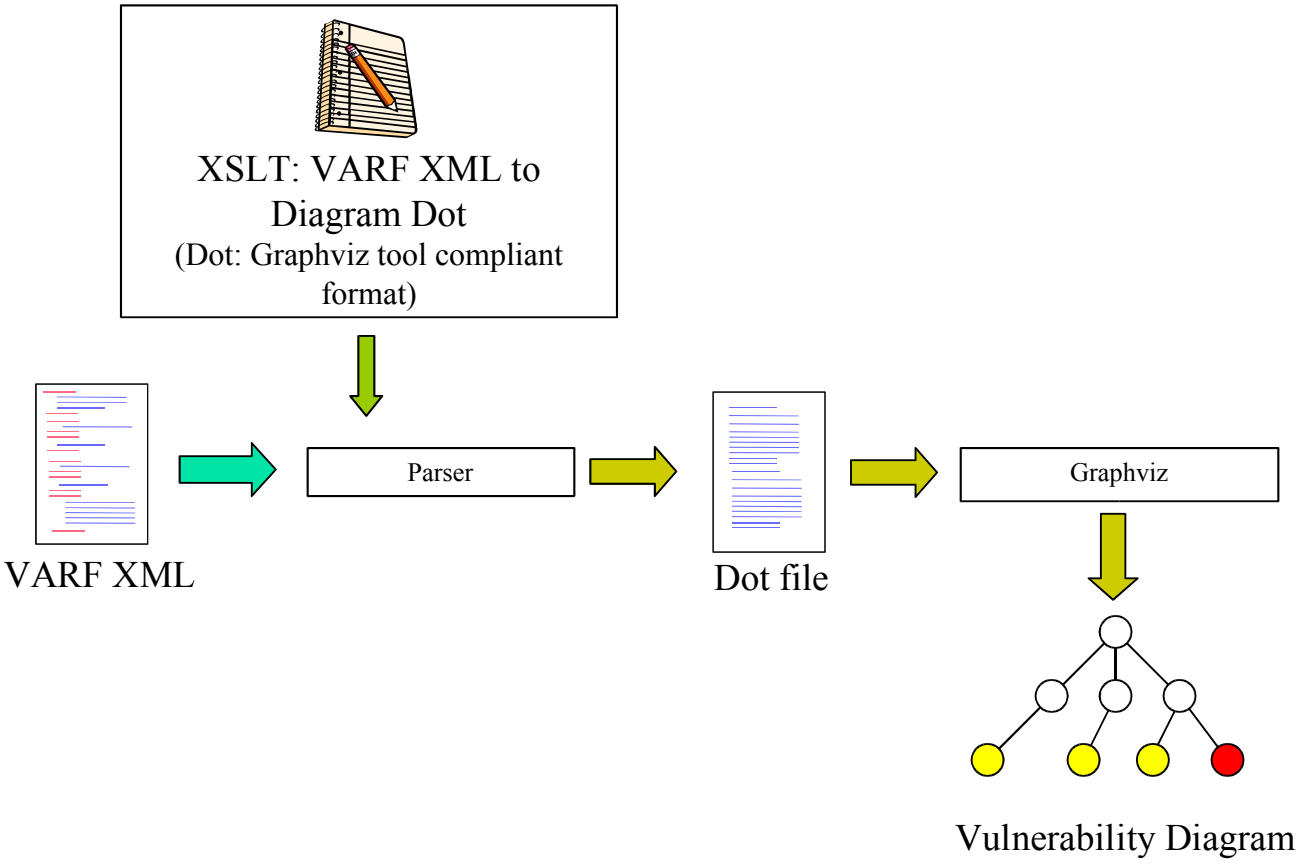




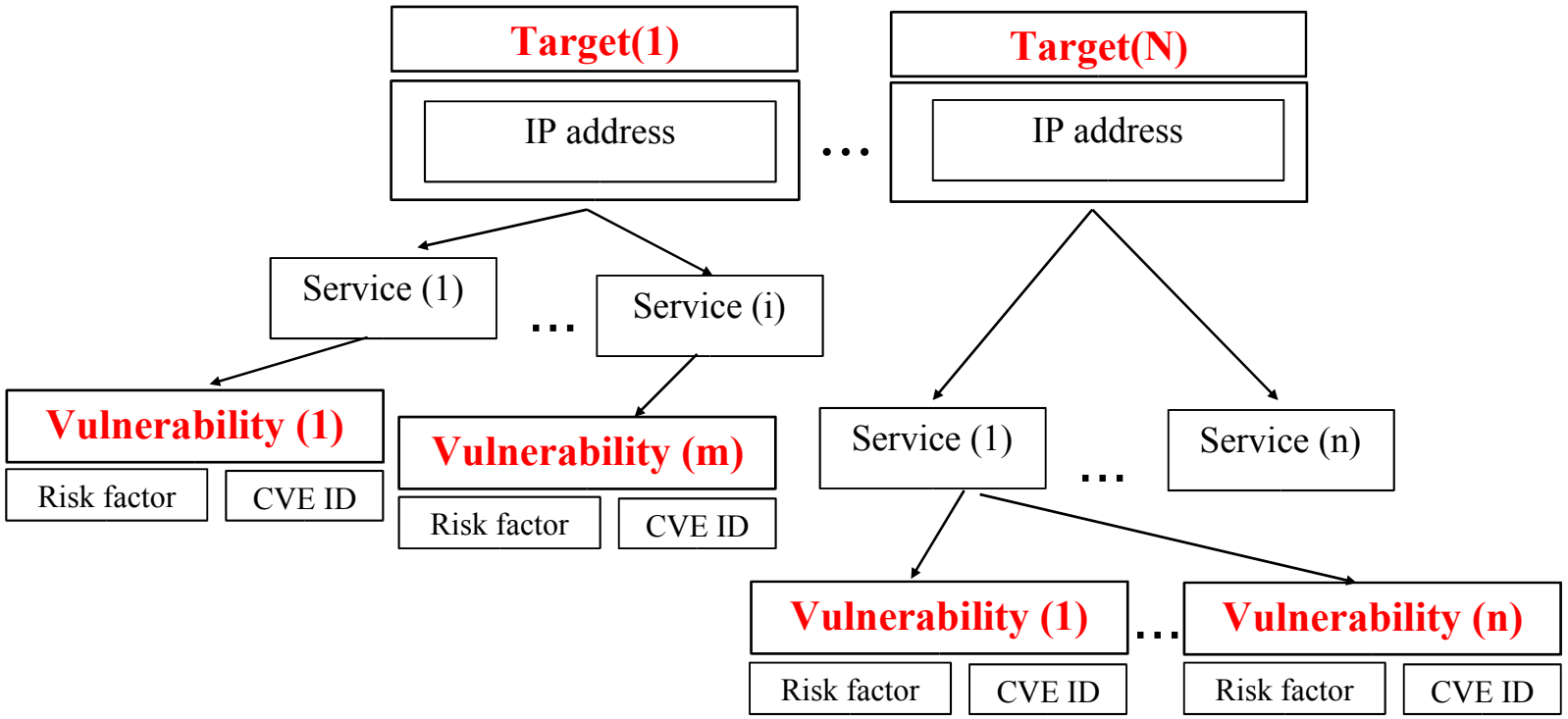
Vulnerability Diagram

- XML represents data in tree
  - Hard for human to understand
  - Lessen the burden by visualization
- Complete vulnerability diagrams
  - Shows all discovered vulnerabilities, but structures are very large
    - Hard to scale
- Reduced vulnerability diagrams
  - Cut sets of vulnerabilities
    - Which services, if suspended, leave the network secure?
  - Results inform administrator which services are, perhaps, too costly.
- Vulnerability diagram can be a subset of attack tree
  - Subsequent analysis is possible

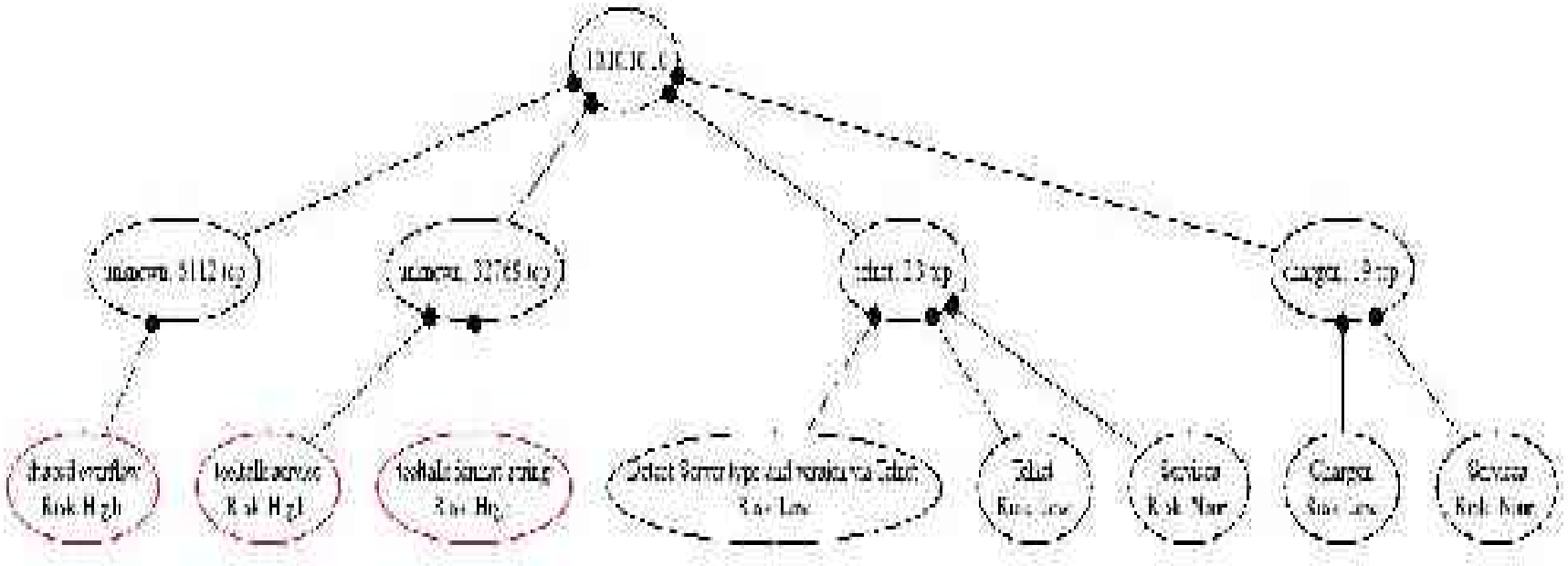
Vulnerability Diagram



Vulnerability diagram (concept)



Vulnerability diagram (example of actual results)



Conclusions

- In order to reduce the window of exposure, the security personnel need a way to set priorities and reduce the volume of vulnerability reports down to the few critical risks that matters.
- Due to proprietary nature of the reports and lack of standardization, this process is burdensome.
- Standards based format to report vulnerabilities would allow easier analysis and sharing of information with other data sets from a variety of compliant tools and systems.
  - VARF was motivated from the above and was based on existing standardization efforts.
- Vulnerability diagrams visualize the vulnerability management effort.