

Master's Thesis Final Presentation
February 24th, 2009

Robert Schuppenies

*Automatic Extraction of
Vulnerability Information for Attack Graphs*



Understanding



Technique



Contribution



Agenda

- × Vulnerabilities & Attack Graphs
- × Problem Statement
- × Vulnerability Information Representation
- × Vulnerability Information Transformation
- × Proof of Concept
- × Conclusion

Agenda

- × **Vulnerabilities & Attack Graphs**
- × Problem Statement
- × Vulnerability Information Representation
- × Vulnerability Information Transformation
- × Proof of Concept
- × Conclusion

Vulnerabilities, cont.



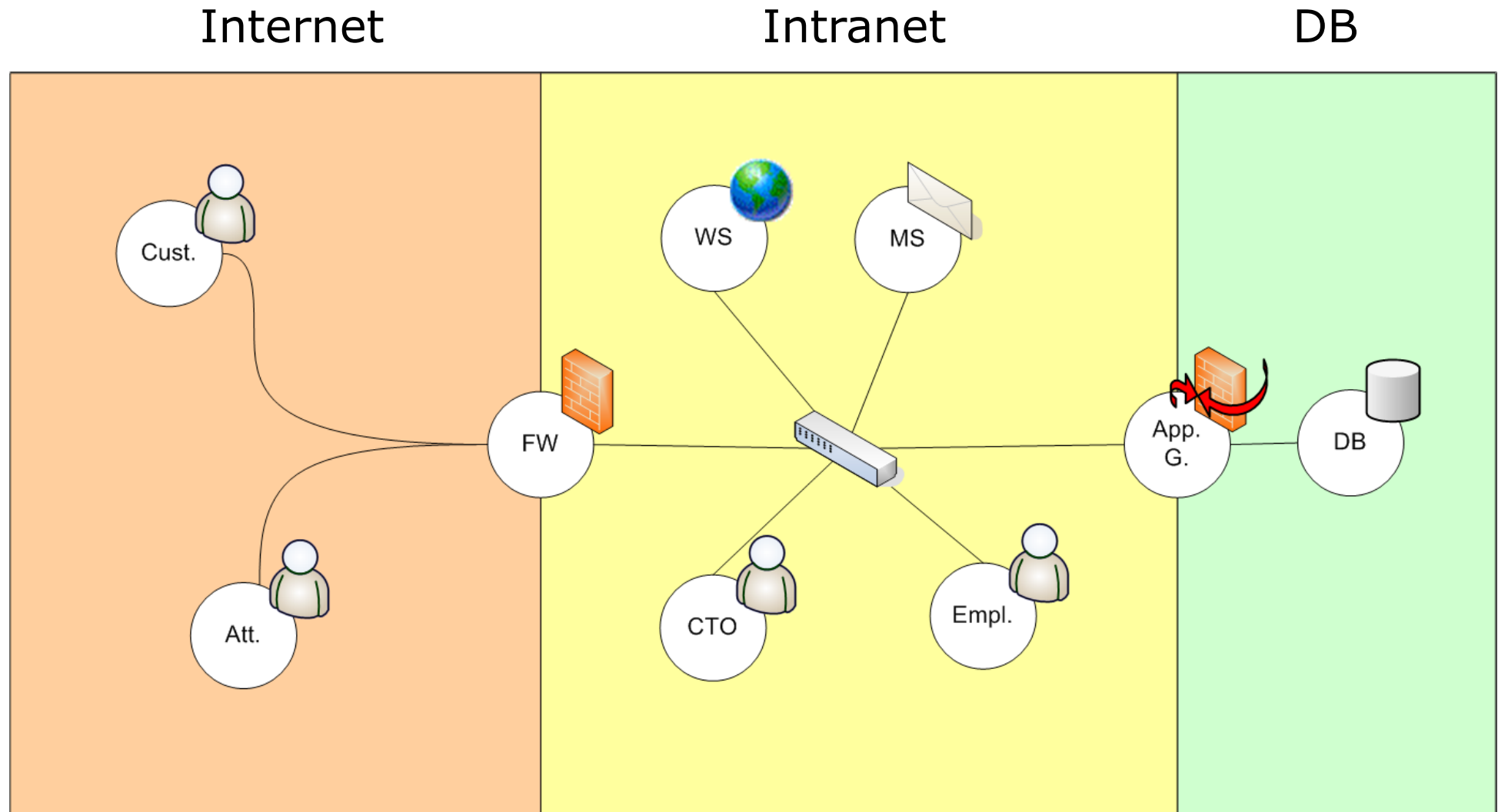
- × **Vulnerability** : **A Weakness of a system**
- × **Exploit** : Makes use of a weakness
- × **Mitigation** : Remedies a weakness

- × **Confidentiality** : Accessible only to authorized entities¹⁾
- × **Integrity** : Modified only by authorized entities¹⁾
- × **Availability** : Accessible/Usable when needed¹⁾

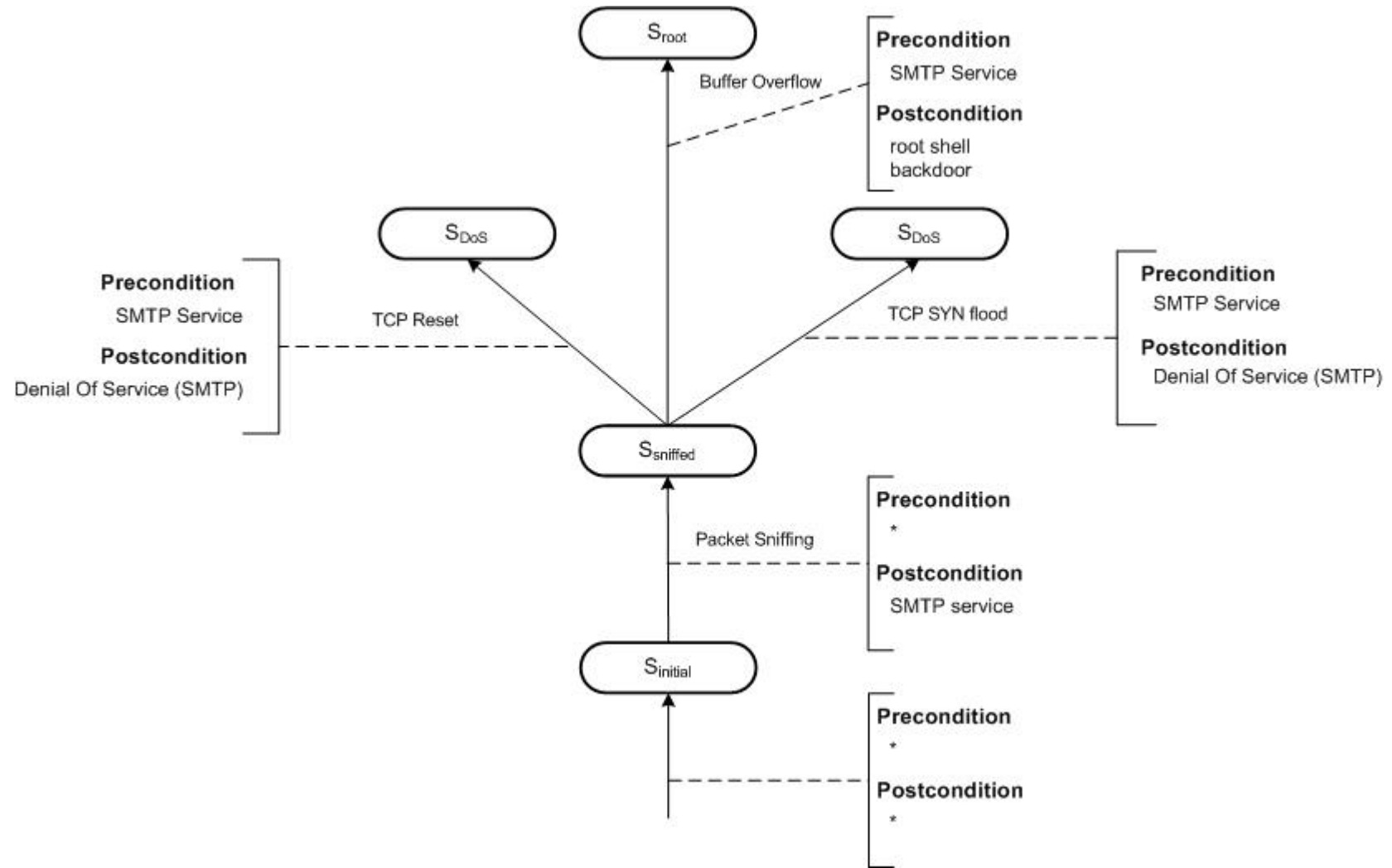
- × Vulnerability Databases (VDBs)
 - × Entries written **by** humans **for** humans

¹⁾ NIST: "Engineering principles for information technology security"

Multi-step Attacks



Attack Graph example

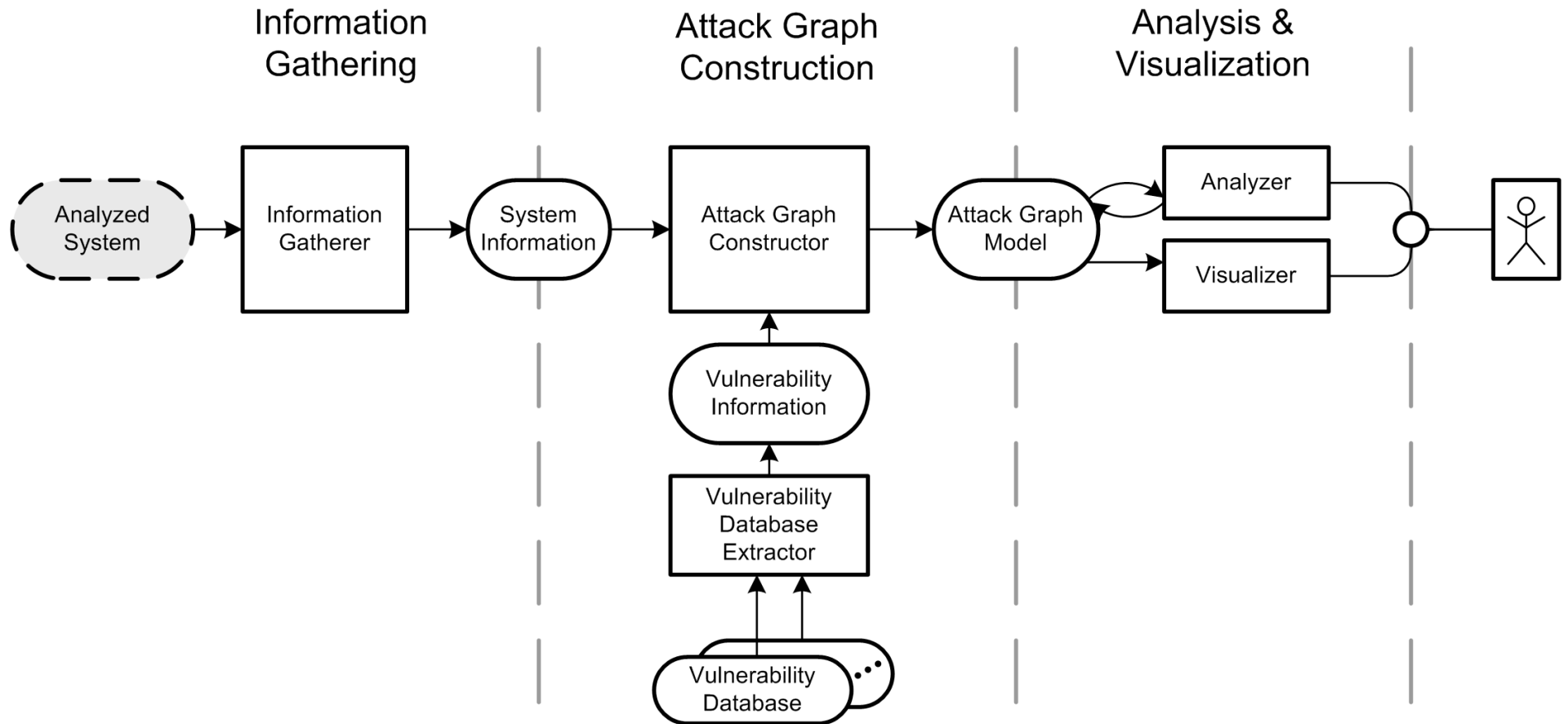


Attack Graphs - Benefits



- × Allow to describe attack combinations
- × Find the shortest path
- × Identify pivotal points in a graph
- × Cost/benefit analysis for network design
- × Correlate “unrelated” events to identify attacks

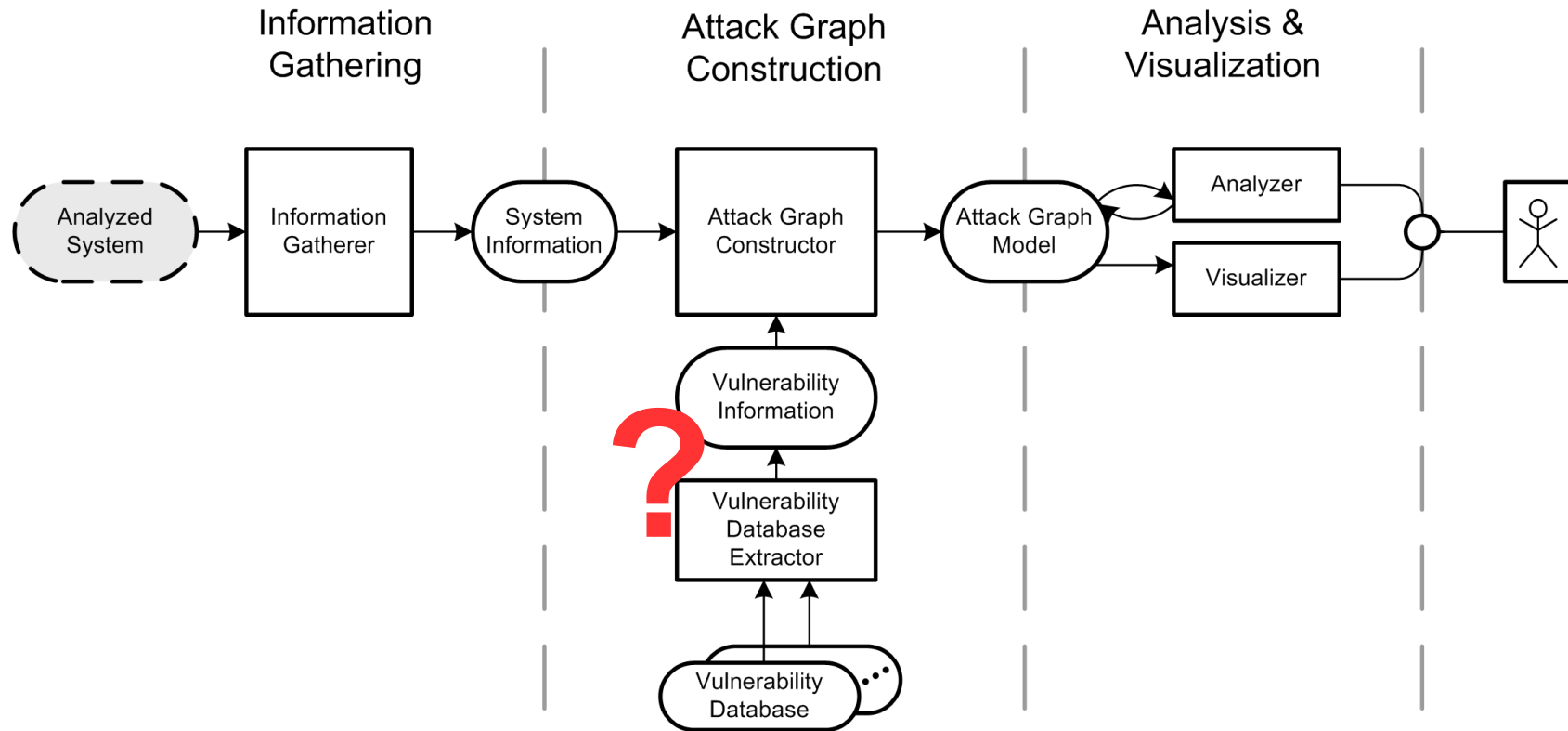
Attack Graphs - Workflow



Agenda

- × Vulnerabilities & Attack Graphs
- × **Problem Statement**
- × Vulnerability Information Representation
- × Vulnerability Information Transformation
- × Proof of Concept
- × Conclusion

Problem Statement

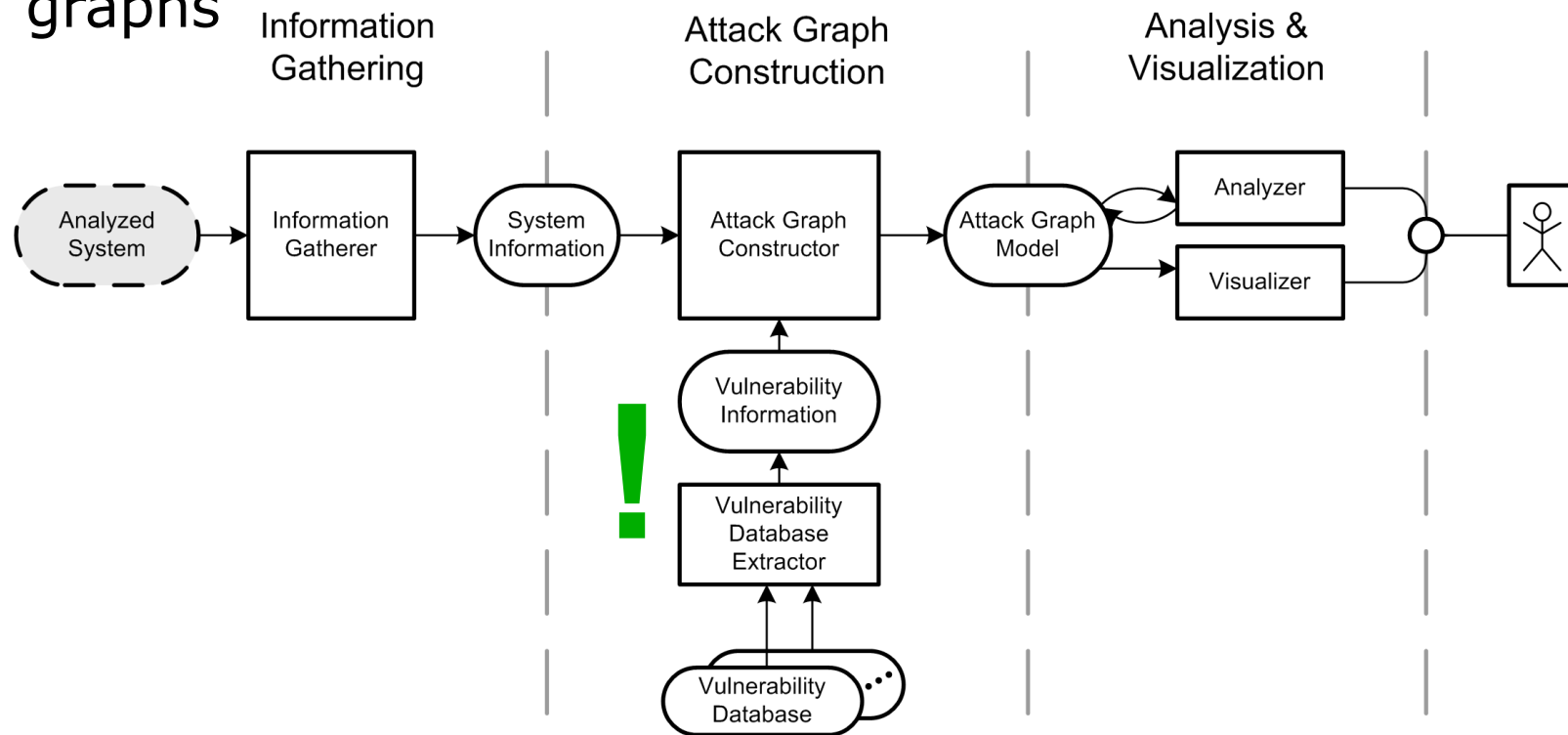


- ✘ Attack graph models have ..
- .. No automatic extraction of attack pre- and postconditions
- .. Very simple or too complex attack models

Master's Thesis - Objectives

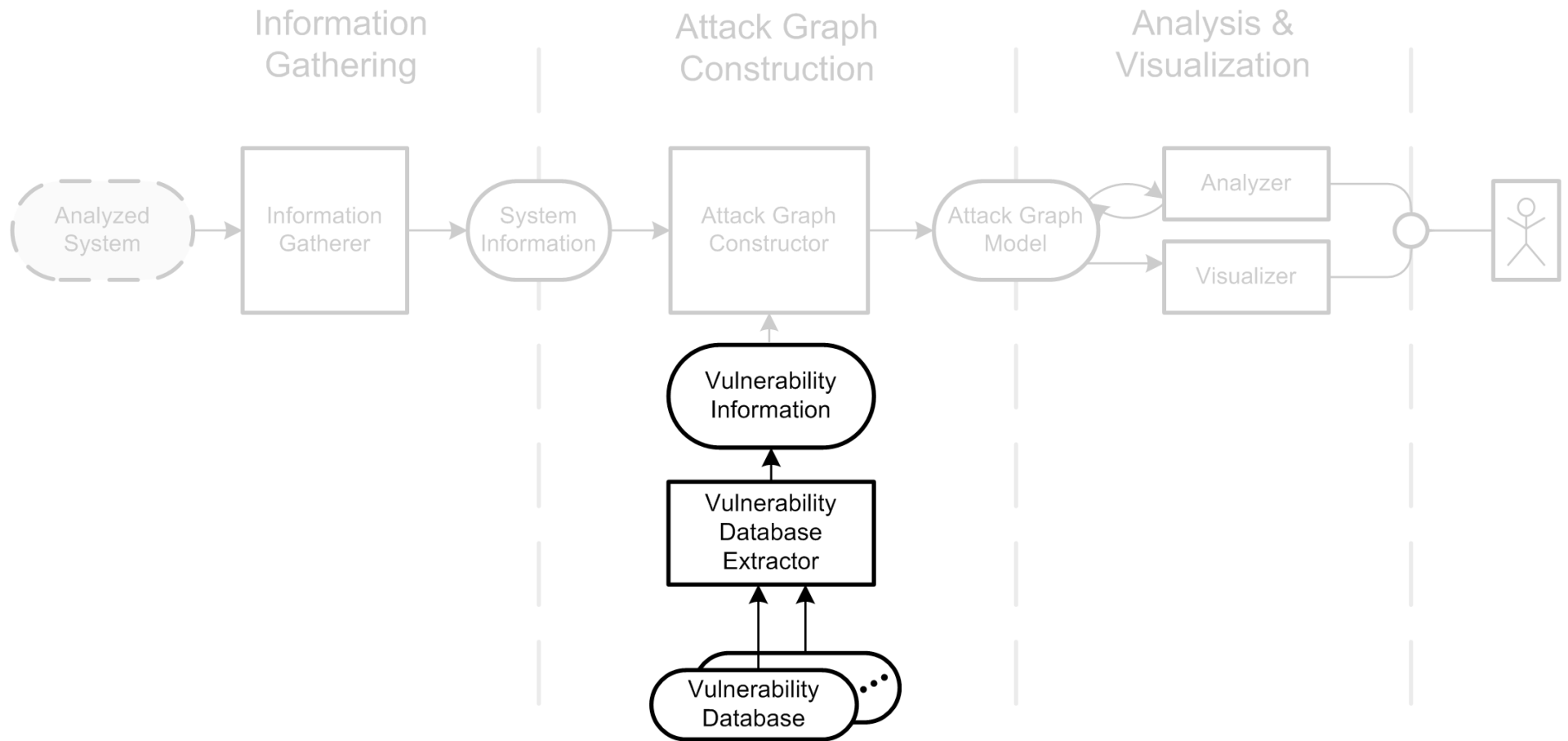


- ✘ Provide data structure to link vulnerabilities
- ✘ Automatic extraction of vulnerability information for attack graphs



pre 0 S_{initial} post A \rightarrow pre A S_{sniffed} post B \rightarrow pre B S_{DoS} post C

Attack Graphs - Workflow



Master's Thesis - Steps

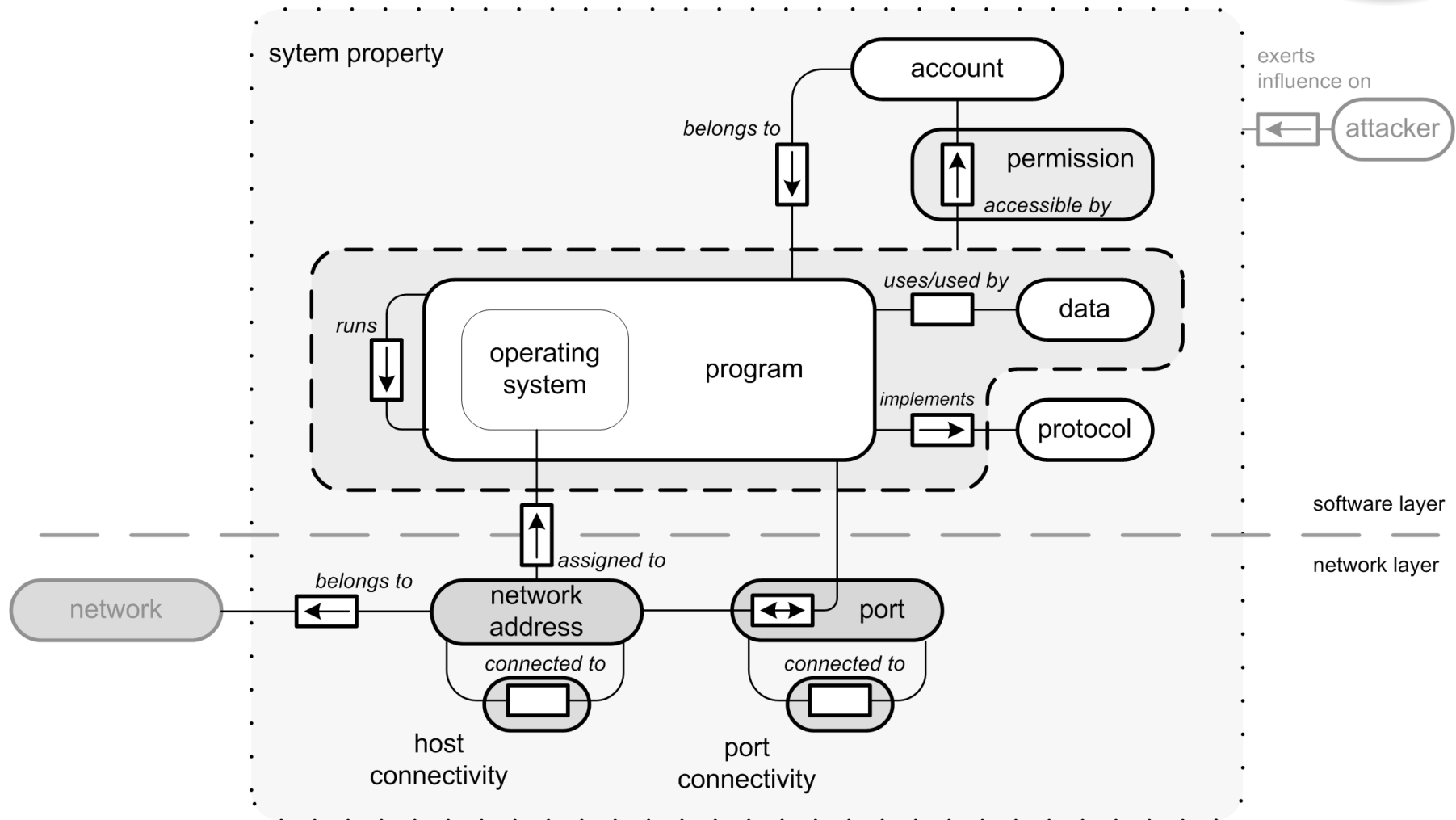


- 1.** Propose data structure
- 2.** Investigate & Extract available VDB information
- 3.** Implement prototype
- 4.** Proof concept with existing attack graph tool

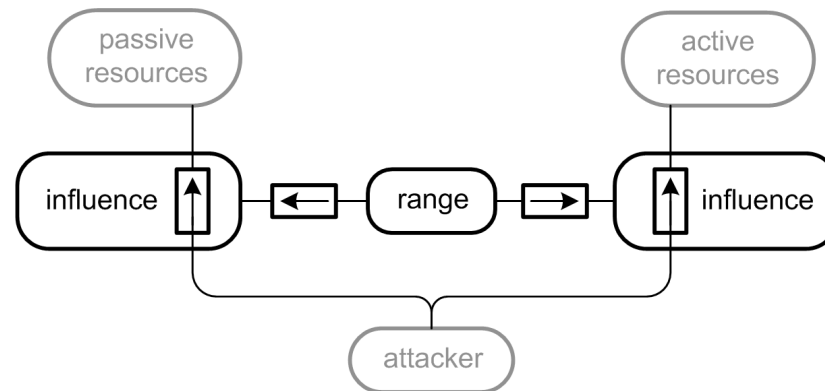
Agenda

- × Vulnerabilities & Attack Graphs
- × Problem Statement
- × **Vulnerability Information Representation**
- × Vulnerability Information Transformation
- × Proof of Concept
- × Conclusion

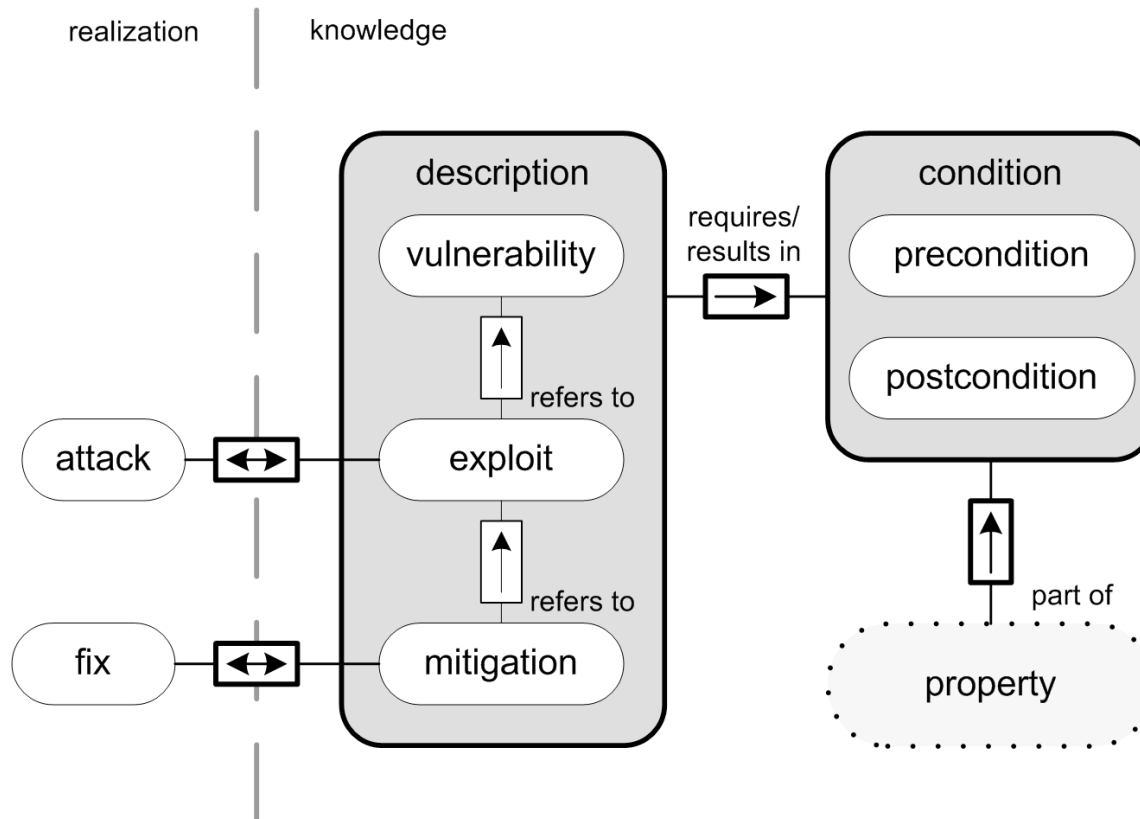
Data Structure – System Properties



Data Structure – Influence Properties

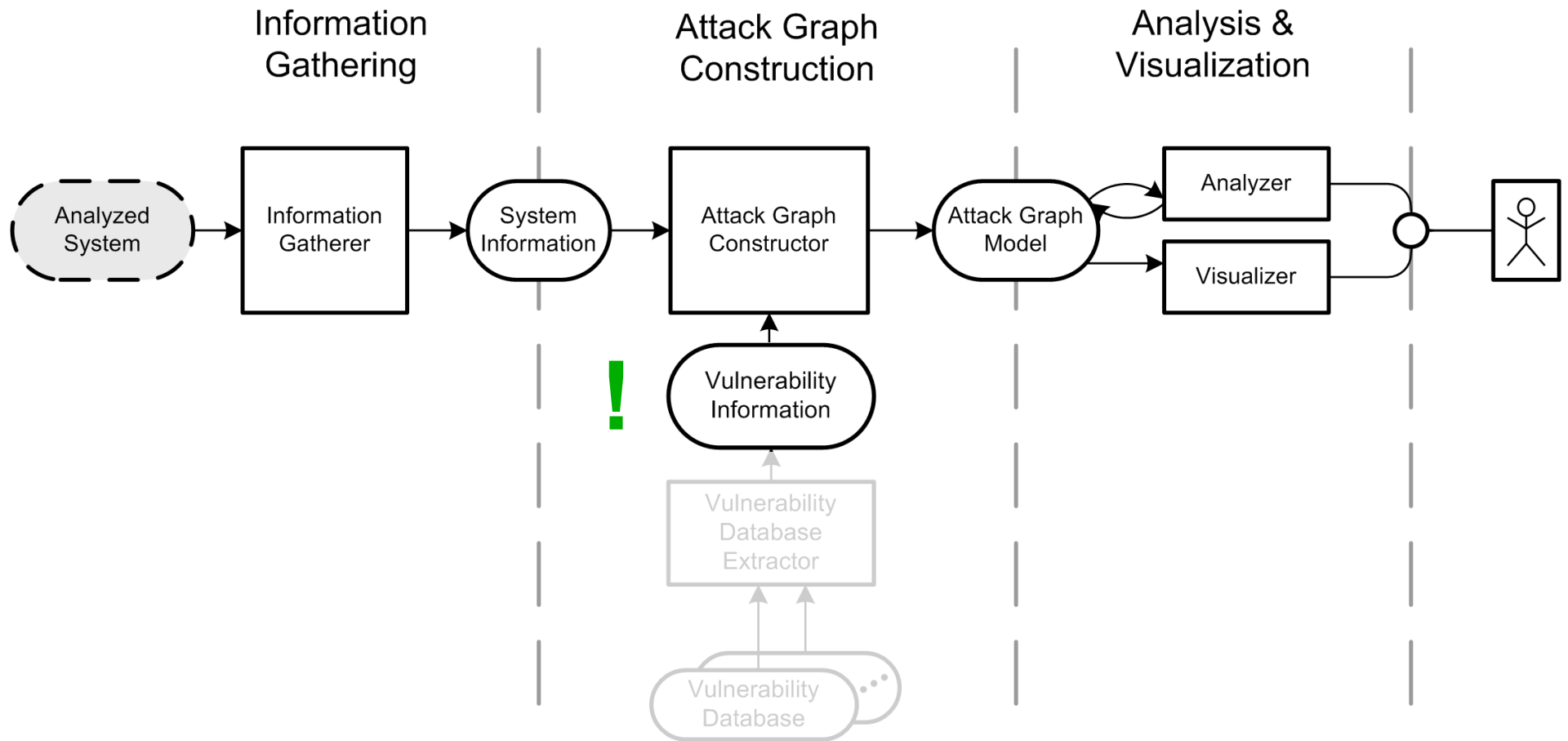


Data Structure – Conceptual View



property: program, account, data, ...

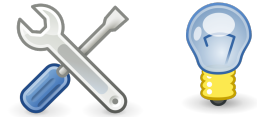
Attack Graphs - Workflow



Agenda

- × Vulnerabilities & Attack Graphs
- × Problem Statement
- × Vulnerability Information Representation
- × **Vulnerability Information Transformation**
- × Proof of Concept
- × Conclusion

Provided Information by VDBs



	D.Soft	S.Focus	Secunia	Securit.	X-Force	CoopVDB	DoE-CIRC	NVD	OSVDB	US-CERT
vendor-specific ID	x	x	x		x		x		x	x
CVE reference	x	x	x	x	x	x	x	x	x	x
title	x	x	x	x	x	x	x		x	x
description	x	x	x	x	x	x	x	x	x	x
range	x	x	x	x	x	x		x	x	
OS	x	x	x	x	x	x	x	x	x	x
software		x	x	x	x	x		x	x	x
CVSS	x				x		x	x		
critical	x		x	x	x		x	x		x
impact	x		x	x	x	x	x	x	x	x
authentication class		x					x	x	x	
access complexity								x		
references	x	x	x	x	x	x	x	x	x	x
format (Y)	H	H	H	H	H	H	H	H, X	C, H, M, S, X	H
exploit	x	x				x		x	x	
solution status	x	x	x	x	x	x	x	x	x	x
solution	x	x	x	x	x	x	x		x	
release date	x	x	x		x		x	x	x	x
last update		x	x					x	x	x
popularity			x						x	

Considered Formats

- × HTML
 - × not standardized syntax and semantics

- × Common Vulnerability Scoring System (CVSS)
 - × Base Metrics, Temporal Metrics, Environmental Metrics
 - × standardized syntax and semantics

- × Open Vulnerability and Assessment Language (OVAL)
 - × system configuration descriptions
 - × standardized syntax only

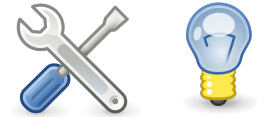
- × Text Parsing
 - × no standard, but surprisingly uniform

✘ Vulnerability Attributes contained in Textual Descriptions:

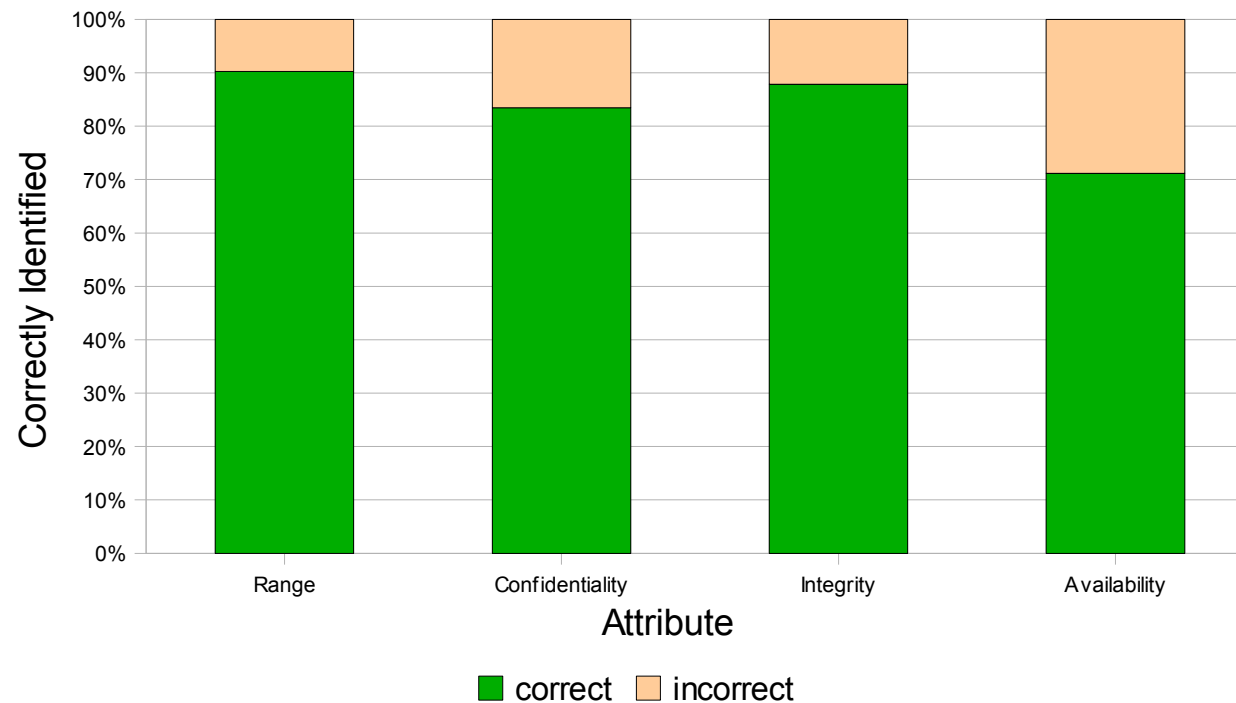
"The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2, Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary code via a crafted RPC request that triggers the overflow during path canonicalization, as exploited in the wild by Gimmiv.A in October 2008, aka "Server Service Vulnerability." "

✘ Identify Attributes based on context

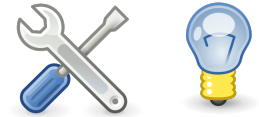
✘ Comparison based on CVSS entries



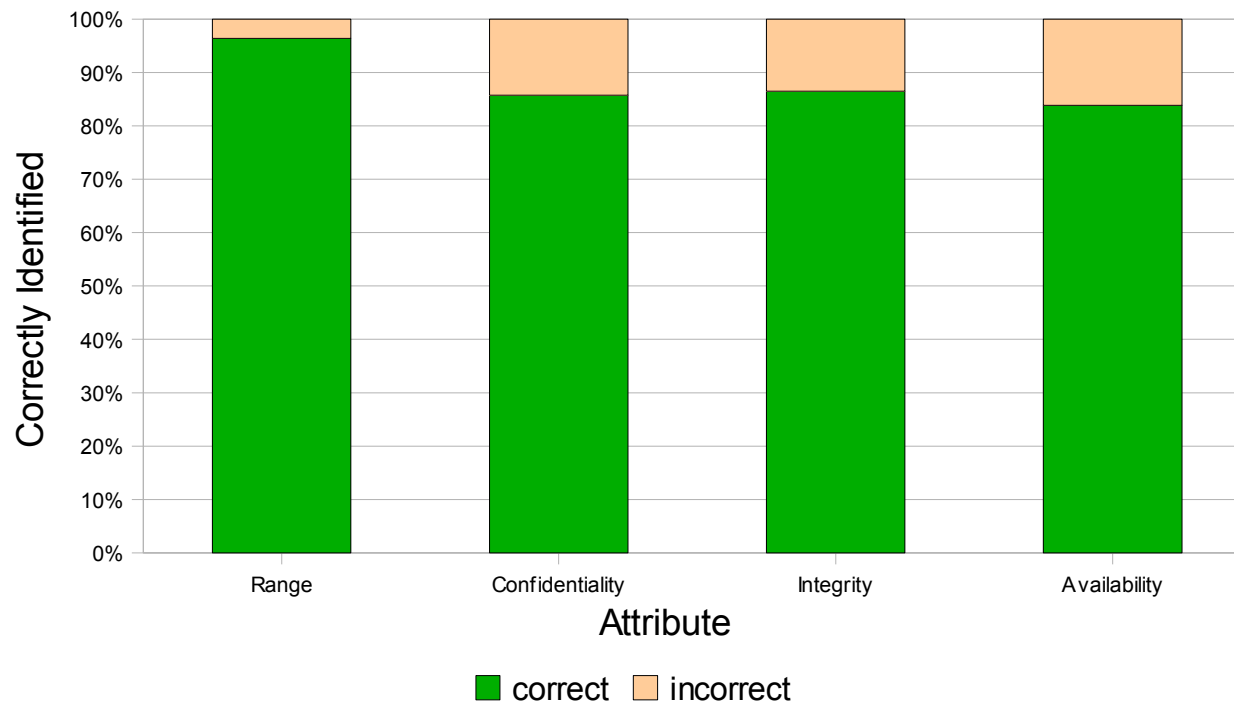
Extraction from Descriptions



- ✘ range: assume remote range if not specified
- ✘ CIA : ignore cross-site scripting entries

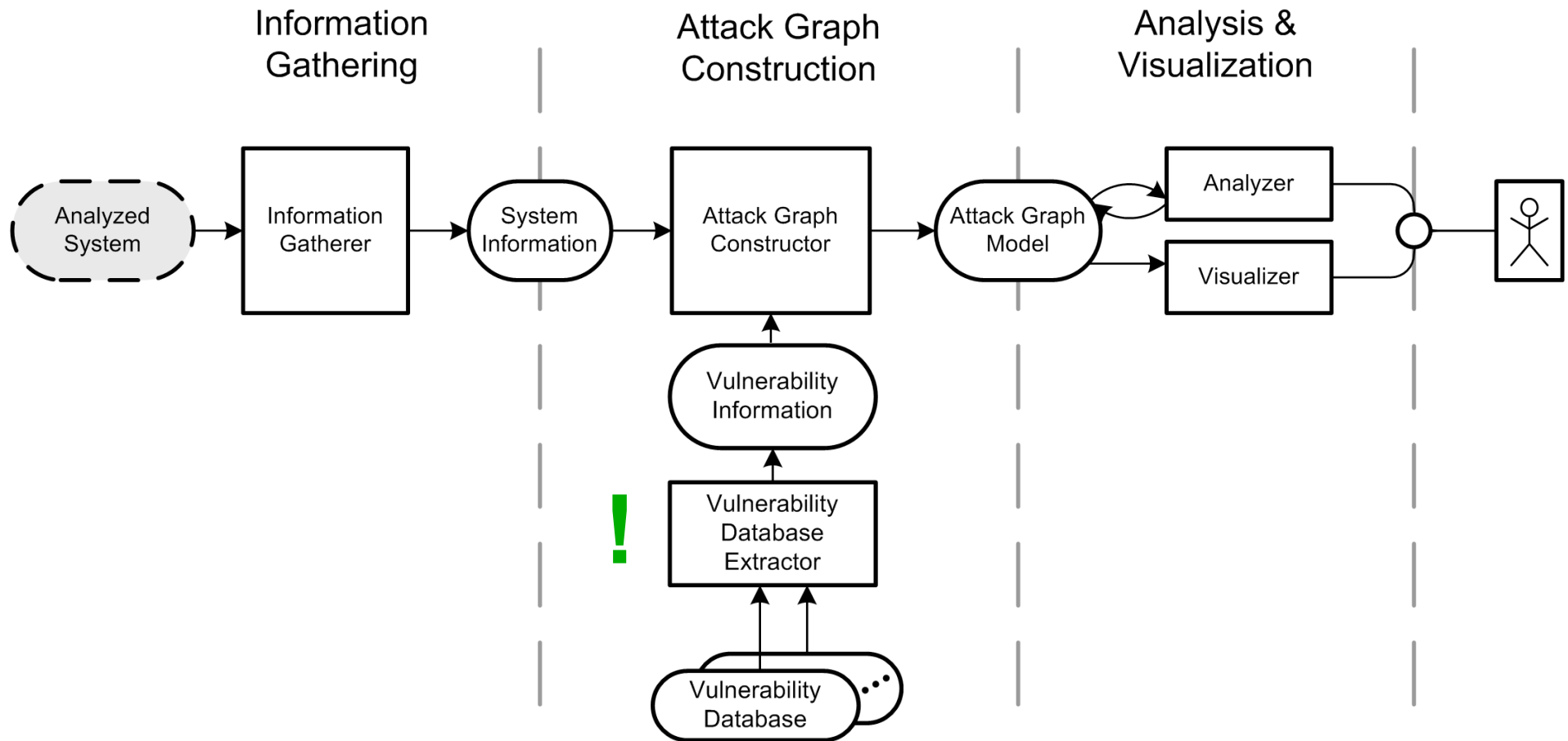


Extraction from Descriptions (with assumptions)



- ✘ range: assume remote range if not specified
- ✘ CIA : ignore cross-site scripting entries

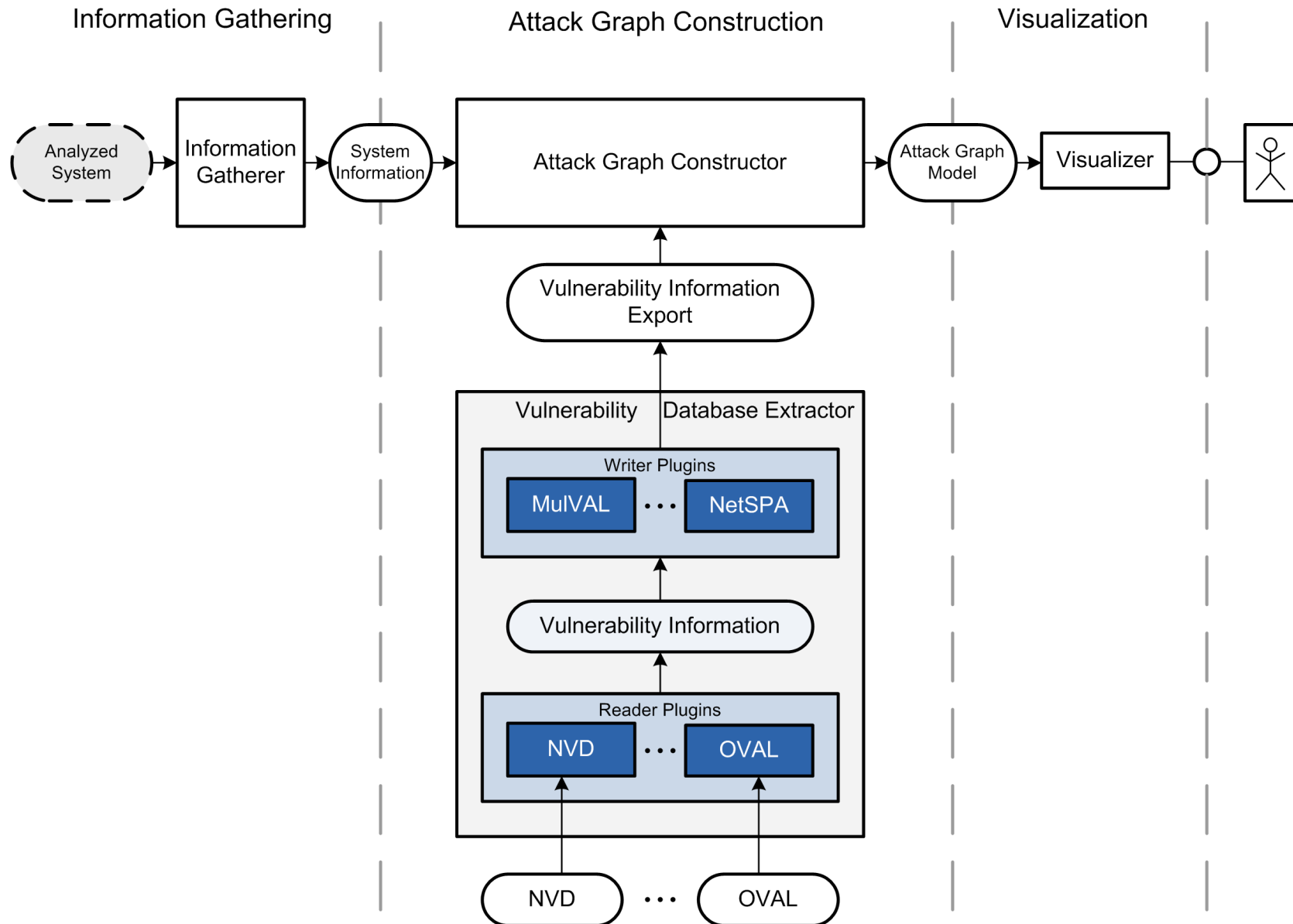
Attack Graphs - Workflow



Agenda

- × Vulnerabilities & Attack Graphs
- × Problem Statement
- × Vulnerability Information Representation
- × Vulnerability Information Transformation
- × **Proof of Concept**
- × Conclusion

Proof Of Concept - Design



Proof Of Concept – A Web-Frontend

Vulnerability Database Converter



Step 1: Configuration

Available Readers

- BinaryReader
- CVEReader
- NVDReader
- OVALReader
- XMLReader

Load Available Readers

Source Data

- Windows
- IOS
- Unix
- Pixos

Available Writers

- BinaryWriter
- MulVALWriter
- XMLWriter

Load Available Writers

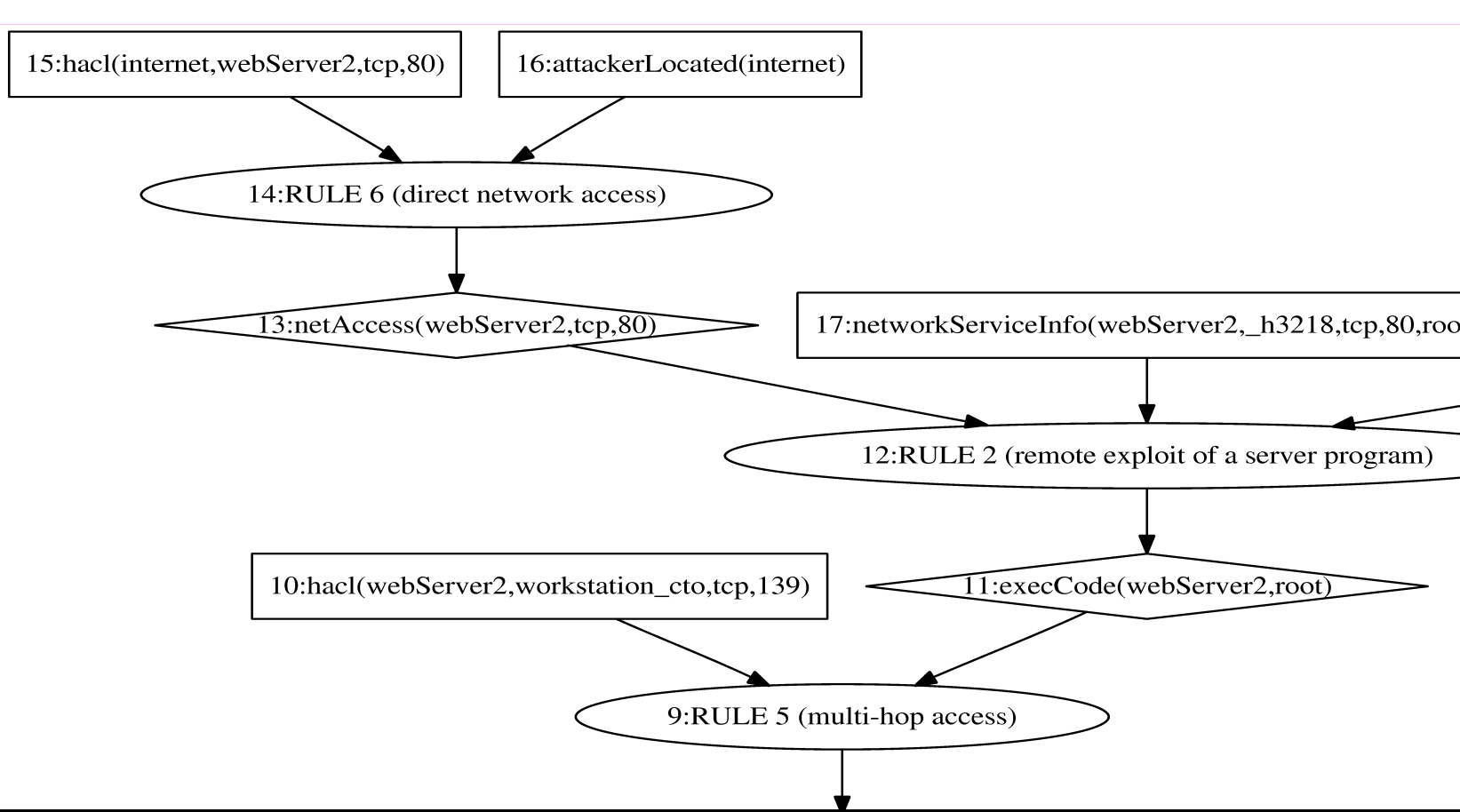
Extract Vulnerability Information

Demonstration with MuIVAL



```

bob@cronus[~/data/studium/thesis/code/tools/mulval_demo/test/svd-scenario]graph_gen.sh input.P -g "execCode(dbserver,)"
<1>|--execCode(dbserver,root)
(2) RULE 2 : remote exploit of a server program
<3>|--netAccess(dbserver,tcp,80)
(4) RULE 5 : multi-hop access
[5]-hacl(dbserver,dbserver,tcp,80)
execCode(dbserver,root)==><1>
(6) RULE 5 : multi-hop access
[7]-hacl(workstation_cto,dbserver,tcp,80)
<8>|--execCode(workstat
(9) RULE 2 : remote
<10>|--netAccess
(11) RULE 5 :
[12]-hacl(v
<13>|--exec
(14) RUL
<15>|
(1
(1
[21]-
[22]-
(23) RULE 5 :
[24]-hacl(v
execCode(wr
[25]-networkServi
[26]-vulExists(wr
[27]-networkServiceInfo(dbser
[28]-vulExists(dbserver,_h290
(29) RULE 2 : remote exploit of
netAccess(dbserver,tcp,80)==
[30]-networkServiceInfo(dbser
[31]-vulExists(dbserver,_h300
(32) RULE 2 : remote exploit of
netAccess(dbserver,tcp,80)==
[33]-networkServiceInfo(dbser
[34]-vulExists(dbserver,_h320
bob@cronus[~/data/studium/thesis/cc
  
```



Agenda

- × Vulnerabilities & Attack Graphs
- × Problem Statement
- × Vulnerability Information Representation
- × Vulnerability Information Transformation
- × Proof of Concept
- × **Conclusion**



- × Common data structure for vulnerability information representation
- × Analysis of vulnerability databases
- × Automation of vulnerability database transformation
- × Automatic transformation of textual vulnerability descriptions

Future Work

- ✘ Implement adapters for other Attack Graph tools
- ✘ Research the new possibilities of AG generation based on extended information
- ✘ Apply data structure to other information types
- ✘ Implement adapters to auto-generate NVD/OVAL/CVSS entries
- ✘ Research semantics of vulnerability descriptions

Questions



Master's Thesis Final Presentation
February 24th, 2009

Robert Schuppenies

*Automatic Vulnerability Extraction
for Attack Graphs*