Security Governance
Master of Science in Cyber Security

AA 2023/2024

ATTACK GRAPH

Attack Graph

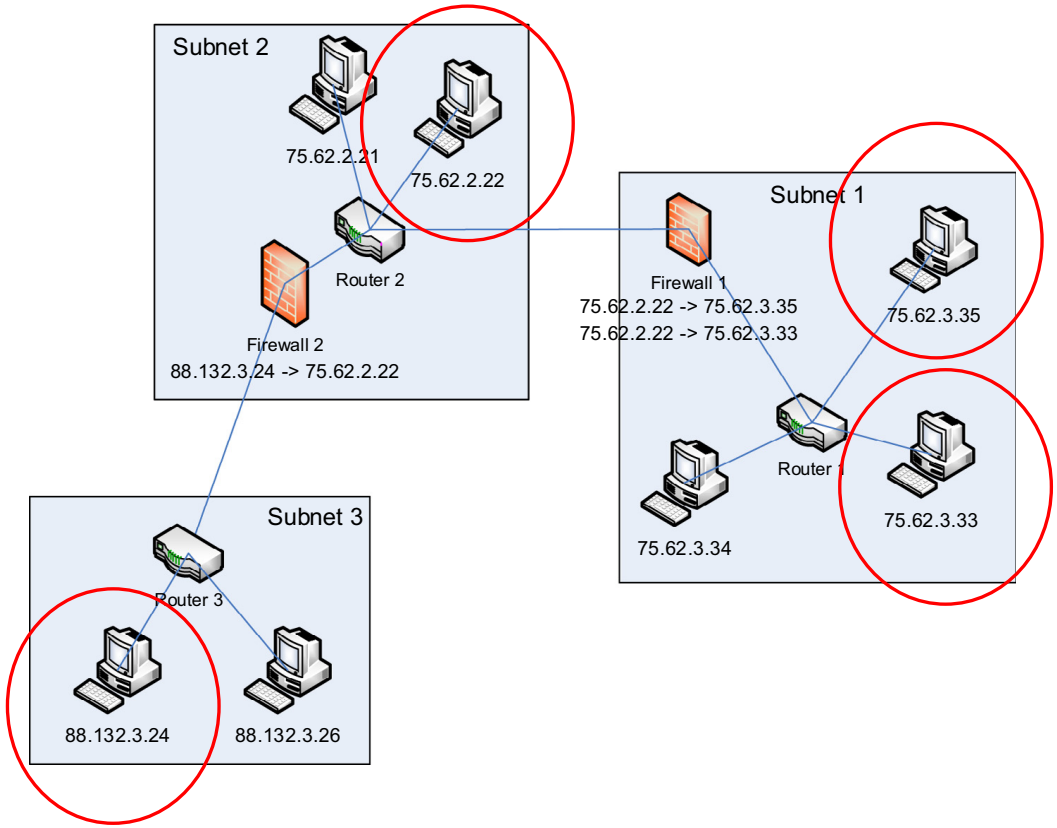
An attack graph represents possible ways via which a potential attacker can intrude into the target network by exploiting a series of vulnerabilities on various network hosts and gaining certain privileges at each step

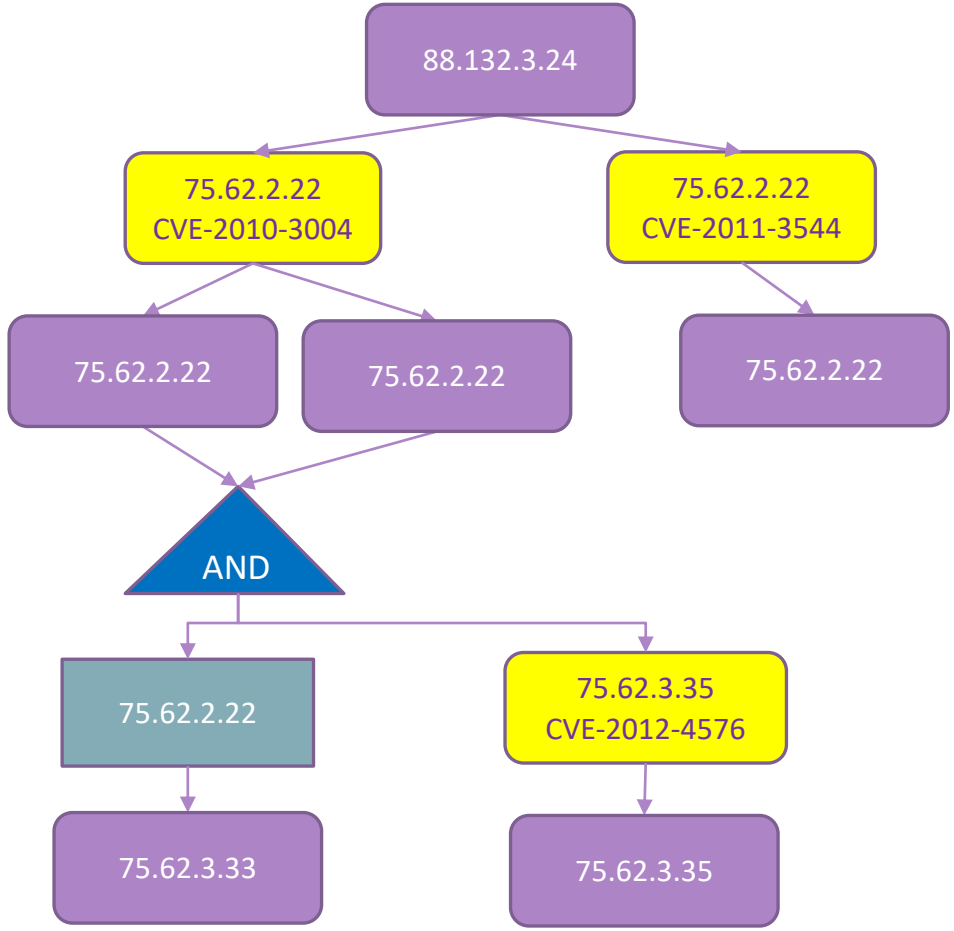
In a typical attack graph

- nodes represent the privileges gained by the attacker on the network hosts
- edges represent the software vulnerability exploits employed by the attacker to gain these privileges

The computation of an attack graph requires the computation of the reachability conditions among the network hosts by considering all network protocol layers, modelling attacks and attack paths, and devising an efficient method to compute possibly huge number of attack paths

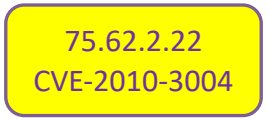
Attack graph example



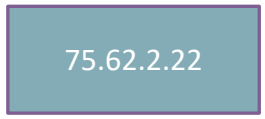


Example of Attack Graph format

Privilege nodes indicating attacker privileges that can be obtained on the software installed on the network hosts with specific IP addresses



Nodes indicating vulnerability exploits that can be applied by an attacker on the installed software



Nodes indicating information source usages that can be applied by an attacker

Conjunction (AND) nodes combining more than one privilege required by an attacker to successfully exploit a vulnerability or use an information source

Example of Attack Graph format

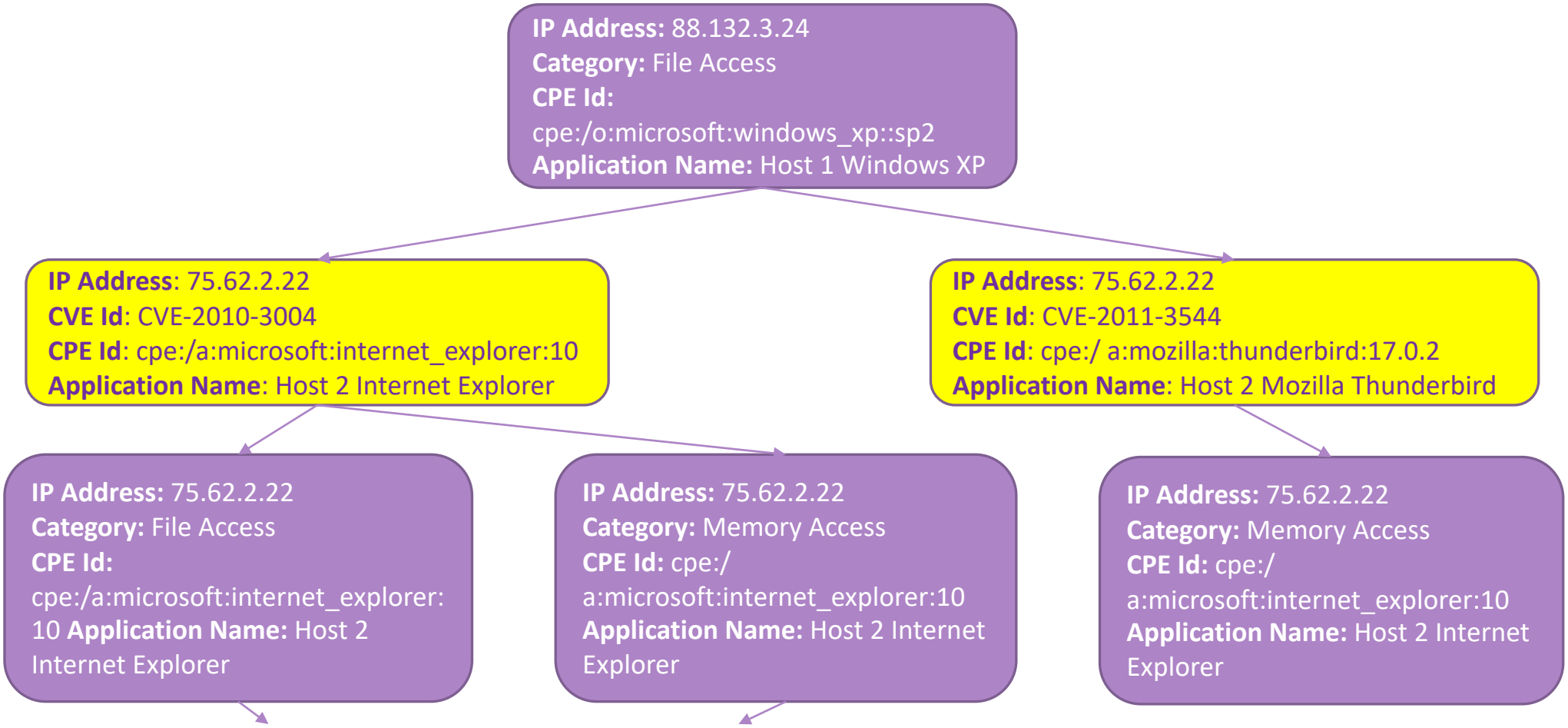
All nodes (except conjunction node) contains the following information

- **IP Address** field shows the IP address related to the corresponding attack graph node
- **CPE Id** fields indicates the unique product identifier of an installed software in Common Product Enumeration (CPE) database
- **Application Name** field indicates user-defined name of the installed software

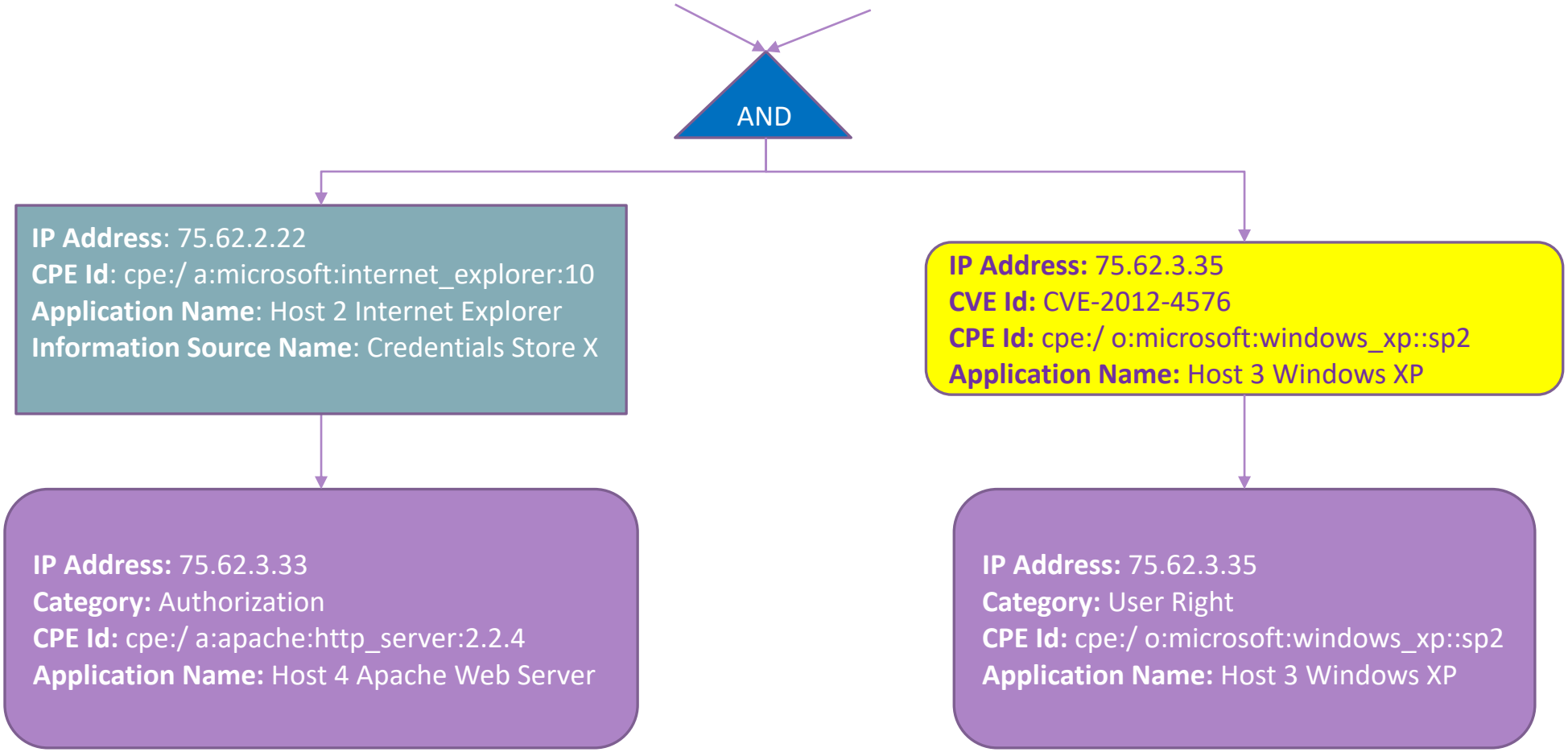
Additionally

- In Privilege nodes there is a **Category** field indicating the software system right related to the privilege
- In Vulnerability nodes there is a **CVE Id** field representing the unique identifier for the exploited vulnerability defined by Common Vulnerability Exposure (CVE) database
- In Information Source Usage node there is **a Information Source Name** field showing the name of the used information source

Attack graph example



Attack graph example



Basic problems in attack graph generation

There are 4 main problems in the attack graph generation process:

1. reachability analysis
2. attack template determination
3. attack graph structure determination
4. attack graph core building mechanism

Reachability Analysis

The attack graph core building process utilizes network reachability data to check for the target hosts' reachability for an attacker from the current attacking host

Network reachability data are mostly represented as a reachability matrix

- columns and rows include the hosts in the network
- each entry represents the reachability condition between the two hosts on the corresponding row and column
- Each entry in the reachability matrix may be a boolean or indicate the protocols used between the two corresponding hosts to reach each other
- can be used to represent any type of connection among the hosts; physical, network, transport or application-level connection

Example of Reachability Matrix

		88.132.3.x		75.62.2.x		75.62.3.x		
		24	26	21	22	33	34	35
88.132.3.x	24	1	1		1			
	26	1	1					
75.62.2.x	21			1	1			
	22			1	1	1		1
75.62.3.x	33					1	1	1
	34					1	1	1
	35					1	1	1

Gathering Information for Reachability Matrix computation

The configuration information can include the following:

- the topology of the target network,
- the applications (software or hardware installations) on the network hosts,
- the employed filtering and access control rules,
- the intrusion detection/prevention system configurations and
- trust relations among the network hosts.

The more network configuration information is obtained, the more accurate the attack graphs will be

Attack template determination

An attack graph contains the privileges gained on the target network hosts by an attacker

These privileges are related to the possible vulnerability exploits.

The relationships between a set of privileges and a vulnerability exploit are determined by using an attack template

An attack template specifies the conditions required by an attacker to perform a set of specific attacks successfully. It also describes the conditions gained by an attacker, after the corresponding attacks are successfully performed. The attack templates created collectively form the attack model.

Attack template determination

The determination of what can be a privilege should be performed in the attack template design process.

- Example privileges include access levels (e.g., user, root), file access/ modification rights and memory access/modification rights.

One can design privileges based on the type of applications that can be installed on a host computer,

- e.g., file modification rights on browser cookies, system or web server files.

When the detail level of the determined privileges increases, the precision of the resulting chains of the vulnerability exploits in the generated attack graphs increases, but the time and space requirements of the attack graph core building process also grow.

Attack graph structure determination

The space complexity of a full attack graph may easily reach an exponential order on the number of hosts in the target network

A specific attack graph structure represents an instance of the attack graph model

Generally, privileges and vulnerability exploits are used as basic attack graph elements.

However other kinds of graph elements may be introduced to reduce the space complexity of a full attack graph and the time complexity of building attack graphs

Attack graph core building mechanism

In both partial and full attack graph generation, the initial privileges possessed by the attacker and the target privileges for the attacker are given as inputs for attack paths determination.

For full attack graph generation, each possible attack path from the initial to the target privileges is found.

The full attack graph generation process can be formulated as a general graph traversal problem, since it has to find all the attack paths.

In essence, most of the attack graph generation algorithms proposed in the literature use some form of searching algorithm to find the corresponding nodes in the resulting attack graph

Issues

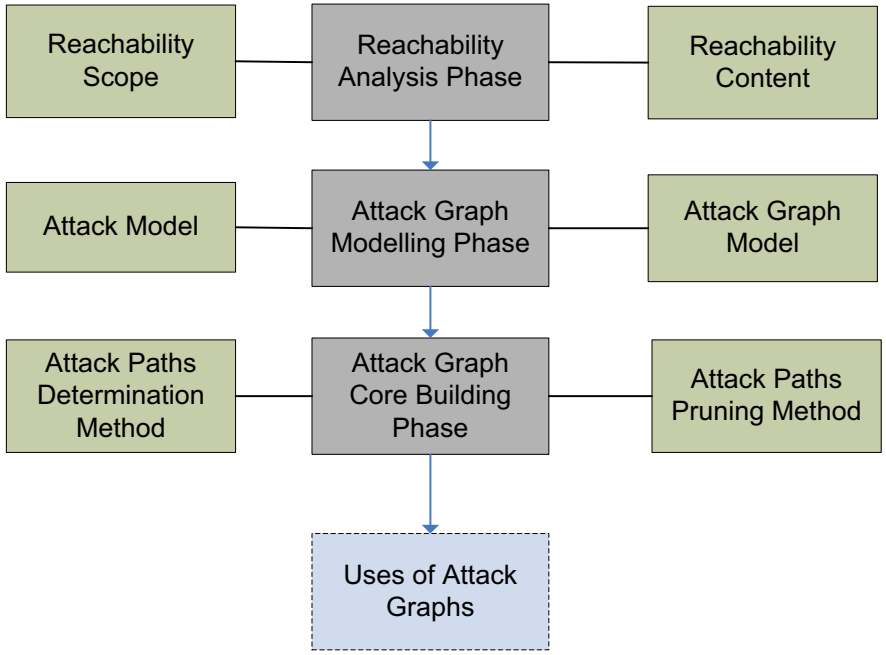
SCALABILITY

Countermeasures

- *monotonicity* assumption is introduced by Ammann et al. (2002). This assumption states that the attacker can not relinquish a privilege that she has already owned. Namely, an attack can not negate any of the privileges obtained by the attacker so far.
- *pruning the attack paths* based on the depth and/or the transitive likelihood of success value of the traversed attack path.
- In partial attack graph generation, only a number of criti cal (shortest) attack paths can be found.
- Cycle-free attack graph

Attack graph generation process taxonomy

The activities performed during the whole attack graph generation process can be classified into three high-level phases



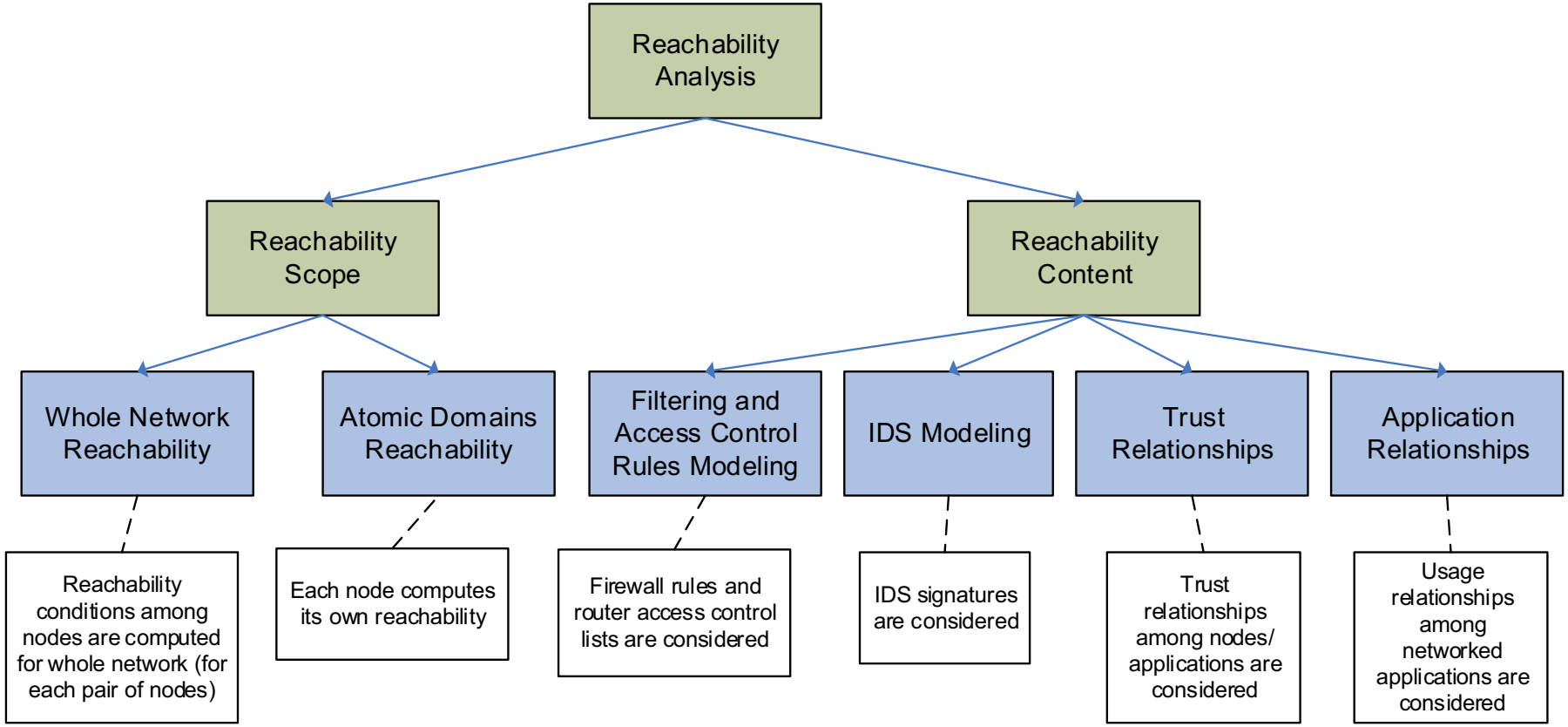
Reachability analysis phase

The reachability analysis phase mainly investigates the network reachability conditions within the target network, which, in a simplistic viewpoint, determine whether two given hosts can access each other

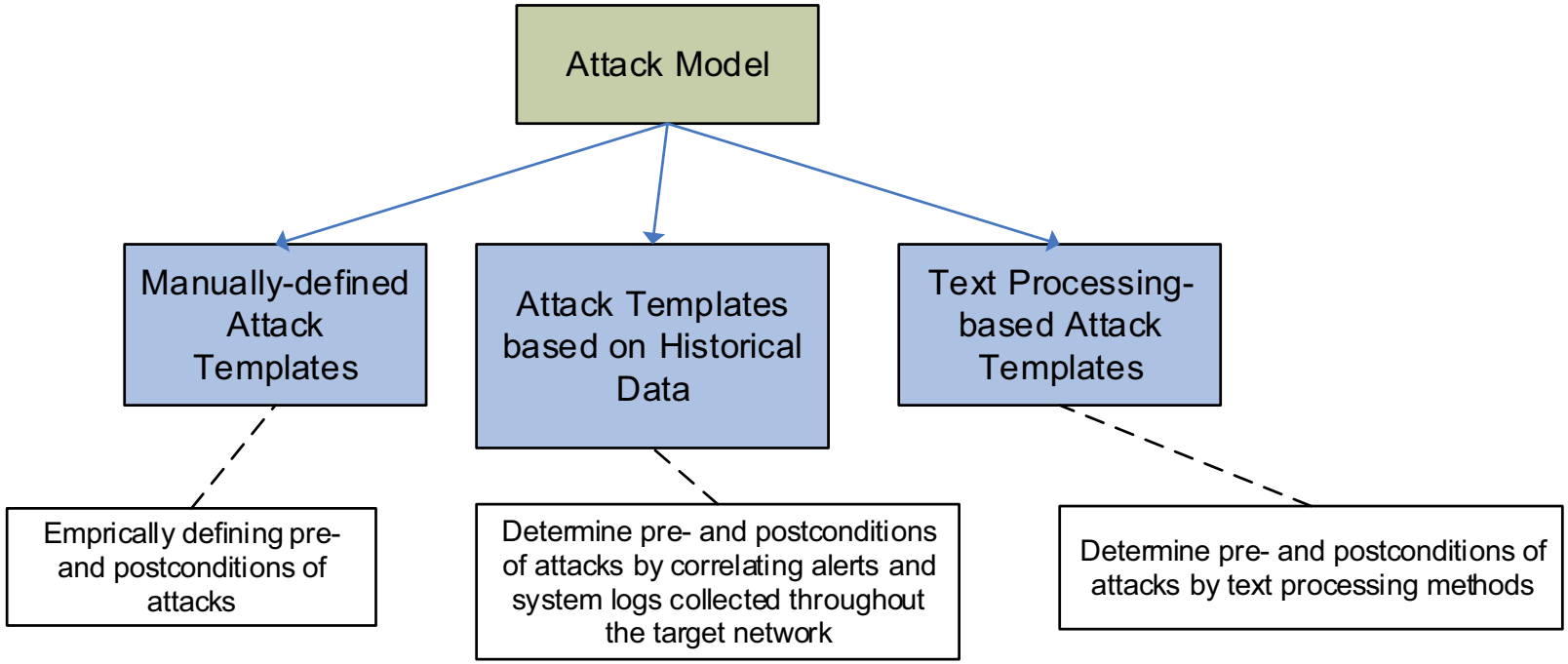
Two main classification criteria for the reachability information are reachability scope and reachability content.

- Reachability scope determines the scope of the network hosts among which the reachability conditions are computed before the attack graph core building process.
- Reachability content determines the network security objects (entities) that are accounted for in the computation of the reachability information.

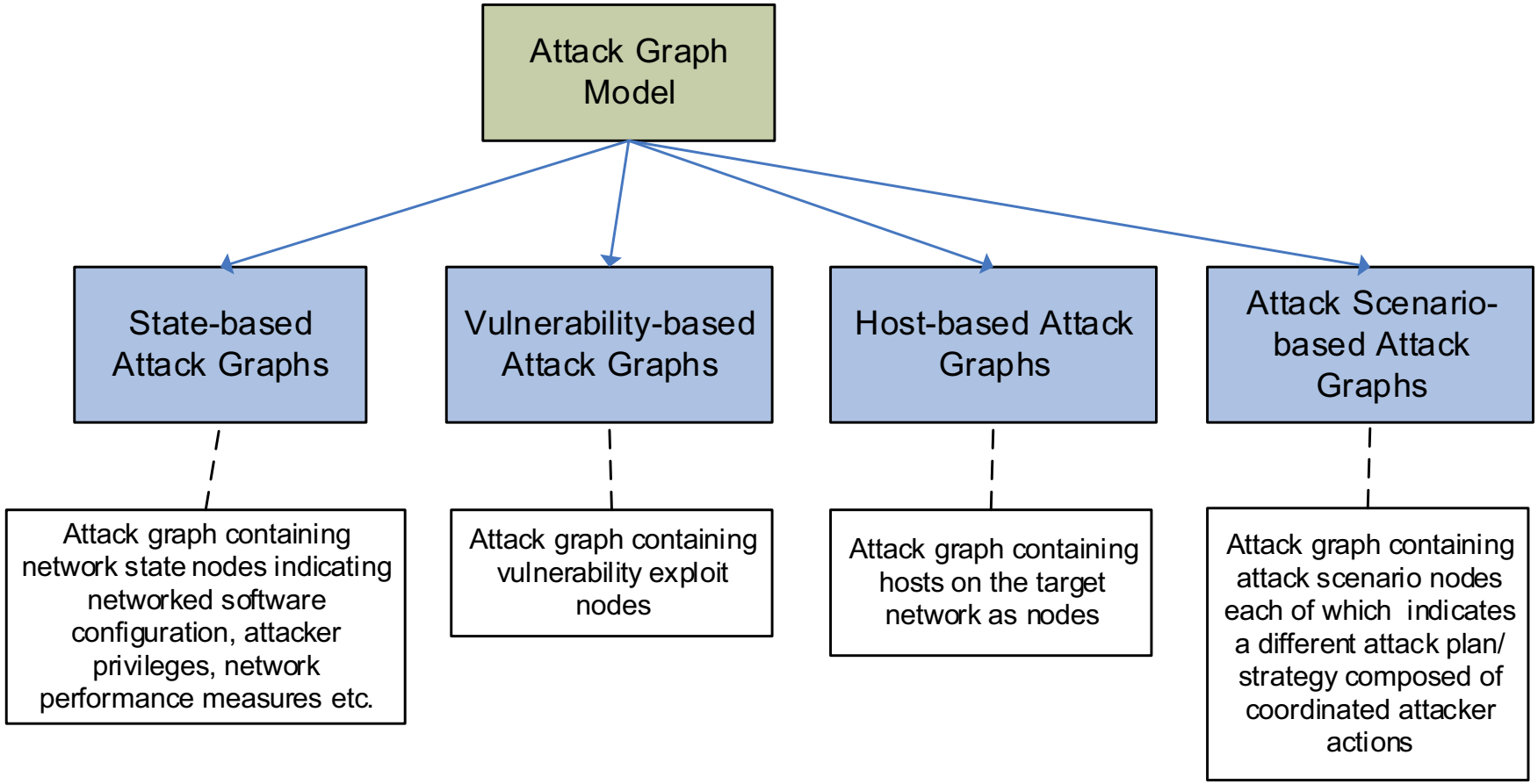
Reachability analysis phase



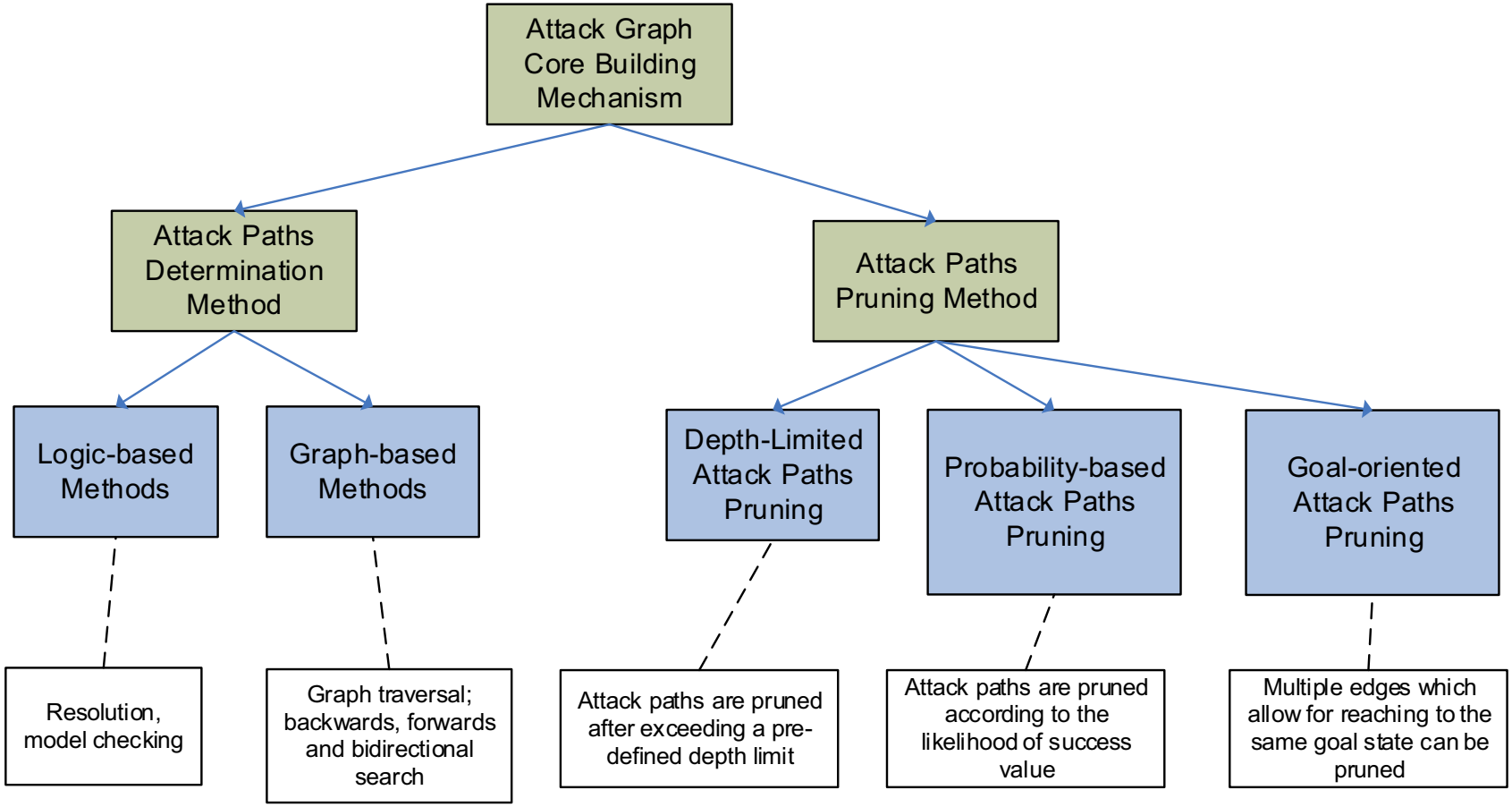
Attack graph modelling phase



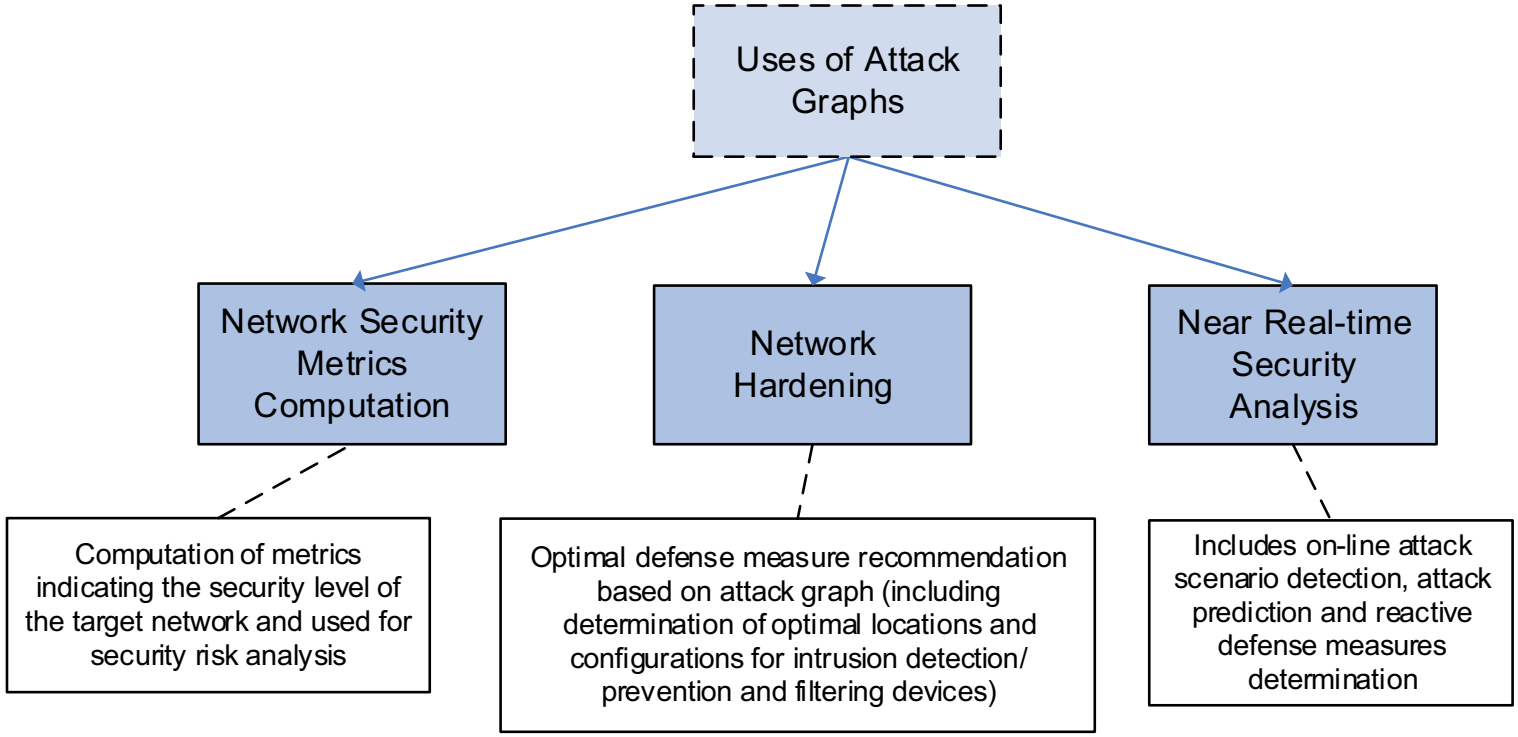
Attack Graph Model



Attack Graph Computation



Attack Graph Usage



NetSPA

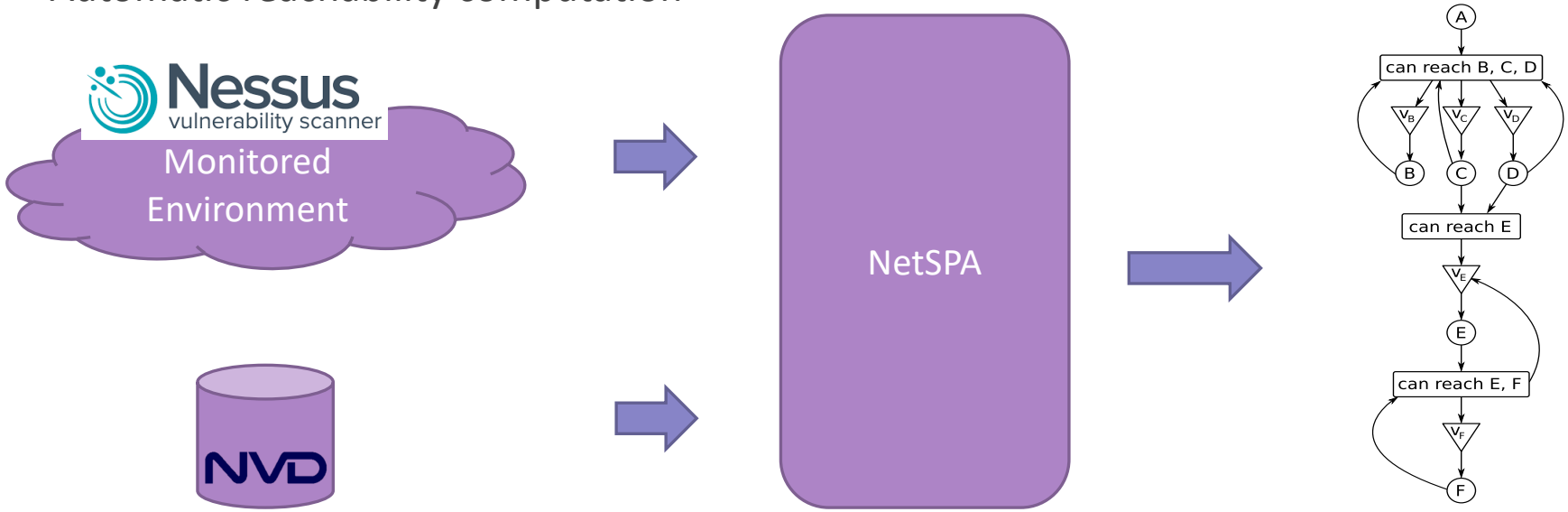
PRACTICAL ATTACK GRAPH GENERATION FOR NETWORK DEFENSE

NetSPA Approach

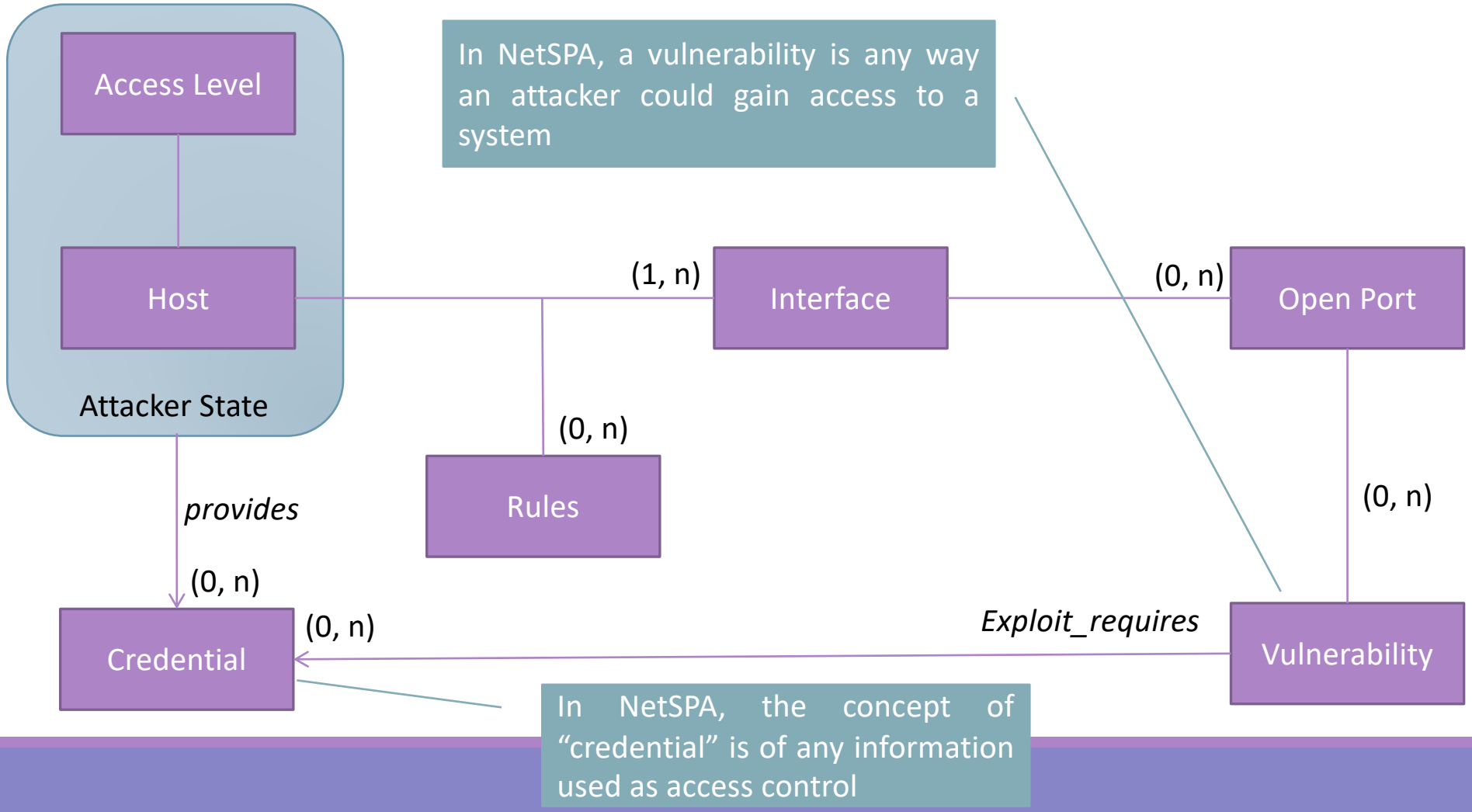
NetSPA is an attack graph generation system

Features

- Multiple-prerequisite (MP) graph
- Interface with common data sources
- Automatic reachability computation



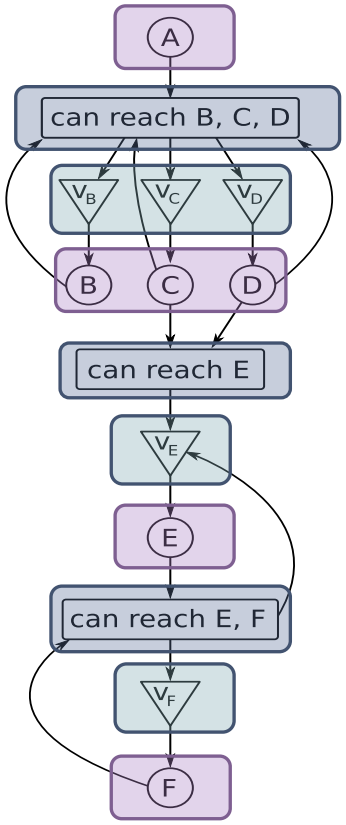
NetSPA Data


Access Level
Host
Attacker State
In NetSPA, a vulnerability is any way an attacker could gain access to a system
(1, n)
Interface
(0, n)
Open Port
(0, n)
Rules
(0, n)
provides
(0, n)
Credential
(0, n)
Exploit_requires
Vulnerability
In NetSPA, the concept of "credential" is of any information used as access control

NetSPA Multi Prerequisite (MP) Graph

The MP graph uses the following three node types:

- ***State nodes*** represent an attacker's level of access on a particular host.
  - Outbound edges from state nodes point to the prerequisites they are able to provide to an attacker.
- ***Prerequisite nodes*** represent either a reachability group or a credential.
  - Outbound edges from prerequisite nodes point to the vulnerability instances that require the prerequisite for successful exploitation.
- ***Vulnerability instance nodes*** represent a particular vulnerability on a specific port.
  - Outbound edges from vulnerability instance nodes point to the single state that the attacker can reach by exploiting the vulnerability.



NetSPA Graph Construction

The graph is built using a breadth-first technique.

No node is explored more than once, and a node only appears on the graph if the attacker can successfully obtain it.

```
1   BFSQueue starts with the root node(s),
            representing the attacker's
            starting STATE(s)
2   while( BFSQueue is nonempty )
3    CurNode = BFSQueue.dequeue()
4    DestSet = all nodes that can be
               reached from CurNode
5    foreach node DestNode in DestSet
6     add an edge from CurNode to DestNode
7     if DestNode is brand-new,
8      BFSQueue.enqueue( DestNode)
```

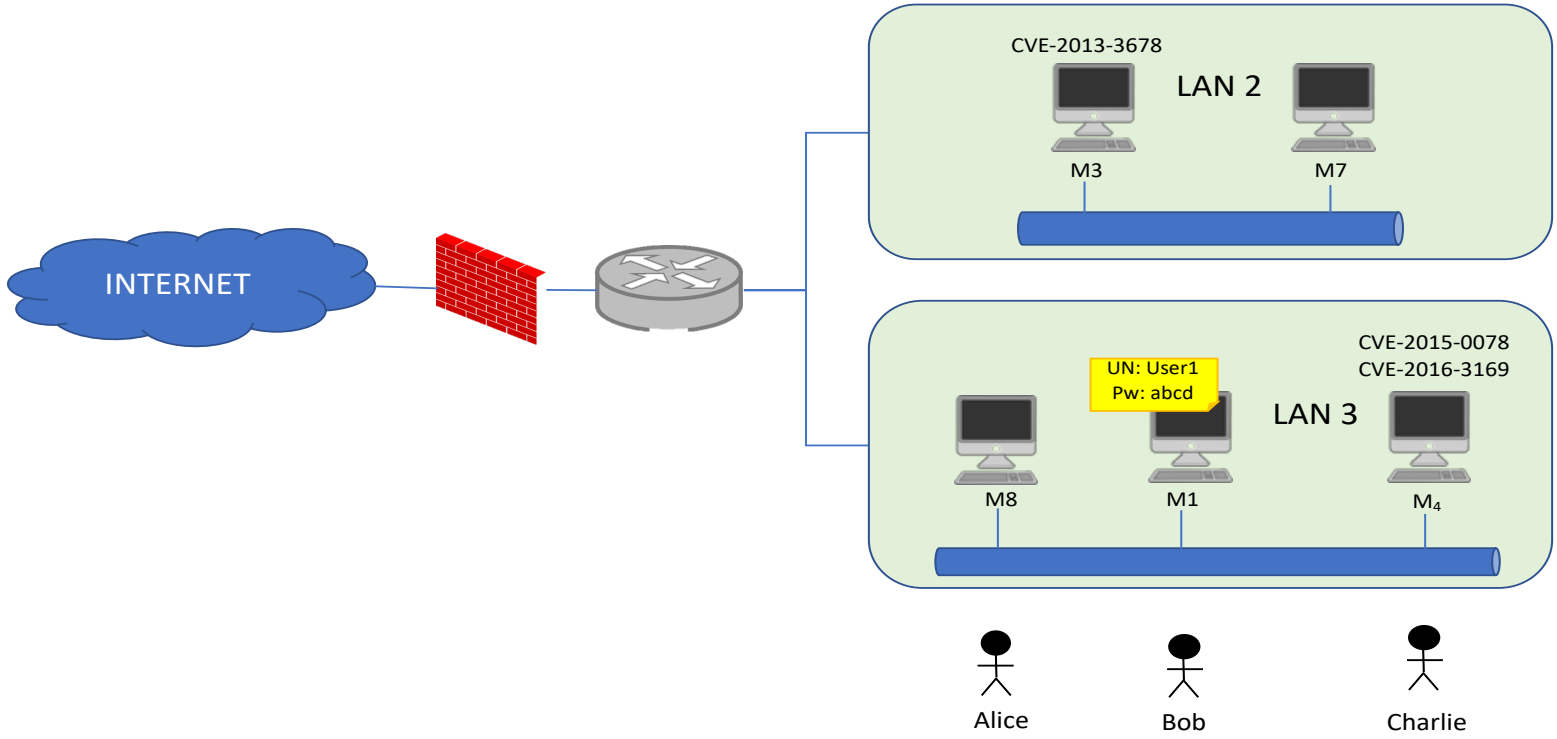
Figure 4. Pseudocode for Main Loop

The result of this line depends on the type of node under Analysis

1. CurNode is a state
2. CurNode is a prerequisite that is a reachability group
3. CurNode is a *prerequisite* that is a *credential*
4. CurNode is a *vulnerability instance*

Exercise

Let’s consider the following network fragment



Exercise

Compute the attack path having as target M3 considering that:

- M3 is reachable only from machines in LAN2 and from M4
- Alice has user access to M8 and she has the bad habits of sharing her credentials with all her colleagues.
- Bob has user access to M1 and he has poor memory. He leaves all his credentials written around on post it.
- Charlie has access to M4 and he is very careful in managing his credentials. However, he tends sometimes to leave his machine logged in while he is out for a coffee.

References

[1] Kerem Kaynar "*A taxonomy for attack graph generation and usage in network security*", Journal of Information Security and Applications 29 (2016) 27-56

[2] Kyle Ingols, Richard Lippmann, and Keith Piwowarski, *Practical Attack Graph Generation for Network Defense*. In Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC '06). IEEE Computer Society, USA, 121–130.