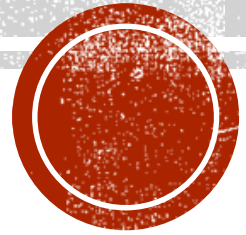
SOFTWARE DEFINED VIRTUAL NETWORKING SECURITY

CHAPTER 8 ATTACK REPRESENTATION

Dijiang Huang, Ankur Chowdhary, and Sandeep Pisharody

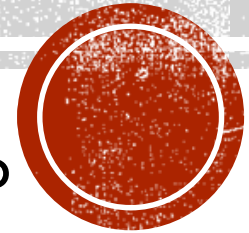


OUTLINE

- Foundation of Cyber Quantification Metrics (CQM)
- Attack Graph
- Attack Tree
- Attack Countermeasure Tree
- Other Attack Representation Models
- Limitations of Attack Representation Methods

FOUNDATION OF CYBER QUANTIFICATION METRICS

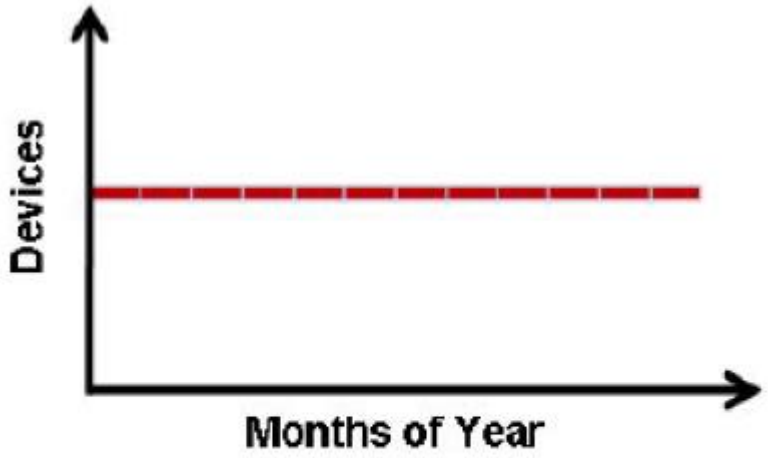
Cybersecurity Metrics, CVSS, CVSS Use Case, Attack Scenario Analysis, Quantitative and Qualitative Metrics



ENTERPRISE NETWORK SECURITY FEATURES

- Networks are getting large and complex.
- New software vulnerabilities are constantly discovered.
- Network Security management is a challenging task.
- Even a small network can have multiple threat vectors.
- System administrators operate by instinct and learned experience.
- There is no objective way of measuring the security risk in a network.
- "If I change the network configuration setting, will my network become more secure or less secure?

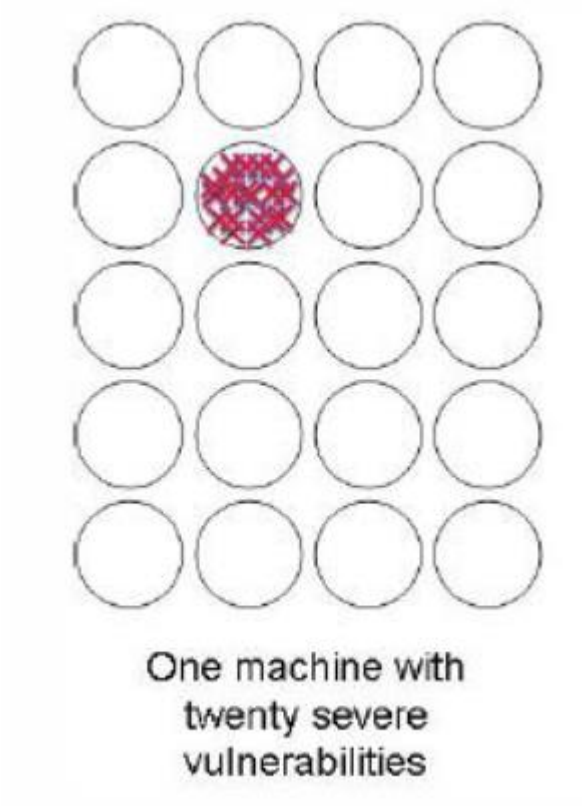
WOULD YOU RATHER



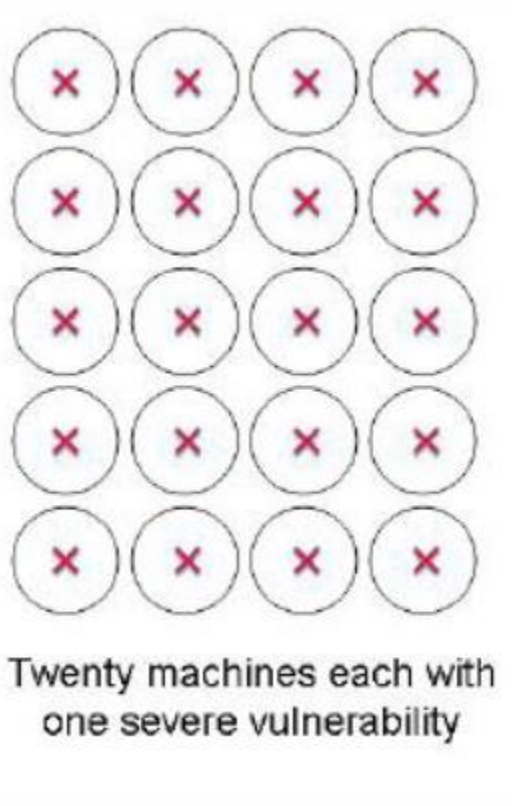
One machine that is vulnerable the entire year

Twelve machines that are vulnerable for one month each

WOULD YOU RATHER



One machine with twenty severe vulnerabilities



Twenty machines each with one severe vulnerability

CHALLENGES IN SECURITY METRICS

- How to measure the security strength of a given network?
- How secure is a server in a given network configuration?
- How much security does the new configuration provide?
- How can I plan on security investment so that it certain level of security is guaranteed?
- We need cybersecurity modeling and analysis tools.

CYBERSECURITY METRICS

- Metrics for Individual Vulnerability exists

1. Impact, Exploitability, Temporal, Environmental, etc.
2. E.g. The Common Vulnerability Scoring System (CVSS) v2/v3.

- Counting the number of vulnerabilities is not enough.

1. Vulnerabilities have different importance.
2. Vulnerability scoring is difficult based on – a) Context of the application b) Configuration of the application.

- How to compose the vulnerabilities for the overall security of the network system.

CYBERSECURITY METRICS

- Should be fast (real-time) and automatic
- Should evaluate the cybersecurity risk from both

1. Qualitative Perspective
2. Quantitative Perspective

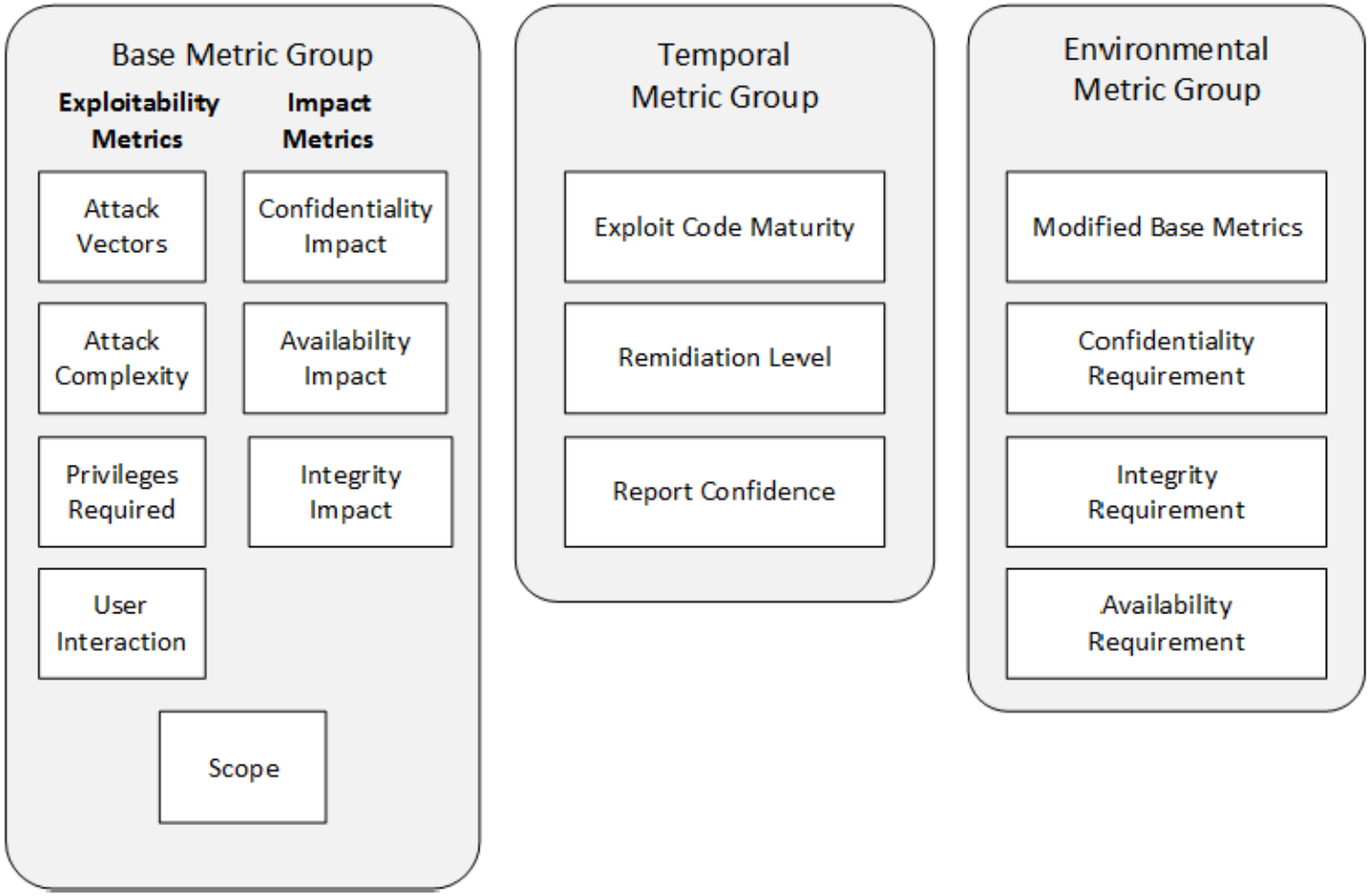
CVSS

- Stands for Common Vulnerability Scoring System (CVSS).
- An open framework for communicating characteristics and impacts of IT vulnerabilities.
- Consists three metric groups: Base, Temporal, and Environmental.

CVSS

- Base metric : constant over time and with user environment.
- Temporal metric : change over time but constant with user environment.
- Environmental metric : unique to user environment.

CVSS METRICS



EXPLOITABILITY METRICS

1. Attack Vector
2. Access Complexity
3. User Interaction

ATTACK VECTOR

- This metric reflects the context by which vulnerability exploitation is possible.
- The metric value will be larger the more remote (logically or physically) an attacker can be in order to exploit the vulnerability.

1. Network (N)
2. Adjacent
3. Local (L)
4. Physical (P)

ACCESS COMPLEXITY

- This metric describes the conditions beyond the control of attacker.
- Such conditions include presence of certain network configuration, or computational exceptions, etc.

1. Low (L)
2. High (H)

PRIVILEGES REQUIRED

- Privileges attacker must possess before successfully exploiting the vulnerability.

1. None (N)
2. Low (L)
3. High (H)

USER INTERACTION

- Requirement for user participation, whether the vulnerability can be exploited by the attacker standalone or user-initiated process must participate in some manner.

1. None (N)
2. Required (R)

SCOPE

- Checks if the vulnerability in software component impacts the resources beyond its means.
- Scope refers to collection of privileges in (application Operating System, Sandbox environment).

1. Unchanged (U): Exploited vulnerability can affect resources managed by the same authority.
2. Changed (C): An exploited vulnerability can affect the resources beyond the authorization privileges intended by the vulnerable component.

IMPACT METRICS

1. Confidentiality Impact
2. Integrity Impact
3. Availability Impact

CONFIDENTIALITY IMPACT (C)

- This metric measures the impact on confidentiality of the information resources managed by the software component due to successfully exploited vulnerability

1. High (H)
2. Low (L)
3. None (N)

INTEGRITY IMPACT (I)

- Measures the impact on trustworthiness or veracity of information because of successfully exploited vulnerability.

1. High (H)
2. Low (L)
3. None (N)

AVAILABILITY IMPACT (A)

The confidentiality and integrity metrics measure the impact on data (files, information), this metric measures the loss of availability of impacted component such as network service, web or database application.

1. High (H)
2. Low (L)
3. None (N)

BASE SCORE CALCULATION

- If (Impact sub score <= 0) 0 else,

Scope Unchanged: Base Score = (Minimum [(Impact + Exploitability), 10])
Scope Changed: Base Score = (Minimum [1.08 × (Impact + Exploitability), 10])

- Impact sub score (ISC) is defined as,

Scope Unchanged: $6.42 \times ISC_{Base}$
Scope Changed: $7.52 \times [ISC_{Base} - 0.029] - 3.25 \times [ISC_{Base} - 0.02]$

- And the Exploitability sub score is,

$8.22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction$

CVSS USE CASE: SHELLSHOCK (CVE-2014-6271)

- Base Score value range is between [0,10].
- GNU Bourne Again Shell (Bash) vulnerability also known as Shellshock has been indexed as CVE-2014-6271 in CVSSv3.0.
- This allowed remote attackers to execute arbitrary code via crafted environment variables.
- The attacker can target the Apache HTTPD Server running dynamic content CGI modules.

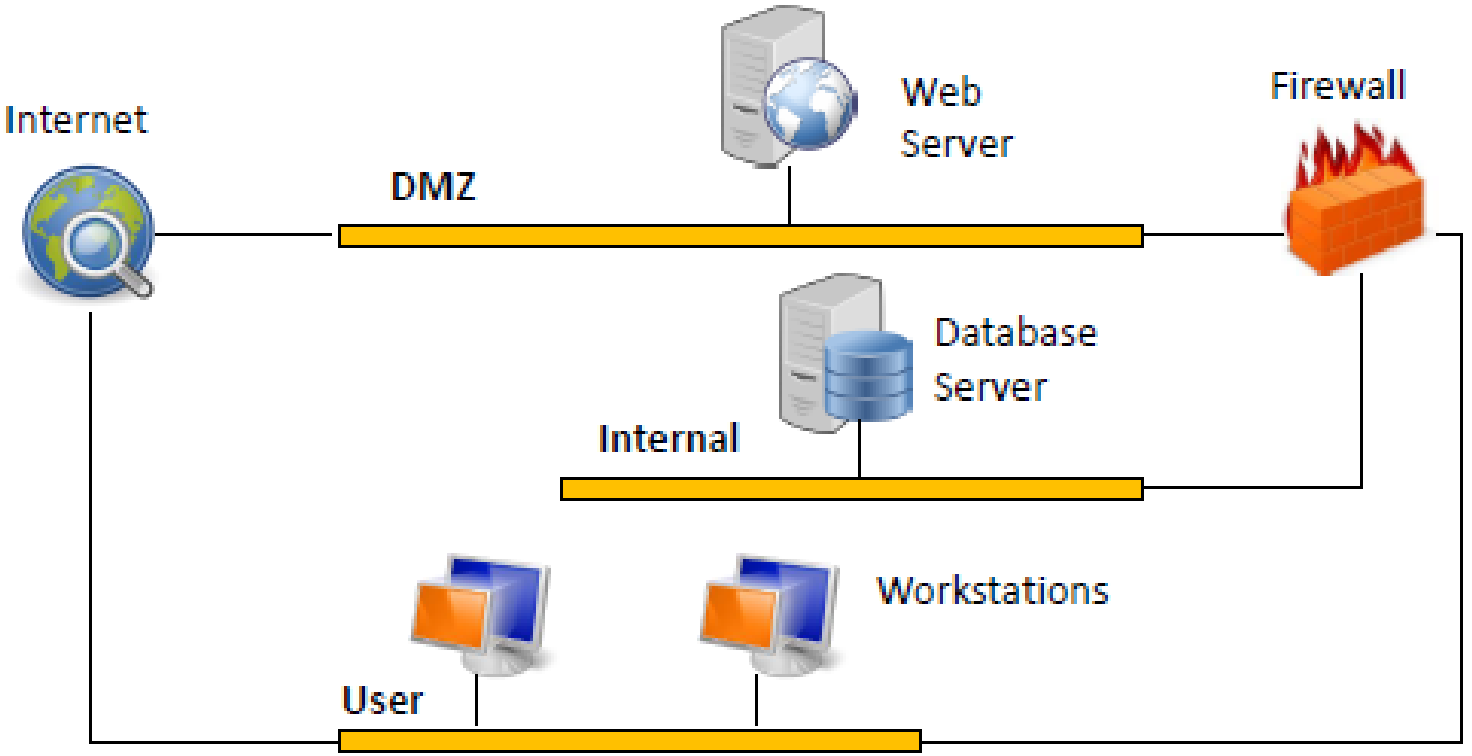
CVSS USE CASE: SHELLSHOCK (CVE-2014-6271)

CVSS Metric	Value	Comment
Attack Vector	Network	Web Server Attack.
Access Complexity	Low	Attacker needs to access service using bash shell interpreter.
Privileges	None	CGI in web server requires no privilege.
User Interaction	None	No user interaction required to launch successful attack.
Scope	Unchanged	GNU bash shell is vulnerable component, which can be used without any change in the scope.

CVSS USE CASE: SHELLSHOCK (CVE-2014-6271)

CVSS Metric	Value	Comment
Confidentiality Impact (C)	High	Attacker can take complete command and control (C&C) of the affected system.
Integrity Impact (I)	High	Attacker can take complete command and control (C&C) of the affected system.
Availability Impact (A)	High	Attacker can take complete command and control (C&C) of the affected system.

ATTACK SCENARIO ANALYSIS



ATTACK METRICS

- What is the value of the host that has the vulnerabilities?
- What is the threat model?
- Consider rapidly scanning worm inside the network attacking all severe vulnerabilities.

ATTACK MODELING

- Attack Model 1: Attacker tries a single exploit at random.
- $P_{\{device-compromise\}} = \frac{P_i}{N}$

ATTACK MODELING

- Attack Model 2: Attacker tries single best exploit
- $P_{\{device-compromise\}} = \max P_i$

ATTACK MODELING

- Attack Model 3: Attacker tries all exploits
- $P_{\{device-compromise\}} = 1 - \Pi(1 - P_i)$

COMPROMISE PROBABILITY

Combination formula for overall probability of failure	Derived from an attack model where the attacker tries	Adding a vulnerability does not improve device security	Using a single vulnerability results in the compromise probability for that vulnerability $\{P\} \rightarrow P$	Adding a vulnerability worsens device security unless the compromise probability is already 1.0	Worsening a vulnerability does not improve device security
$1 - \prod(1 - P_i)$	✓ all exploits	✓	✓	✓	✓
$\sum P_i/N$	✓ single exploit at random	× [1.0] vs. [1.0, 0.1]	✓	× [0.5] vs. [0.5, 0.5]	✓
$\max(P_i)$	✓ single best exploit	✓	✓	× [0.5] vs. [0.5, 0.5]	✓

QUALITATIVE AND QUANTITATIVE METRICS

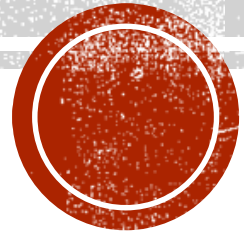
- Component Metric can be generated based on numerical value of CVSS metric. This metric represents the conditional probability of attack success, e.g.,
- p(c1):{Network Access}, p(c2): {Host Compromised Probability}.
- What is probability of host compromise given network access;

$$p(c2|c1)$$

QUALITATIVE AND QUANTITATIVE METRICS

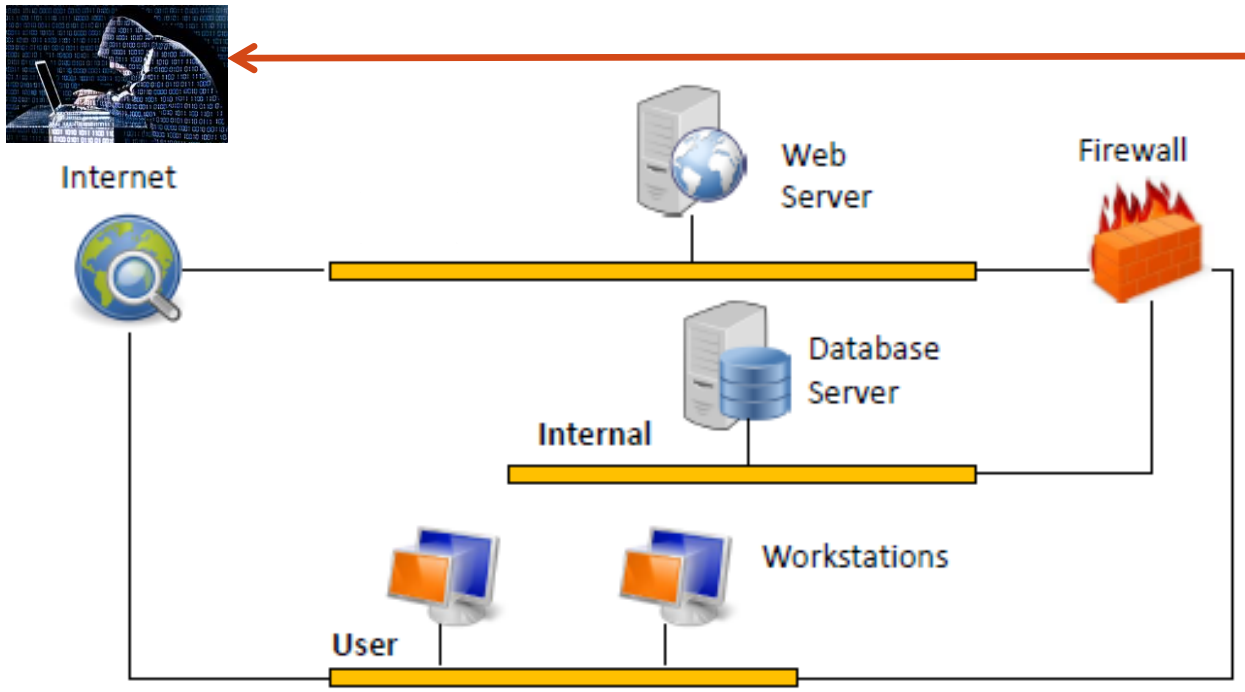
- Cumulative Metric is used to represent the aggregate probabilities over the attack graph.
- E.g., If a dedicated attacker tries exploiting all possible attack paths.
- Cumulative Metric denotes the probability of attack success along at-least one path.

ATTACK GRAPH



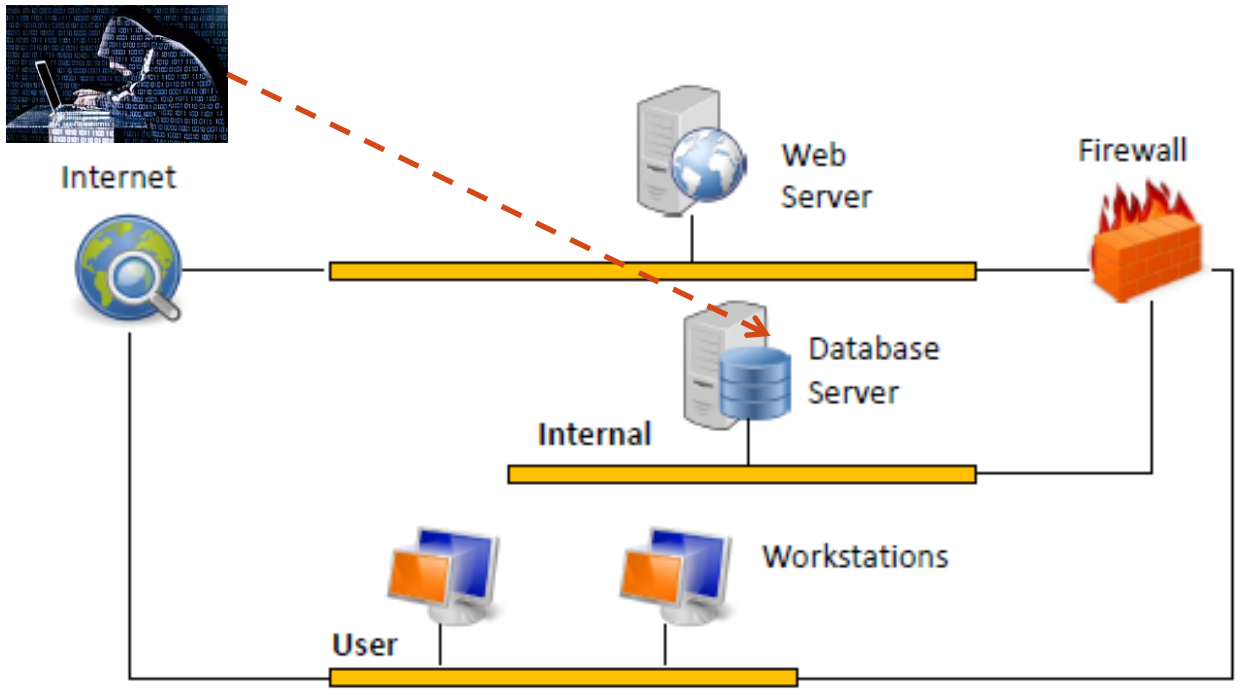
Attack Graph Basics, Probabilistic Attack Graphs, Risk Mitigation using Probabilistic Metrics, Ranking Attack Graphs

MOTIVATING EXAMPLE



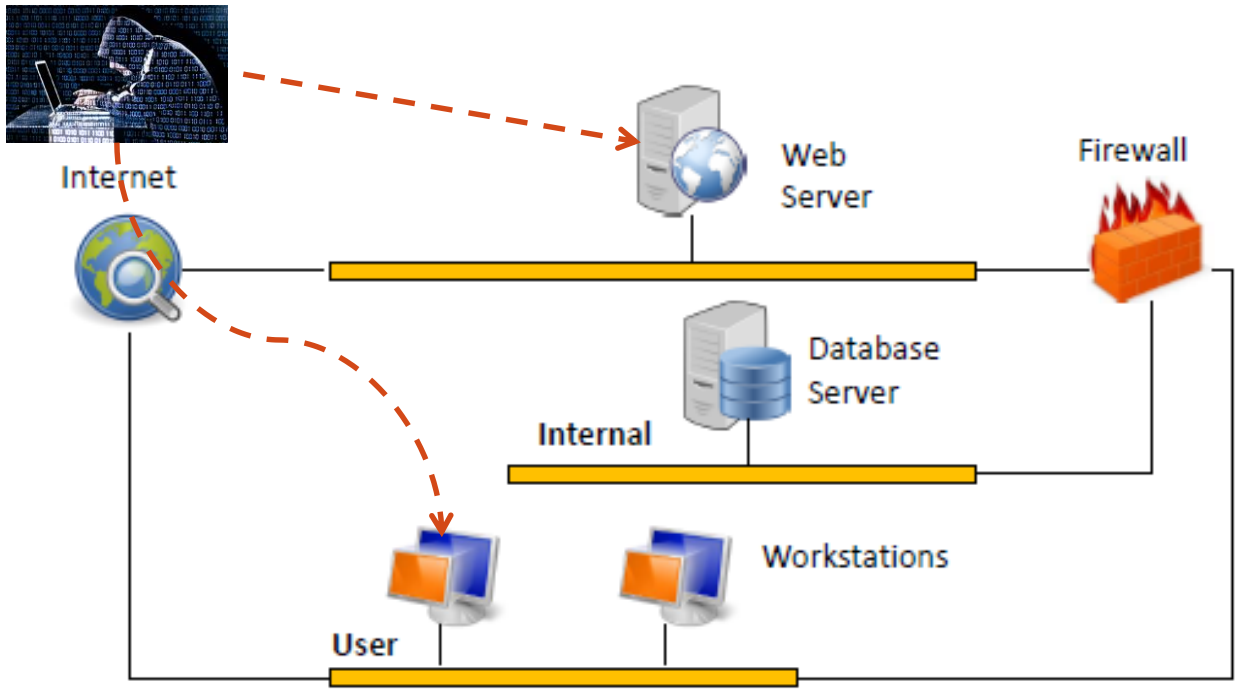
- There is an Attacker located on the internet.
- Attacker can access publically available services of a company.
- There are known vulnerabilities on the running host OS and services.

MOTIVATING EXAMPLE



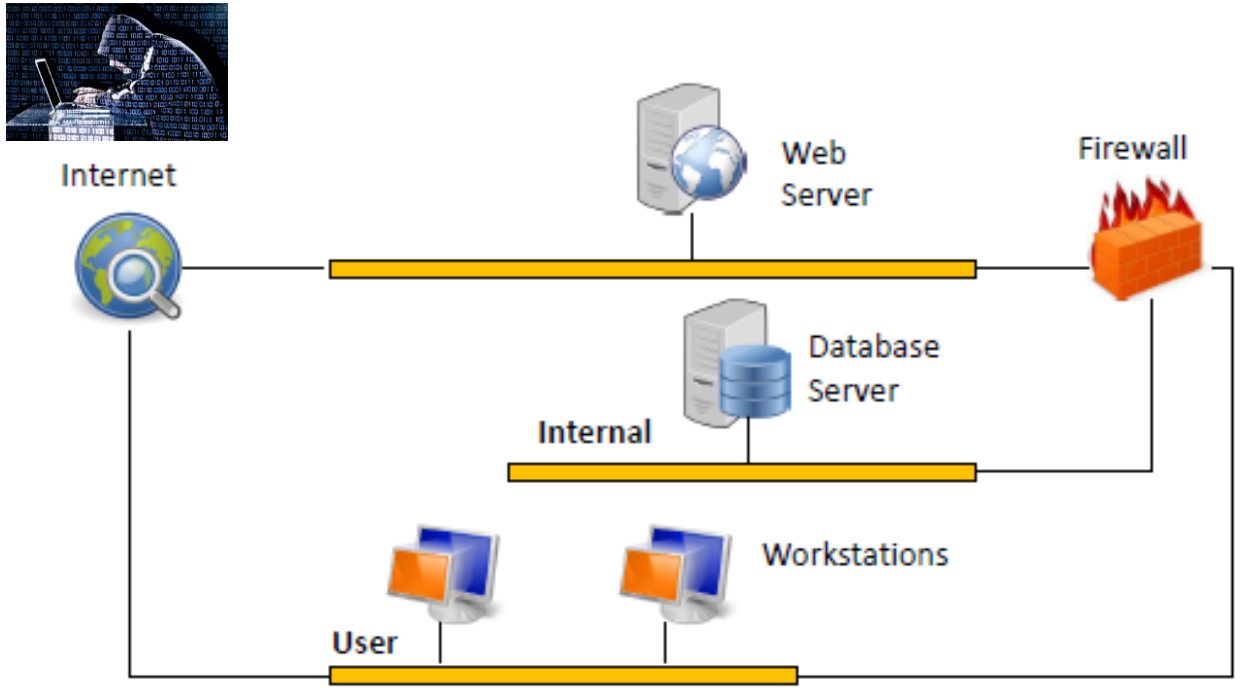
- Goal of the attacker is to exploit the services present behind the Firewall containing sensitive data for the company.
- Web server is publically accessible.
- Firewall on the way.

MOTIVATING EXAMPLE



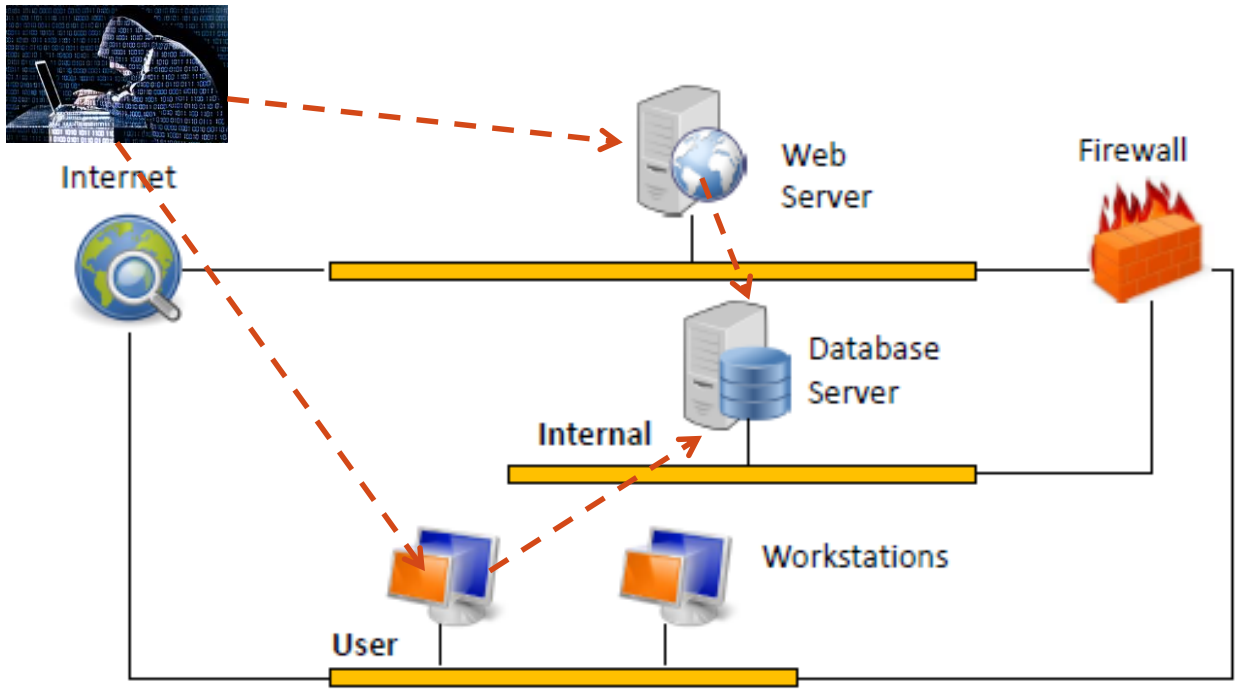
- Step 1a: Attacker can try remote code execution and compromise Web Server.
- Step 1b: Attacker can exploit vulnerabilities on user login system and gain access to workstations.

MOTIVATING EXAMPLE



- Step 2: Attacker can communicate with DB server based on Access Control List (ACL) at Firewall.

MOTIVATING EXAMPLE



- Step 2: Attacker identifies and exploit the SQL Injection vulnerability on DB Server.
- Step 3: Persistent Access : Attacker can use the access to establish connection back to command and control (C&C) server

ATTACK ANALYSIS

- The attack was multi-stage.
- The attacker had specific attack vectors to target the vulnerabilities.
- The attack was multi-host.
- Attacker broke into/circumvented several systems.

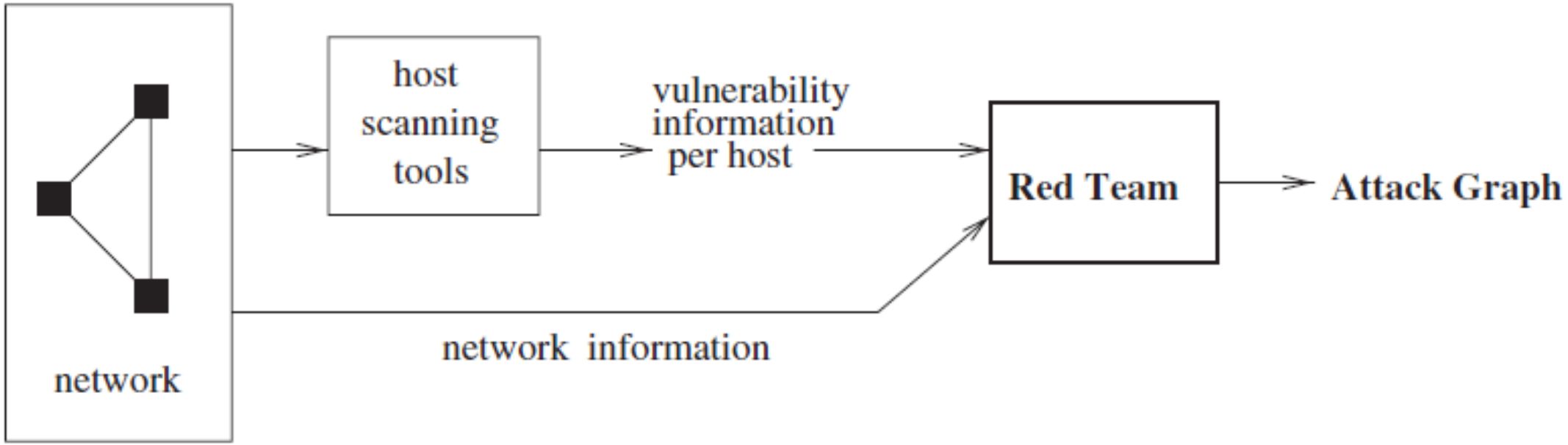
ATTACK ANALYSIS

- The attacker can gain access due to configuration errors – loosely defined ACL policies on Firewall
- The attacker can compromise hosts/software by exploiting the known vulnerabilities.
- The attacker can take the path of least-resistance to reach the goal node.
- Security is as weak as the weakest-link in the chain. Configuration errors on single node can bring down the entire network.

ATTACK ANALYSIS

- The complexity of defending against all vulnerabilities and configuration issues is non-trivial.
- Need for automated tools.
- The attack was multi-host.
- Goals: 1) Is the network vulnerable to currently known attacks?
- 2) If so, how? Which attack paths lead to compromise?
- 3) The attack representation method (ARM) should be scalable.

ATTACK GRAPH



ATTACK GRAPH

- An attack graph or AG is a tuple $G=(S, T, S_0, S_G)$.
- S is set of all states.
- $T \subseteq S \times S$ represents the transition relation.
- $S_0 \subseteq S$ represents the set of initial states.
- $S_G \subseteq S$ represents the set of success states.

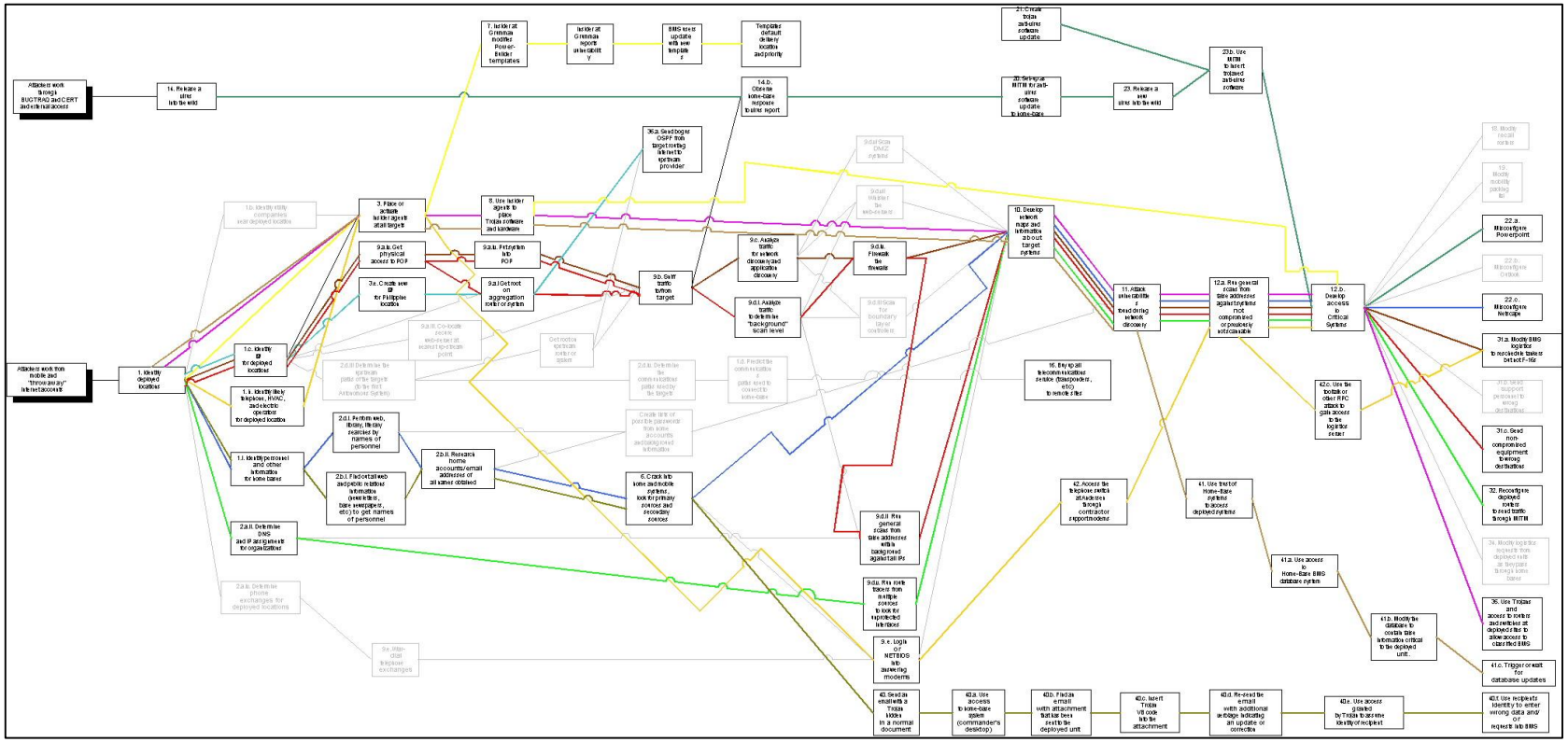
ATTACK GRAPH

- Attack Graph G = {N, E} consists of nodes (N) and edges (E). The node set can be represented as
- The nodes of attack graph can be denoted by $N = \{N_f \cup N_c \cup N_d \cup N_r\}$, where N_f denotes the primitive/fact nodes, e.g., vulExists (WebServer, BufferOverflow).
- N_c denotes the exploit, e.g., execCode (WebServer, apache).
- N_d denotes the privilege level, e.g., (user, Firewall).
- N_r represents the root or goal node, e.g., (root, DatabaseServer).

ATTACK GRAPH

- Edges of the attack graph can be denoted by $E = \{E_{pre} \cup E_{post}\}$.
- $E_{pre} \subseteq (N_f \cup N_c) \times (N_d \cup N_r)$ ensures that preconditions N_c and N_f must be met in order to achieve N_d.
- $E_{post} \subseteq (N_d \cup N_r) \times (N_f \cup N_c)$ means that post-condition N_d is achieved on satisfaction of N_c and N_f.
- Initial Conditions: Nodes of Attack Graph used as starting point by the attacker, i.e., $N_i \subset (N_f \cup N_c)$.

NEED FOR AUTOMATED TOOLS FOR ATTACK GRAPH GENERATION



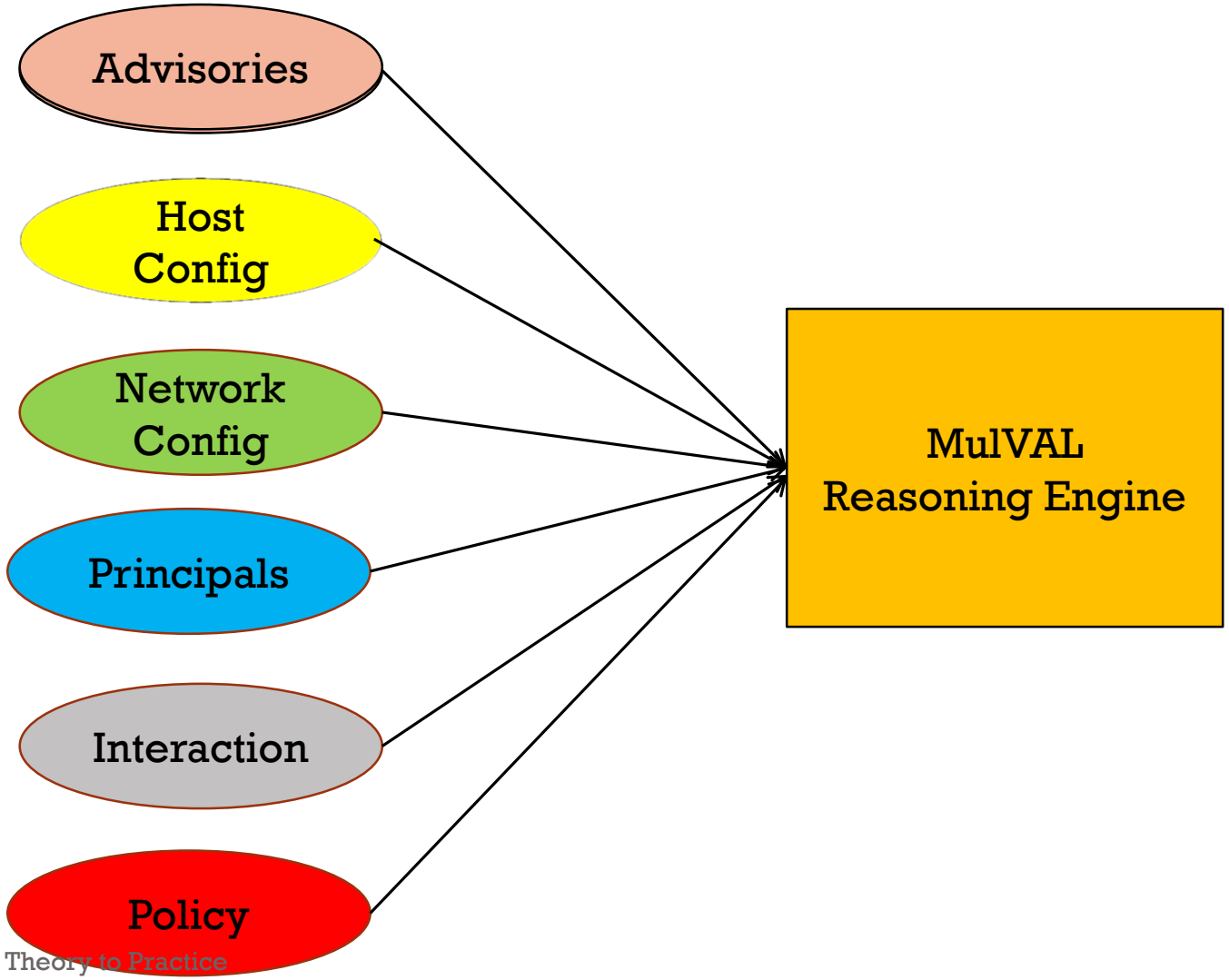
Example of Attack Graph drawn by Red Team manually.

MULVAL – AUTOMATED ATTACK GRAPH GENERATION TOOL

- An end-to-end framework and reasoning system.
- Conducts multi-stage, multi-host vulnerability analysis.
- Adopts datalog modeling language,.
- Utilizes Bug Spec, configuration, reasoning rules, system permissions.
- Vulnerability database, scanning information, security policies converted to Datalog format and fed into MulVAL engine.
- Old works did not how to automatically integrate vulnerability specifications from the bug-reporting community into the reasoning model.

MULVAL

- Vulnerabilities Reported
- Software Configuration
- Network Router and Firewall Config
- Users of the network
- Interaction model of components
- Type of access permission



MULVAL REASONING ENGINE

- A literal $p(t_1, t_2, \ldots, t_k)$ is a predicate applied to its arguments, each of which is either constant or a variable.
- Let literals in a sentence of MulVAL be represented as $L_0 :- L_1, \ldots, L_n$
- If $L_1, \ldots, L_n$ are true, L_0 is triggered.
- A clause with an empty body (RHS) is called a fact.
- A clause with a non-empty body is called a rule.

MULVAL REPRESENTATION

- Advisories
  - vulExists (webServer, 'CAN-2002-0392', httpd)
  - vulProperty ('CAN-2002-0392', remoteExploit, privilegeEscalation)
- Host Configuration
  - networkService (webServer, httpd, TCP, 80, apache)
- Network Configuration
  - hacl(internet, webServer, TCP, 80)
- Principals
  - hasAccount (user, projectPC, userAccount)
  - hasAccount (sysAdmin, webServer, root)

MULVAL REPRESENTATION

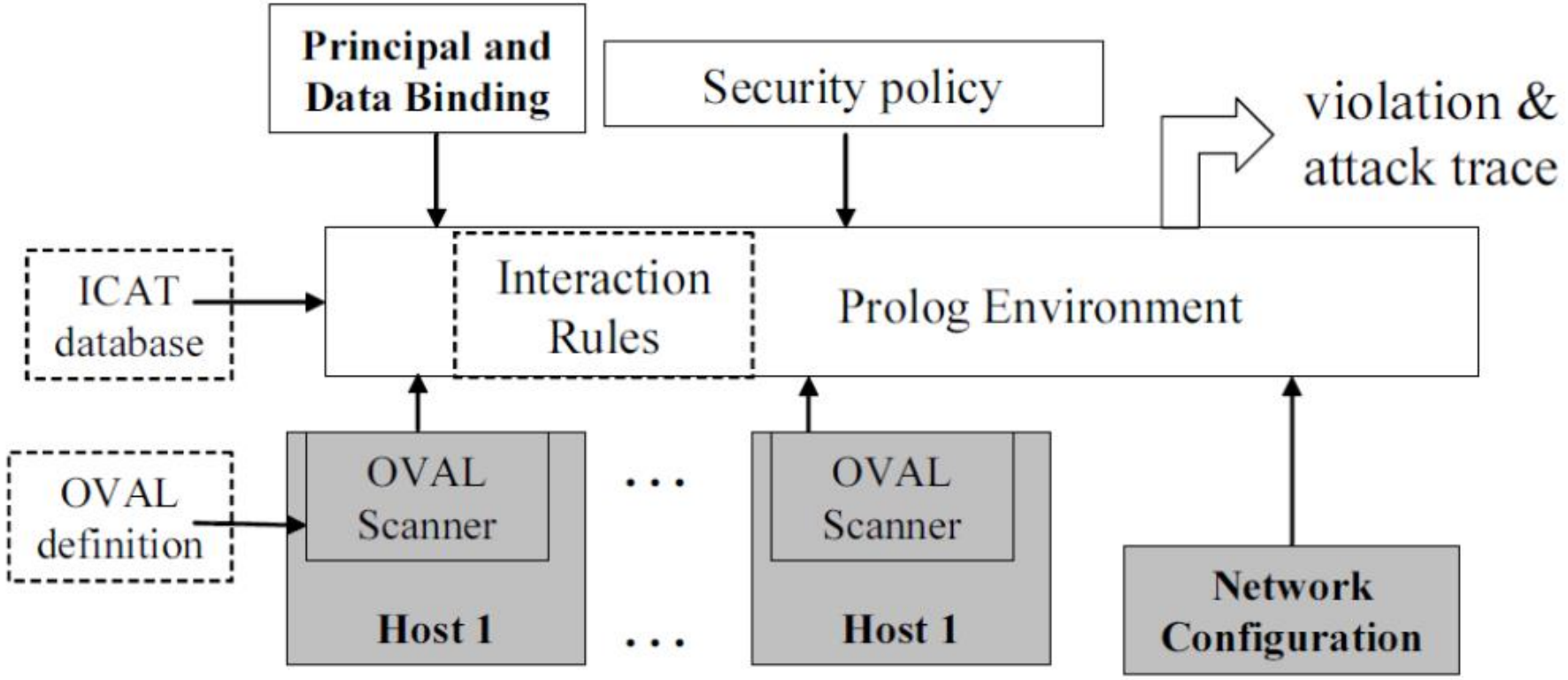
- Interaction

```
execCode (Attacker, Host, Priv) :-
        vulExists (Host, VulID, Program),
        vulProperty (VulID, remoteExploit, privEscalation),
        networkService (Host, Program, Protocol, Port, Priv),
        netAccess (Attacker, Host, Protocol, Port),
        malicious (Attacker).
```

- Policy

```
allow (Everyone, read, webPages)
allow (sysAdmin, write, webPages)
```

MULVAL FRAMEWORK



MULVAL CHECK FOR NEW VULNERABILITIES

- MulVAL conducts test to verify violation of confidentiality, integrity and availability on the target system reports the results.
- networkService (Host, Program, Protocol, Port, Priv).
- clientProgram (Host, Program, Priv).
- setuidProgram (Host, Program, Owner).
- filePath (H, Owner, Path).
- nfsExport (Server, Path, Access, Client).
- nfsMountTable (Client, ClientPath, Server, ServerPath).

MULVAL INTERACTION RULES

- Before

```
execCode(Attacker, Host, User) :-
  networkService(Host, Program,
                 Protocol, Port, User),
  vulExists(Host, VulID, Program,
            remoteExploit, privEscalation),
  netAccess(Attacker, Host, Protocol, Port).
```

- After

```
execCode(Attacker, Host, User) :-
  networkService(Host, Program,
                 Protocol, Port, User),
  vulExists(Host, VulID, Program,
            remoteExploit, privEscalation),
  netAccess(Attacker, Host, Protocol, Port),
  assert_trace(because(
 'remote exploit of a server program',
   execCode(Attacker, Host, User),
     [networkService(Host, Program,
                     Protocol, Port, User),
      vulExists(Host, VulID, Program,
                remoteExploit, privEscalation),
      netAccess(Attacker, Host,
                Protocol, Port)])).
```

VULNERABILITY EFFECT

- Exploitable Range

  - Local: a local exploit requires that the attacker already have some local access on the host.
  - Remote.

- Policy

  - Confidentiality loss.
  - Integrity loss.
  - Denial of Service.
  - Privilege Escalation

  - Example: vulProperty ('CVE-2004-00495', localExploit, privEscalation).

MULVAL EXPLOIT RULES

- execCode (P, H, UserPriv)
- Principal P can execute arbitrary code with UserPriv on machine H.
- netAccess (P, H, Protocol, Port)
- Principal P can sent packets to Port on machine H through protocol
- Example: Remote Exploit of the client program.

```
execCode (Attacker, Host, Priv) :-
        vulExists (Host, VulID, Program),
        vulProperty (VulID, remoteExploit, privEscalation),
        networkService (Host, Program, Protocol, Port, Priv),
        netAccess (Attacker, Host, Protocol, Port),
        malicious (Attacker).
```

MODELING MULTI-STAGE ATTACKS

- If an attacker P accesses machine H with Owner's privilege, then he can have arbitrary access to files owned by the Owner.

```
accessFile (P, H, Access, Path) :-
        execCode (P, H, Owner),
        filePath (H, Owner, Path).
```

- If an attacker can modify files under Owner's directory, he can gain privilege of the Owner.

```
execCode (Attacker, H, Owner) :-
        accessFile (Attacker, H, write, Path),
        filePath (H, Owner, Path),
        malicious (Attacker).
```

HOST ACCESS CONTROL LIST/ POLICY

- hacl (Source, Dest, Protocol, DestPort)
- Multihop network access

```
netAccess (P, H2, Protocol, Port) :-
        execCode (P,  H1, Priv),
        hacl (H1, H2, Protocol, Port).
```

- allow(Principal, Access, Data)

```
allow (Everyone, read, webPages).
allow (user, Access, projectPlan).
allow (sysAdmin, Access, Data).
```

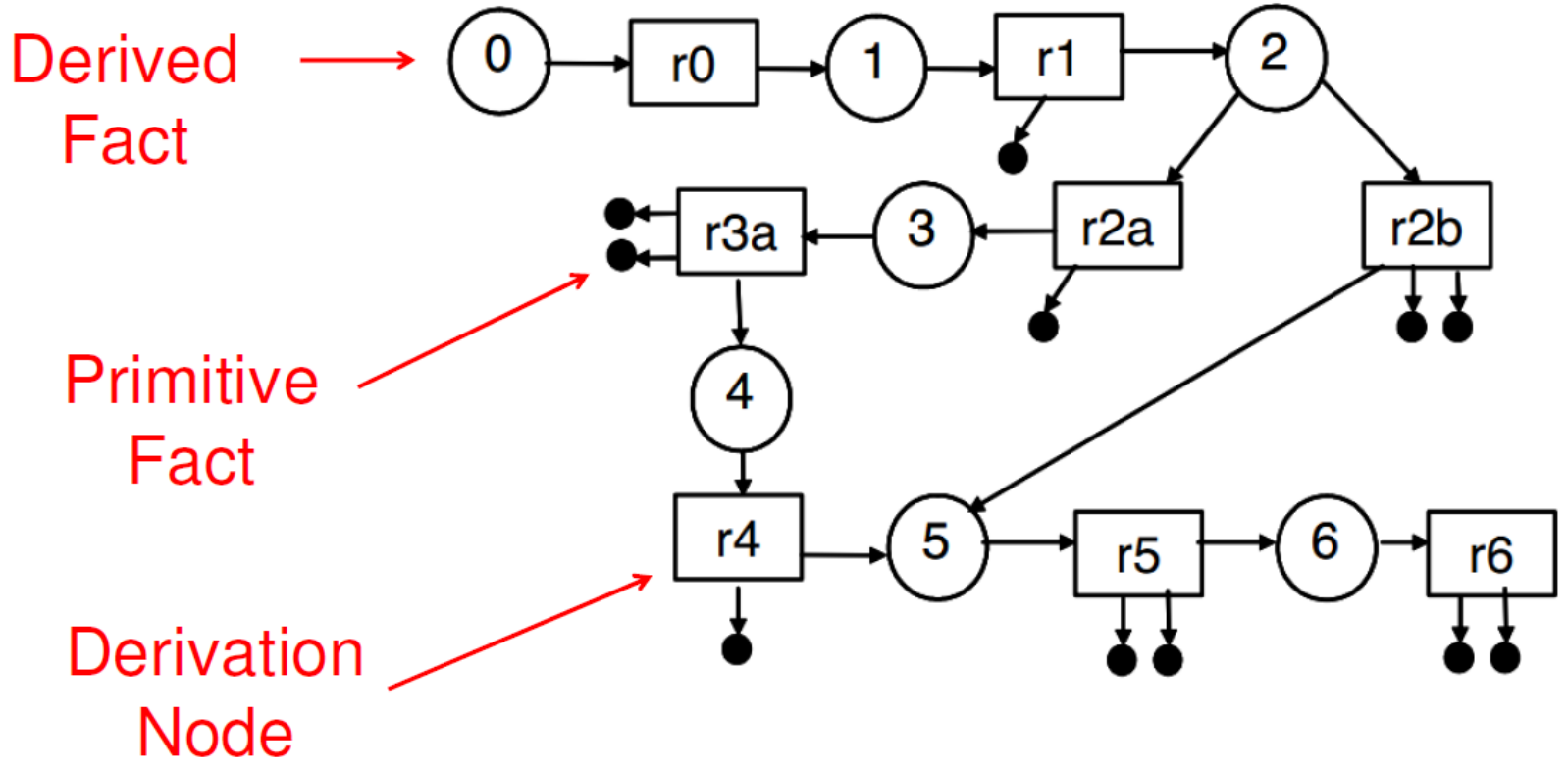
MULVAL BINDING INFORMATION/ ALGORITHM

- hasAccount (user, projectPC, userAccount).
- hasAccount (sysAdmin, webServer, root).
- dataBind (projectPlan, workstation, '/home').
- dataBind (webPages, webServer, '/www').

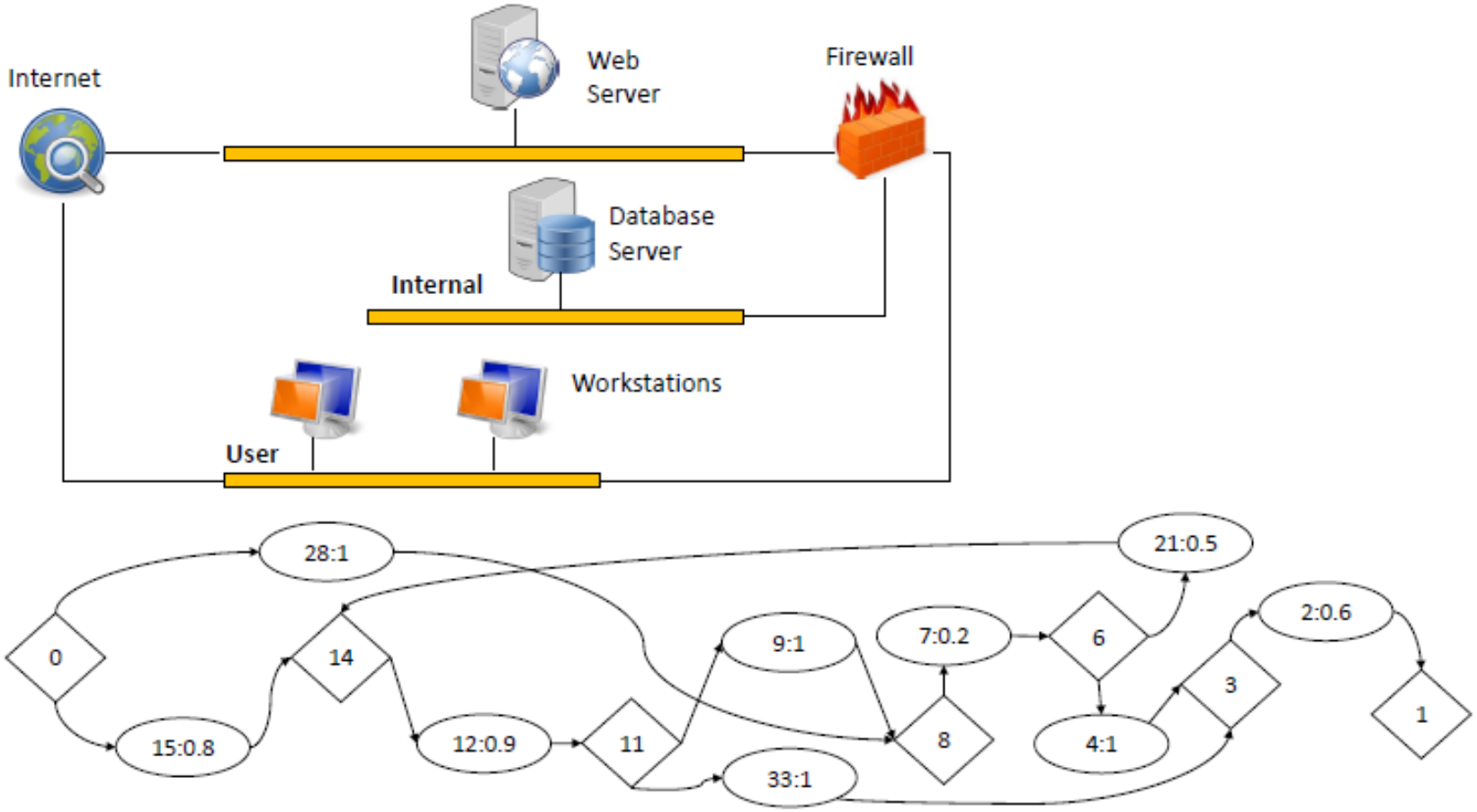
```
access (P, Access, Data) :-
        dataBind (Data, H, Path),
        accessFile (P, H, Access, Path).
```

```
policyViolation (P, Access, Data) :-
        access (P, Access, Data),
        notAllow (P, Access, Data).
```

LOGICAL ATTACK GRAPH



LOGICAL ATTACK GRAPH



LOGICAL ATTACK GRAPH

- The initial privilege of attacker located on the Internet is represented by Node 0.
- The exploit of web server Node 7 (7:0.2) in attack graph can occur if an attacker can access the web server via TCP port 80 Node 8.
- Network Access to database server Node 3 can be obtained by web server Node 6 or workstation Node 11.
- A multi-hop attack can be launched by the remote attacker to first exploit web server vulnerability and use it as a stepping stone to access database server (0, 28, 8, 7, 6, 4, 3, 2, 1).
- Another attack scenario involves attacker tricking workstation user to click on a malicious website (0, 15, 14, 12, 11,..).
- The attack graph can help network admin to identify multiple attack paths that can lead to a compromise of various hosts.

PROBABILISTIC ATTACK GRAPHS

- Multi-stage network attacks exploit the dependencies between security vulnerabilities.
- Binary representation of network security secure or insecure doesn't show relative importance of the vulnerabilities and relatively secure option amongst security configurations.
- Probabilistic attack graphs utilize security metrics based on existing vulnerability scoring systems such as CVSS.
- The model assigns probabilistic values to the network vulnerabilities i.e. likelihood of a vulnerability to be exploited.
- The model incorporates casual dependencies between the network vulnerabilities.

PROBABILISTIC ATTACK GRAPH

- Priori Probability: Likelihood of a threat becoming active and difficulty of vulnerability exploitation.
- Probability value of each internal risk node $e \in N_c$ is denoted by $G_m[e]$.
- We normalize the CVSS score between $(0,1]$ for probabilistic attack graphs.
- $\Pr(e) = \frac{BS(e)}{10} \; \forall e \in N_c$.
- The likelihood of attack propagation depends upon conjunctive and disjunctive relation between exploits.

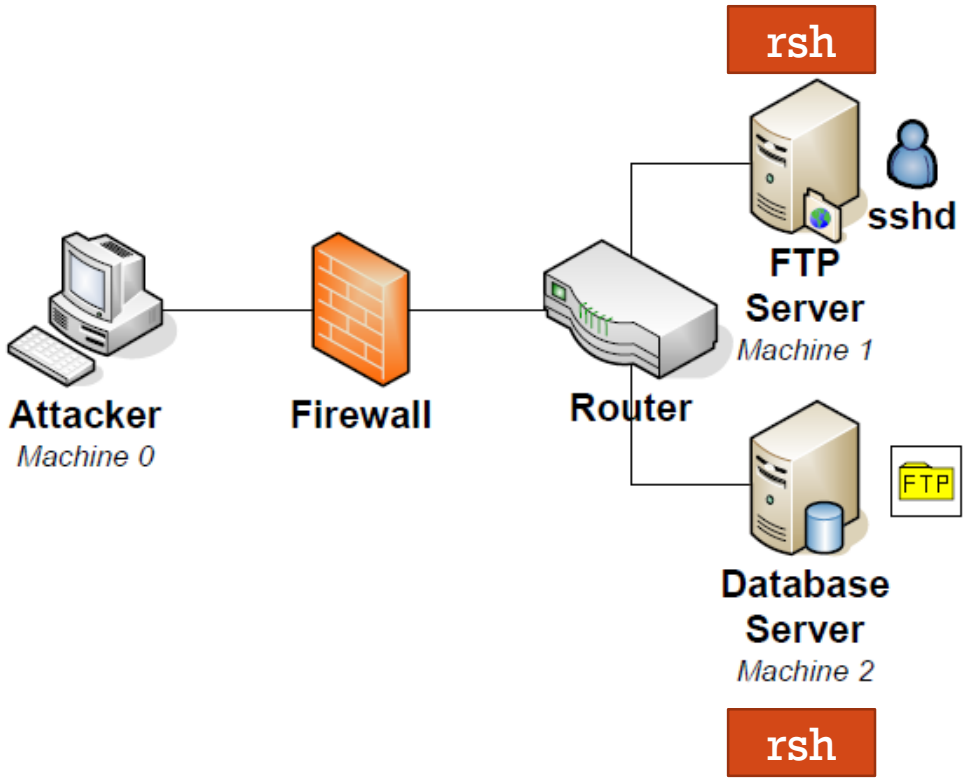
PROBABILISTIC ATTACK GRAPH

- Consider the predecessor W=parent(n) of any attack step node in attack graph $e \in N_c$, *conditional probability of step node;*
- $\Pr\left(\frac{n}{W}\right) = G_m[n] \times \Pi_{s \in W} \Pr\left(\frac{s}{W}\right).$
- If predecessor node has a disjunctive relation with predecessor set W=parent(n), conditional probability values is given by;
- $\Pr\left(\frac{n}{W}\right) = 1 - \Pi_{s \in W}\left(1 - \Pr\left(\frac{s}{W}\right)\right).$

PROBABILISTIC ATTACK GRAPH

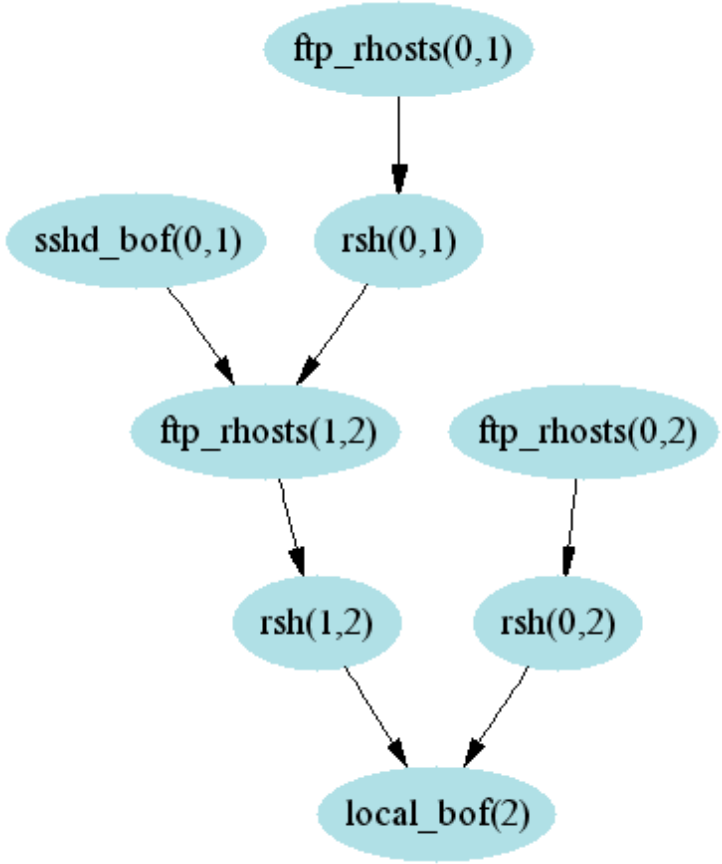
- For the attack step nodes, $n \in N_c$, with predecessor set W=parent(n), cumulative probability;
- $\Pr(n) = \Pr\left(\frac{n}{W}\right) \times \Pi_{s \in W} \Pr(s)$.
- For the attack privilege nodes $n \in N_c$ with predecessor set W=parent(n), cumulative probability;
- $\Pr(n) = 1 - \Pi_{s \in W}(1\text{-}\Pr(s))$.

ATTACK GRAPH EXAMPLE

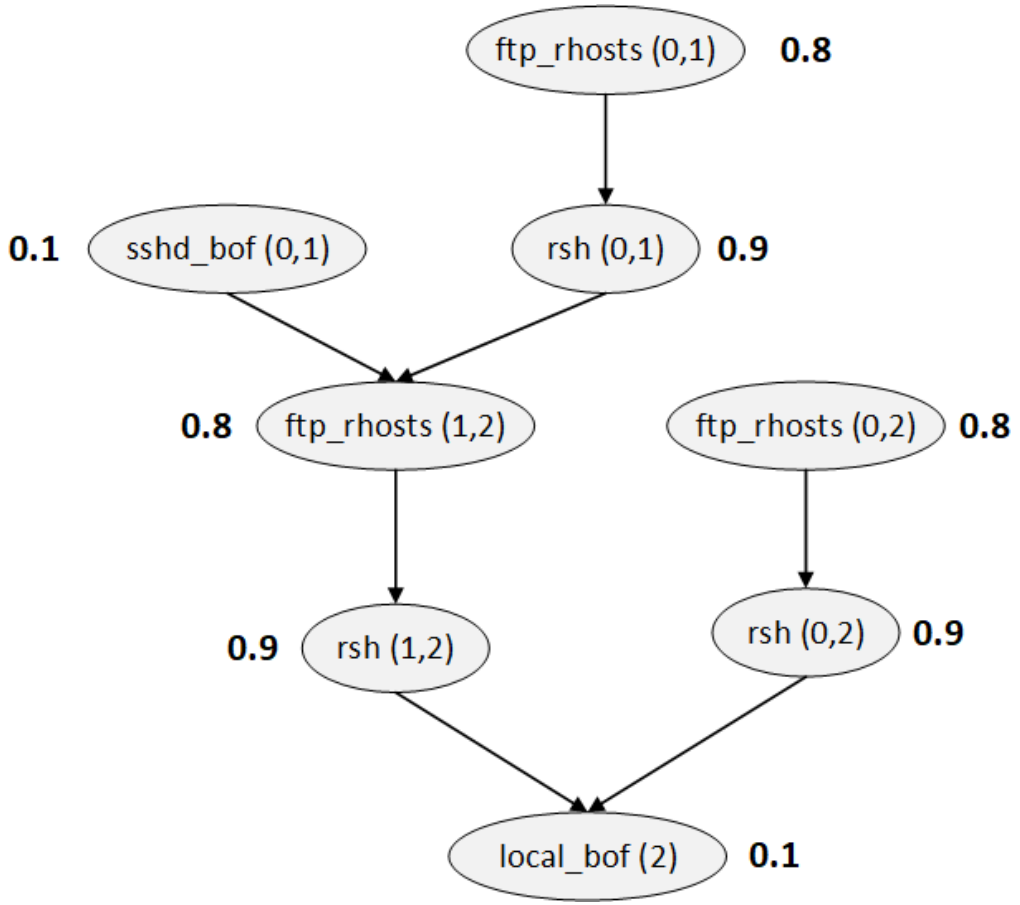


- *sshd_bof(0,1)→ftp_rhosts(1,2)→rsh(1,2)→local_bof(2).*
- *ftp_rhosts(0,1)→rsh(0,1)→ftp_rhosts(1,2)→rsh(1,2)→local_bof(2).*
- *ftp_rhosts(0,2)→rsh(0,2)→local_bof(2).*

ATTACK GRAPH EXAMPLE

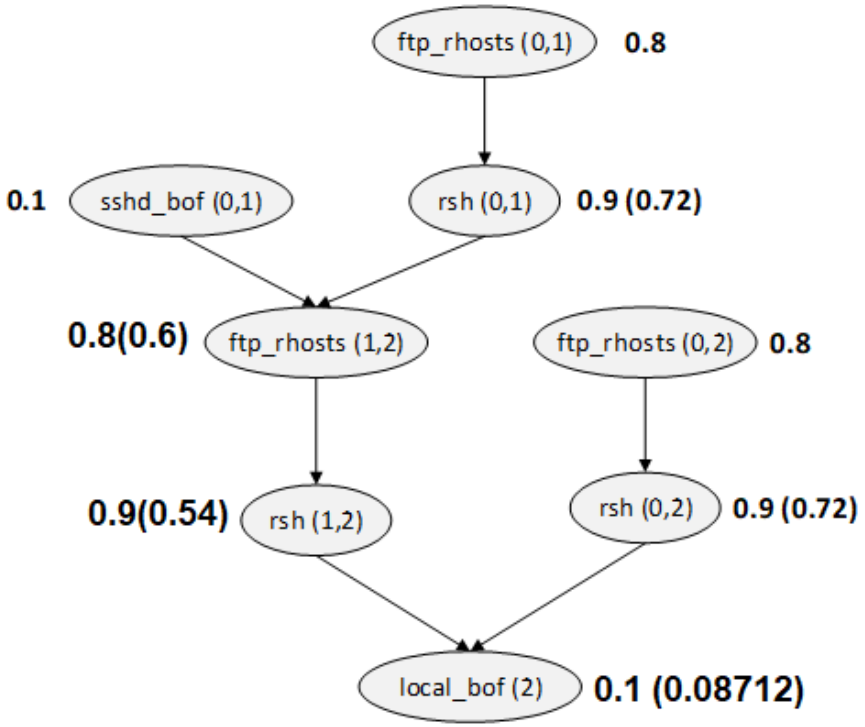


PROBABILISTIC ATTACK GRAPH



- Numbers are estimated probabilities of occurrence for individual exploits, based on their relative difficulty.
- The *ftp_rhosts* and *rsh* exploits take advantage of normal services in a clever way and do not require much attacker skill
- A bit more skill is required for *ftp_rhosts* in crafting a .rhost file.
- *sshd_bof* and *local_bof* are buffer-overflow attacks, which require more expertise.

PROBABILITIES PROPAGATED THROUGH ATTACK GRAPH



- When one exploit must follow another in a path, this means **both** are needed to eventually reach the goal, so their probabilities are multiplied: $p(A \text{ and } B) = p(A)p(B)$

- When a choice of paths is possible, **either** is sufficient for reaching the goal: $p(A \text{ or } B) = p(A) + p(B) - p(A)p(B)$.

RISK MITIGATION USING PROBABILISTIC METRICS

- We can use the probabilistic metrics derived from attack graph for making changes in the network
- To block traffic from the attacker;

1) Make no changes to the network (baseline)
2) Block *ftp* traffic to prevent *ftp_rhosts(0,1)* and *ftp_rhosts(0,2).*
3) Block *rsh* traffic to prevent *rsh(0,1)* and *rsh(0,2).*
4) Block *ssh* traffic to prevent *sshd_bof(0,1)*.

PROBABILITIES PROPAGATED THROUGH ATTACK GRAPH

- **Nodes ftp_rhosts(1,2) , ftp_rhosts(0,2), and local_bof(2) are OR nodes**

Initial Values of vulnerability exploitation

- P(ftp_rhosts(0,1)) = 0.8
- P(ftp_rhosts(0,2)) = 0.8
- P(ftp_rhosts(1,2)) = 0.8
- P(sshd_bof(0,1)) = 0.1
- P(rsh(0,1)) = 0.9
- P(rsh(1,2))=0.9
- P(rsh(0,2)) = 0.9
- P(local_bof(2)) = 0.1

PROBABILITIES PROPAGATED THROUGH ATTACK GRAPH

Make no-change

- P(rsh(0,1)) = 0.9*0.8 = 0.72
- P(ftp_rhosts(1,2)) = (0.72+0.1 – (0.72*0.1))*0.8 ~ 0.60
- P(rsh(1,2))=0.9*0.60 = 0.54
- P(rsh(0,2))= 0.9*0.8 = 0.72
- P(local_bof(2)) = (0.54+0.72 – (0.54*0.72))*0.1 ~ 0.08712

PROBABILITIES PROPAGATED THROUGH ATTACK GRAPH

Block ssh (sshd_bof(0,1))

- P(rsh(0,1)) = 0.9*0.8 = 0.72
- P(ftp_rhosts(1,2)) = (0.72*0.8) =0.576
- P(rsh(1,2))=0.9*0.576 ~ 0.52
- P(rsh(0,2))= 0.9*0.8 = 0.72
- P(local_bof(2)) = (0.52+0.72-(0.52*0.72))*0.1 ~ 0.086

PROBABILITIES PROPAGATED THROUGH ATTACK GRAPH

Block rsh(0,1), rsh(0,2)

- P(ftp_rhosts(1,2)) = (0.1*0.8) =0.08
- P(rsh(1,2))=0.9*0.08=0.072
- P(local_bof(2)) = 0.072*0.1=0.0072

PROBABILITIES PROPAGATED THROUGH ATTACK GRAPH

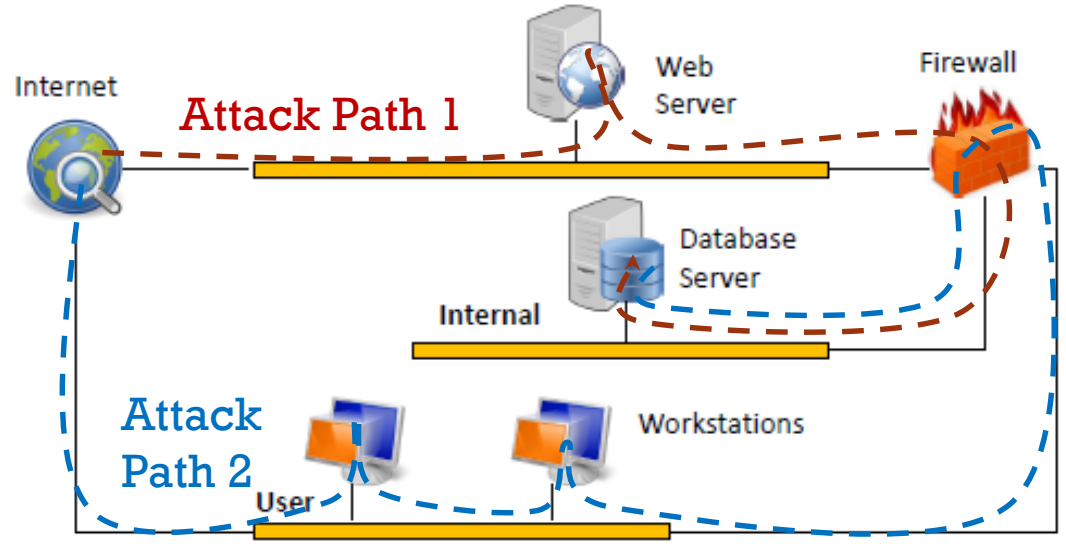
Block ftp_rhosts(0,1), ftp_rhosts(0,2)

- P(ftp_rhosts(1,2)) = (0.1*0.8) =0.08
- P(rsh(1,2))=0.9*0.08=0.072
- P(local_bof(2)) = 0.072*0.1=0.0072

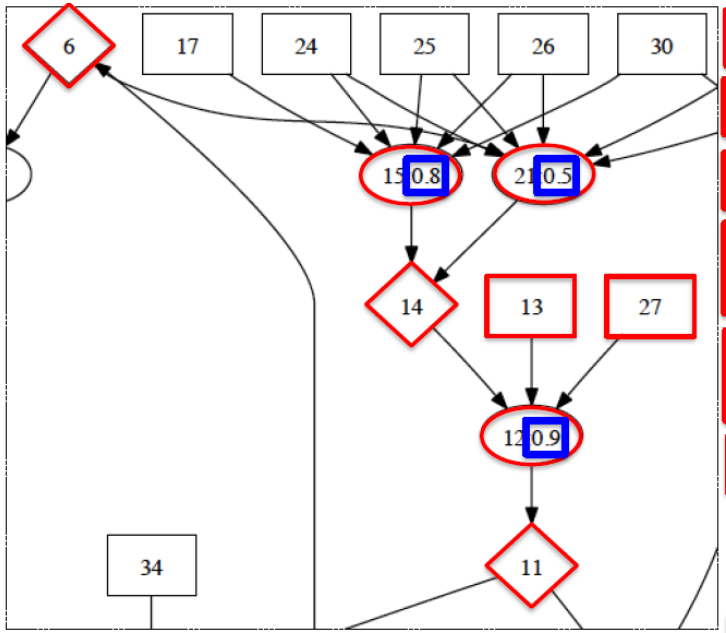
RISK MITIGATION OPTIONS COMPARISON

- We can make comparison of the relative security among the options.
- Make no-change p=0.052.
- Blocking *rsh traffic* from the Attacker leaves remaining 4-step attack path with total probability $p = 0.1 \times 0.8 \times 0.9 \times 0.1 = 0.0072$.
- Blocking *ftp traffic, p=0.0072*.
- But blocking *ssh traffic* leaves 2 attack paths with probability $p \sim 0.086$, i.e., compromise is more likely as compared to blocking *rsh* or *ftp*.

RISK ANALYSIS EXAMPLE



RISK ANALYSIS EXAMPLE



6:execCode(webServer,apache)

11:execCode(workStation,normalAccount)

12:remote exploit of a client program

13:hasAccount(secretary,workStation,normalAccount)

14:canAccessMaliciousInput(workStation,secretary,internetExplorer)

15:Browsing a malicious website

17:hacl(workStation,internet,httpProtocol,httpPort)

21:Browsing a compromised website

24:isUserMachine(workStation)

25:isWebBrowser(internetExplorer)

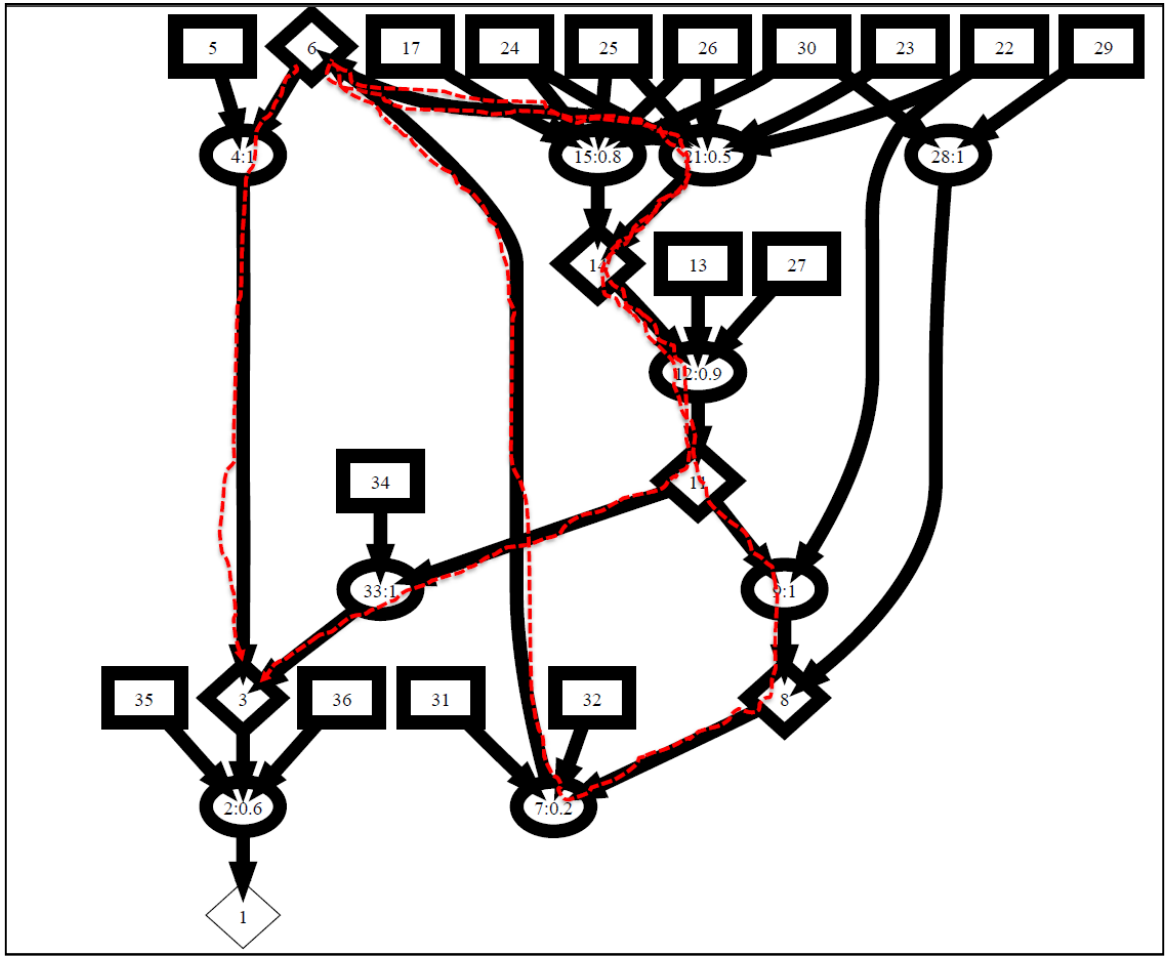
26:inCompetent(secretary)

27:vulExists(workStation,cve_IE,internetExplorer,remoteExploit,privEscalation)

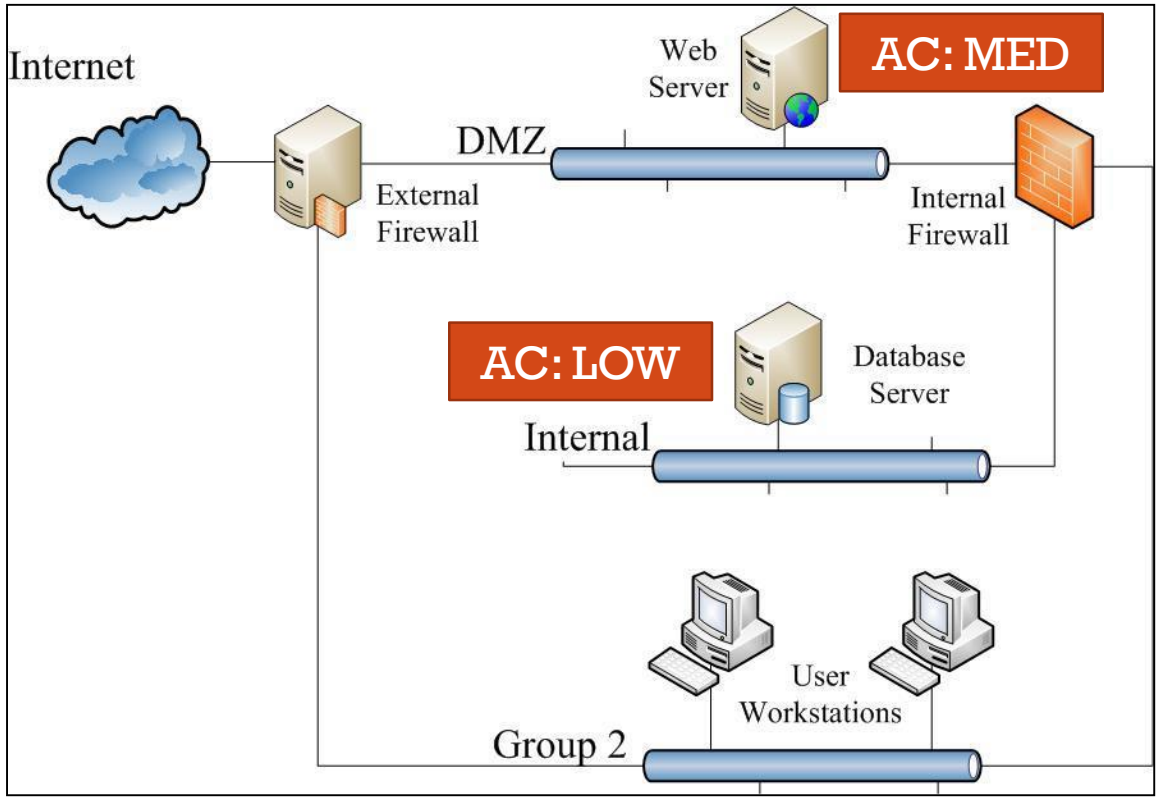
30:attackerLocated(internet)

34:hacl(workStation,dbServer,dbProtocol,dbPort)

RISK ANALYSIS EXAMPLE



RISK MITIGATION PRIORITIZATION



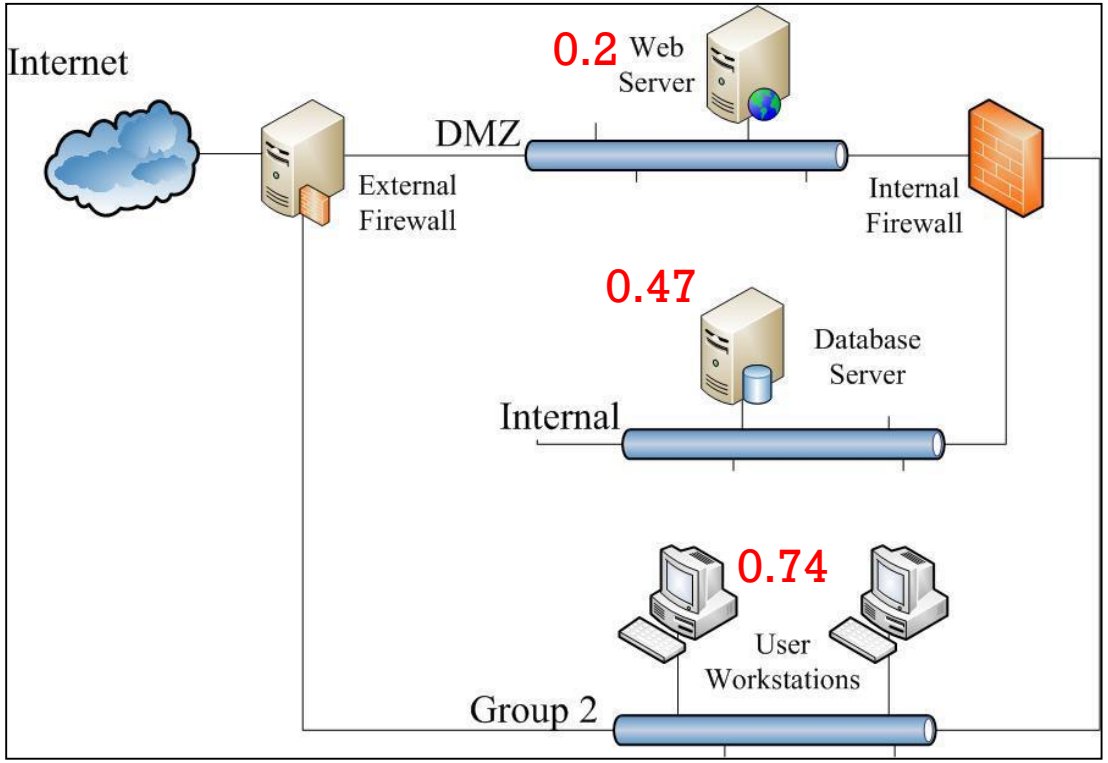
- Given three hardening options:

1) Patching the web server.
2) Patching the db server.
3) Patching the workstation.

Which one would you patch first?

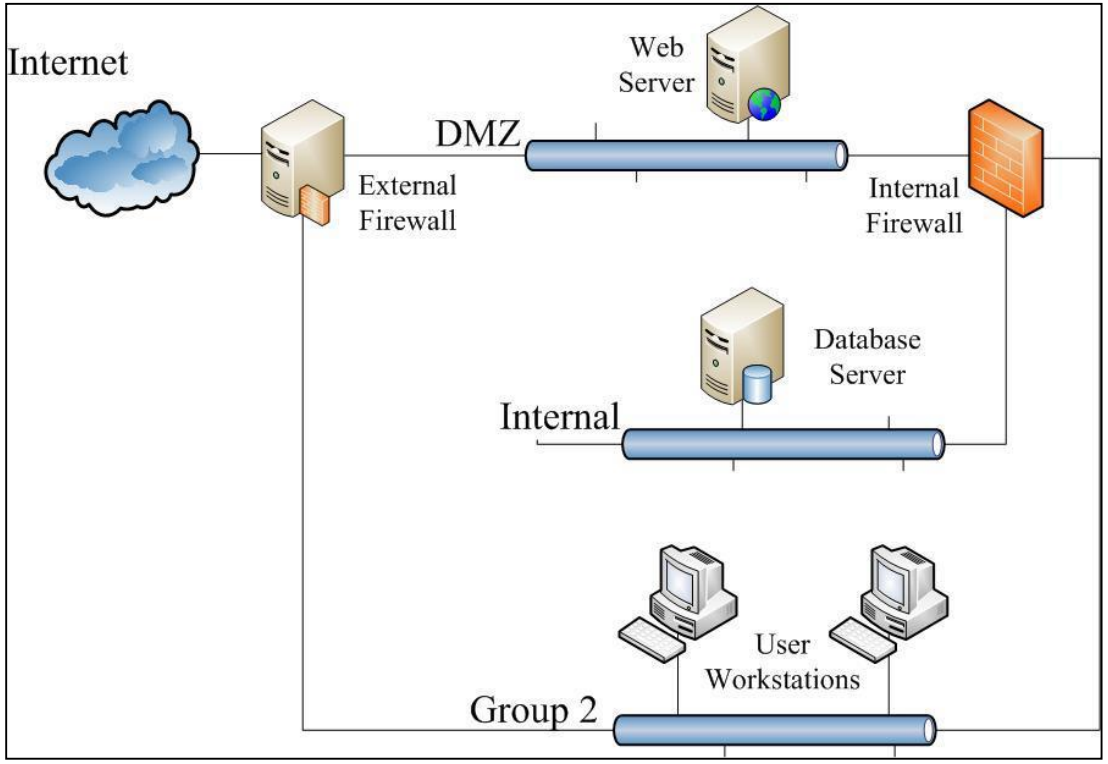
AC: high:0.2, medium:0.6, low:0.9

RISK MITIGATION: PATCH WEB SERVER



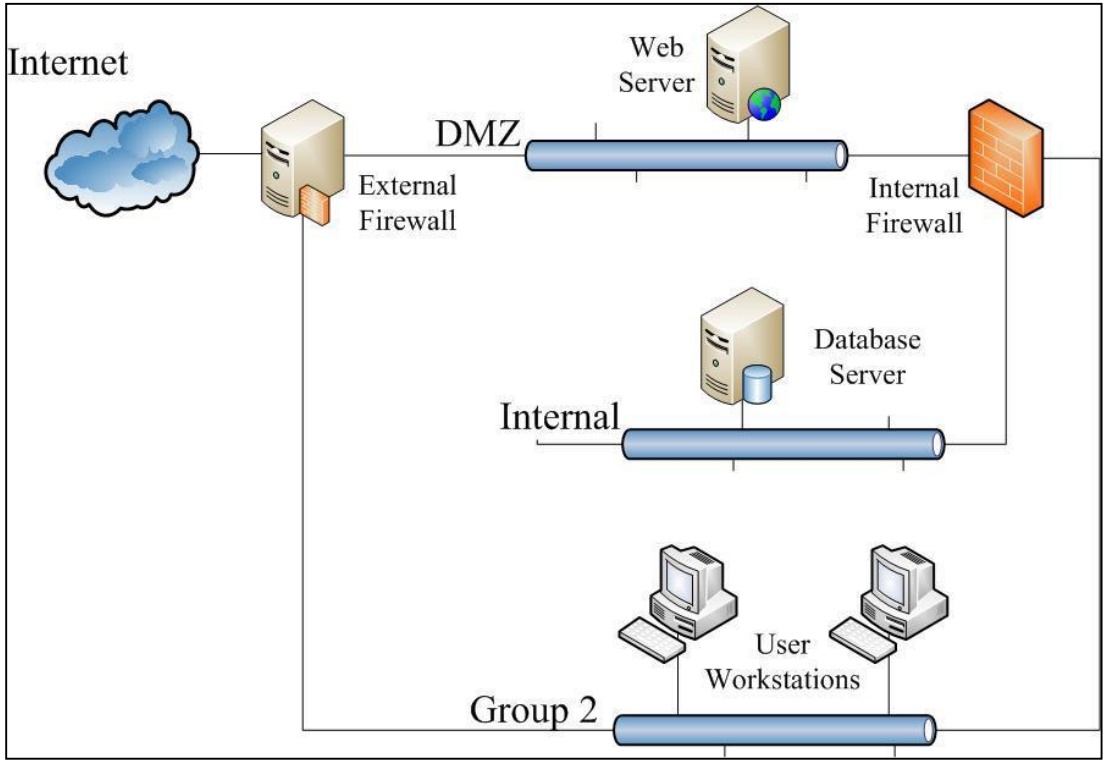
- Before: execCode(dbServer, root): 0.47
  execCode(webServer,apache:0.2
  execCode(workS,normalAcct):0.74
- After: execCode(dbServer, root): 0.43
  execCode(webServer,apache:0.0
  execCode(workS,normalAcct):0.74

RISK MITIGATION: PATCH DB SERVER



- Before: execCode(dbServer, root): 0.47
  execCode(webServer,apache:0.2
  execCode(workS,normalAcct):0.74
- After: execCode(dbServer, root): 0.0
  execCode(webServer,apache:0.2
  execCode(workS,normalAcct):0.74

RISK MITIGATION: PATCH WORKSTATIONS



- Before: execCode(dbServer, root): 0.47
  execCode(webServer,apache:0.2
  execCode(workS,normalAcct):0.74
- After: execCode(dbServer, root): 0.12
  execCode(webServer,apache:0.2
  execCode(workS,normalAcct):0.0

RISK MITIGATION FOR THE ATTACK GRAPH

Host	Attack Prob.	Patch Web Server	Patch DB Server	Patch Workstation	Network Access Change
DB Server	0.47	0.43	0	0.12	0.12
Web Server	0.2	0	0.2	0.2	0.2
Workstations	0.74	0.74	0.74	0	0.74

RISK MITIGATION FOR THE ATTACK GRAPH

- The Database Server can be patched to eliminate the risk of losing important information,
- The downtime associated with patching the Database Server may impact the business.
- Problem: An attacker may still target workstations;
- The patch on the workstation can eliminate the attack path to the Database server.
- Problem: The attacker can still target the Web Server.
- Firewall option - be considered as a countermeasure, to block access from Workstations to Database and Web Server.

OPTIMIZING SECURITY HARDENING

- Let Pr[execCode(dbServer, root)] = p_1

  Pr[execCode(webServer, apache)] = p_2

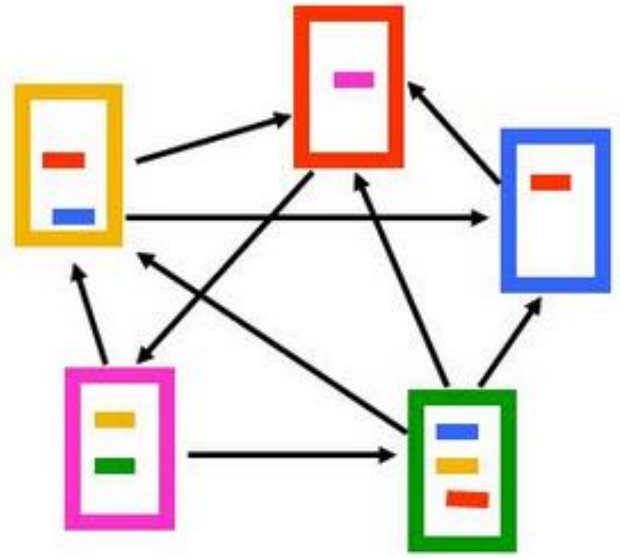
  Pr[execCode(workStation, normalAcct)] = p_3
- If c_1, c_2, c_3 are the cost of compromise of these hosts respectively.
- Expected Loss LE = $c_1 \times p_1 + c_2 \times p_2 + c_3 \times p_3$.
- The hardening measures $H_1, H_2, \ldots, H_n$ have cost as well.
- The goal is to optimize the cost of hardening measures that can minimize LE.

ATTACK GRAPH RANKING

- States in the attack graph can be assigned ranking.
- A ranking algorithm can be used based on state probability values – Mehta, et. al.
- The scheme utilizes ranking similar to Google's Page Rank algorithm.
- Random Surfer Model.

PAGE RANK

- Pick a page at random.
- With probability (1-d) jump to a random page.
- With probability d follow random outgoing link.
- Rank according to stationary distribution.

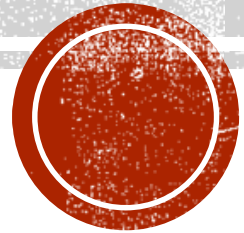


1. Red Page
2. Purple Page
3. Yellow Page
4. Blue Page
5. Green Page

ATTACK GRAPH RANKING

- If the attack graph has N nodes, In(j) be set of nodes linking node 'j', and Out(j) be set of out-links from node 'j'.
- *Probability of attacker being on node 'i' is given by,*
- $\pi_i = \frac{1-d}{N} + d \times \sum_{j \in In(i)} \frac{\pi_j}{|Out(j)|}$
- The PageRank vector for attack graph is $R = (r_1, r_2, \ldots, r_N)^T$, where the rank of node 'i' is r_i.
- Equation for π_i is computed recursively until convergence is achieved.

ATTACK TREE



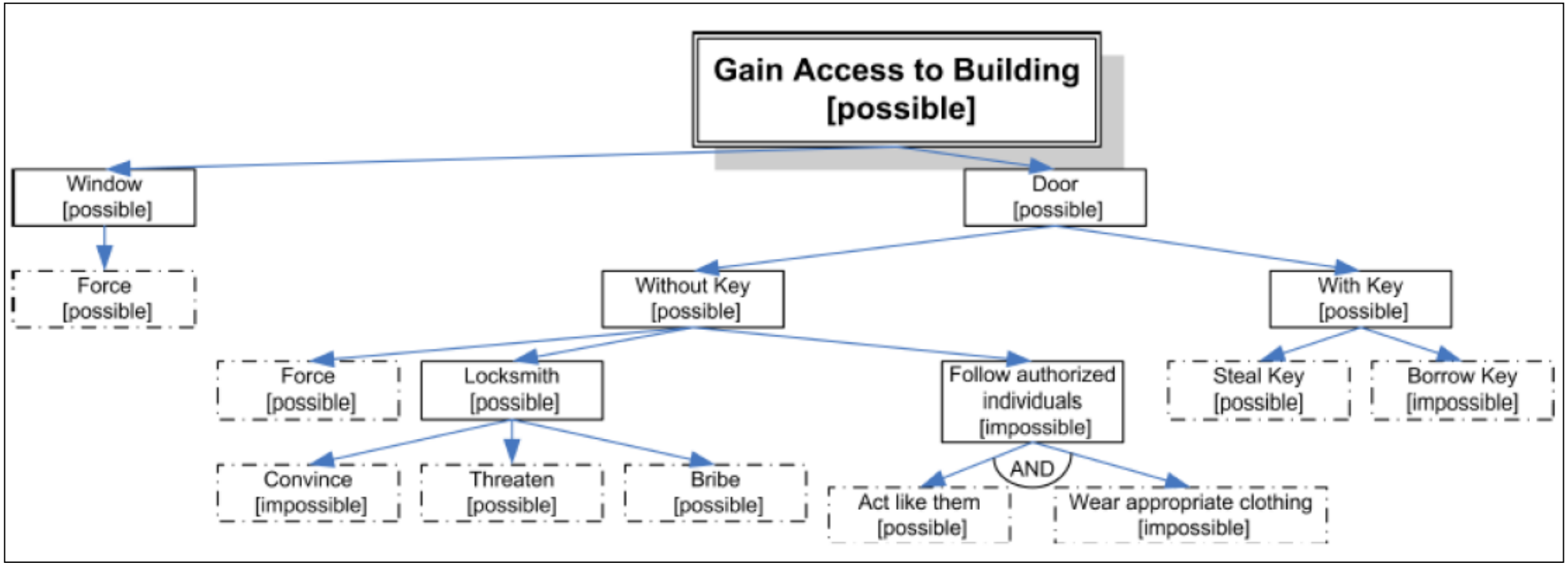
ATTACK TREE

- Method for representing system security.
- Network attacks can be modeled using attack trees (ATs).
- Attack Tree consists of AND nodes and OR nodes.
- AND nodes represent conditions to be fulfilled to achieve goal node.
- OR nodes represent one or more ways of achieving goal node.

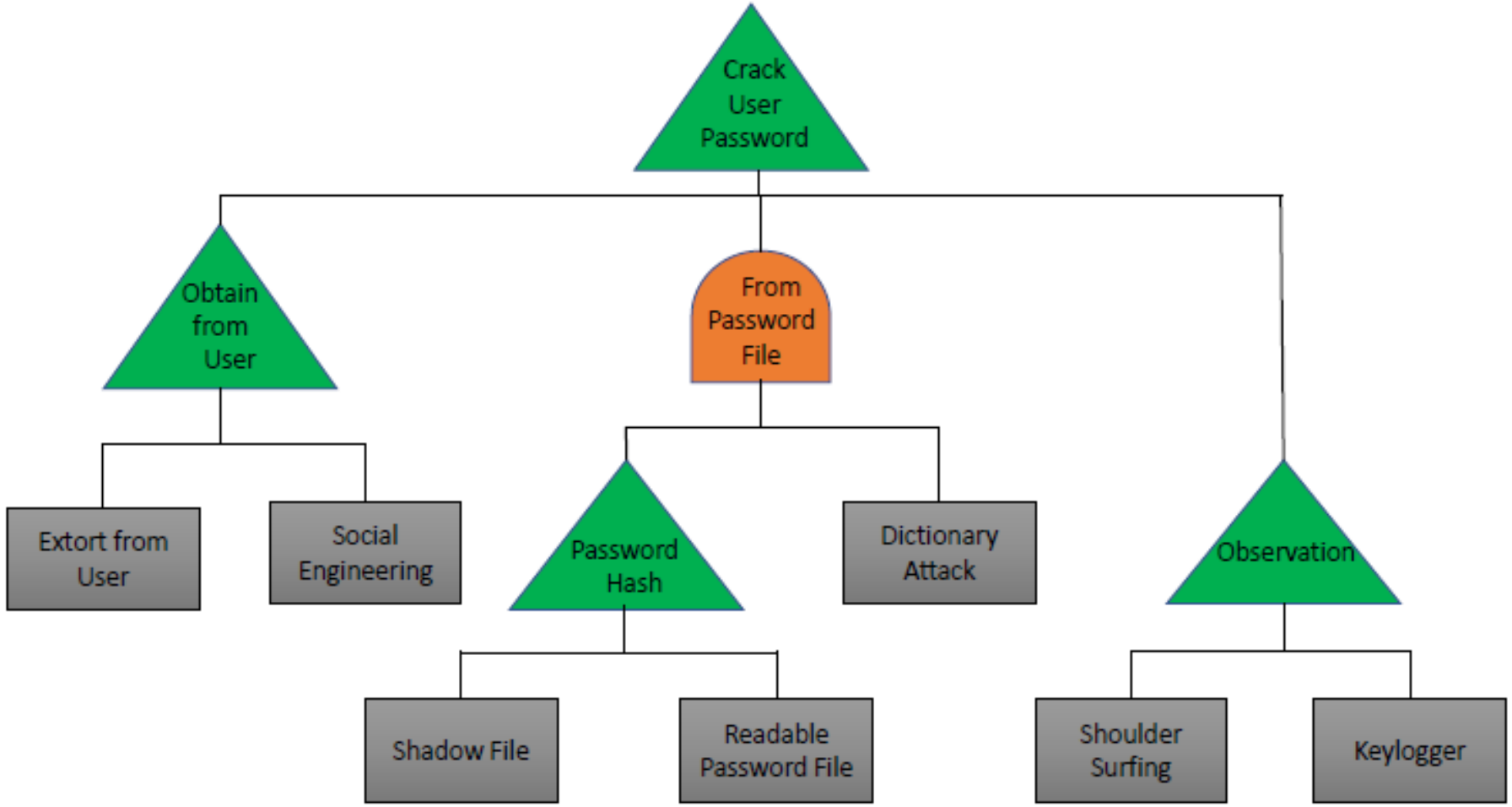
ATTACK TREE

- An Attack Tree can be defined by three tuple $(N, \rightarrow, N_r)$.
- N is all possible nodes in the tree;
- $S^+(N)$ is a multi-set of all possible subsets of nodes N;
- $\rightarrow \subseteq N \times S^+(N)$ denotes transition relation;
- N_R represents the goal node of the attack tree.
- $S^+(N) = \{\{$Obtain from User, Crack Password$\}, \{$From Password File, Crack Password$\}, \{$Observation, Crack Password$\}\}$.
- $N_R = \{$Crack Password$\}$.

ATTACK TREE EXAMPLE 1



ATTACK TREE EXAMPLE 2



ATTACK TREE EXAMPLE

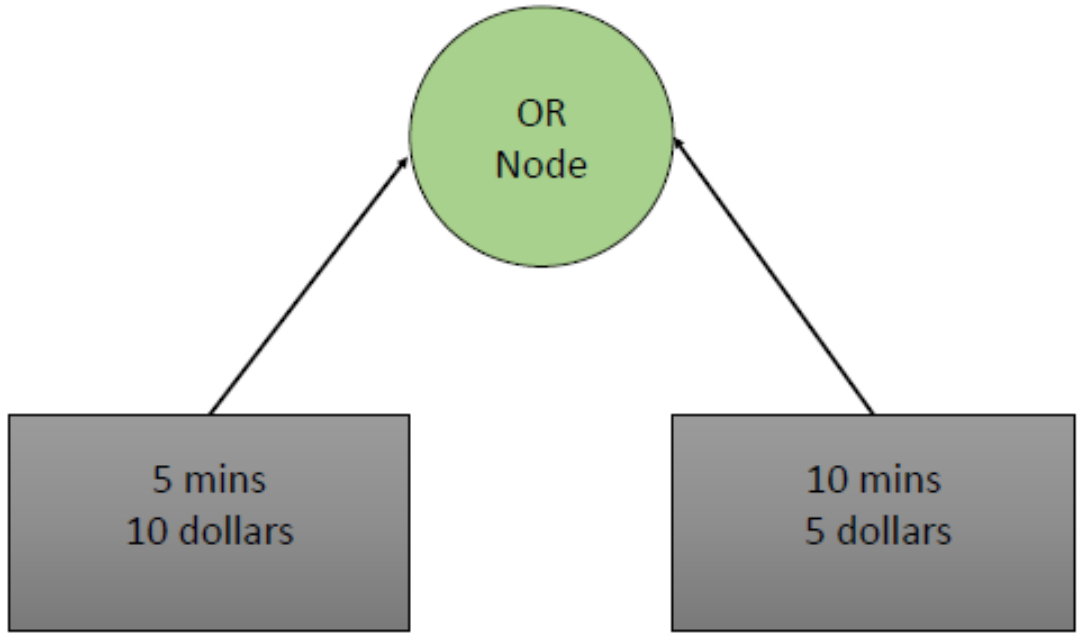
- In the AT example *Password Hash* can be obtained by the attacker from *Shadow File* OR *Readable Password File*.
- Similarly password can also be obtained by *Extortion* or *Social Engineering*.
- Both these techniques in addition to other methods such as *Observation* can be used to crack password.
- The combination of tree nodes (attack suites) can help attacker reach his desired goal.

ATTACK COMPONENT AND ATTACK SUITE

- Attack Component is a set of attack vectors available to the attacker to reach desired goal, e.g., C= {weak credentials, buffer overflow, root access}.
- A finite non-empty multi-set of C is defined as an attack.
- Attack suite can be defined as finite set of attacks {{weak credentials, root access},{buffer overflow, root access}}.
- Universe of attacks $A = M^{+}(C)$.
- Universe of attack suites S=P(A).

ATTACK TREE UNDECIDABLE CASE

- Attacker needs to invest 5 mins and 10$ in case of leaf node on left.
- 10 mins and 5$ in case of leaf node on the right.
- Attack profile with cheapest attack in shortest amount of time is undecidable in this example.



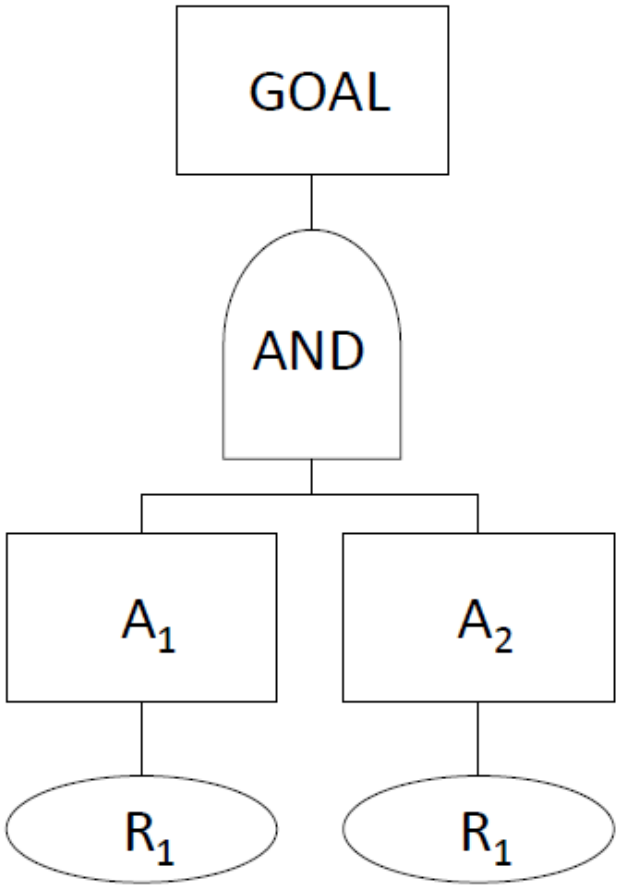
ATTACK COUNTERMEASURE TREE (ACT)

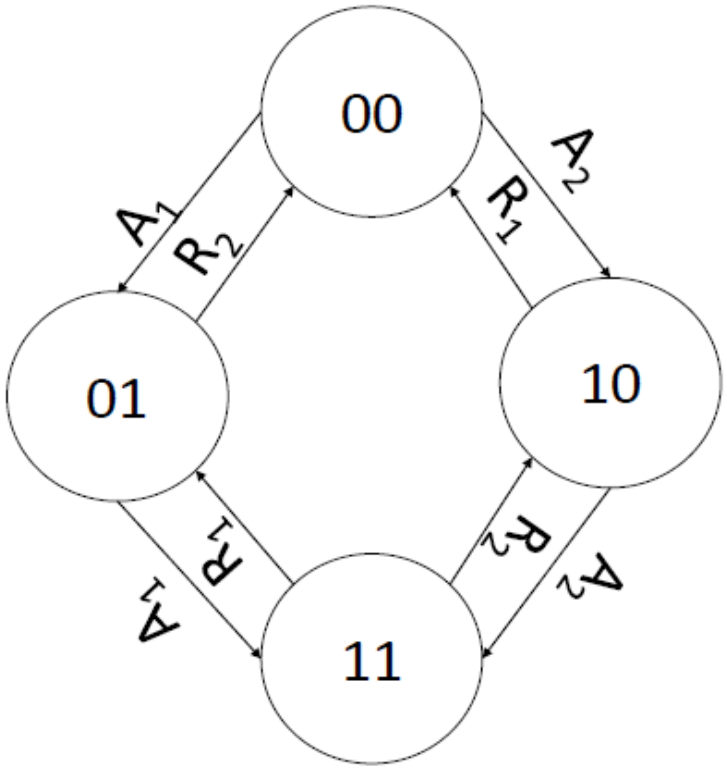
ACT Qualitative and Quantitative Analysis

ATTACK COUNTERMEASURE TREE (ACT)

- Attack Trees do not take into account the defense mechanisms.
- Attack Response Trees use state space representation of attacks and countermeasures.
- Scalability challenges in ARTs because of state-space explosion.
- The number of states are $2^{Leaf-Nodes}$.

ATTACK RESPONSE TREE





ATTACK COUNTERMEASURE TREE (ACT)

- ACTs use non-state space representation of attacks and countermeasures.
- Detection and mitigation mechanisms are defined not only at leaf nodes, but also at the intermediate nodes.
- ACT can be used to perform a probabilistic analysis of system risk, Return of Investment (ROI), attack impact, etc.

ACT FORMALISM

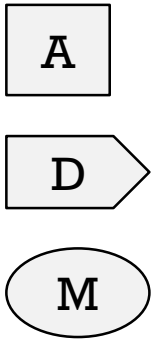
Symbol	Meaning
A_k	Attack Event
D_k	Detection Event
M_k	Mitigation Event
CM_k	Countermeasure
$ACT = \{V, \psi, E\}$	V: Vertices, ψ: Logical Gates, E: Edges
$\phi(X)$	Structure function of ACT
pA_k	Probability of occurrence of attack event A_k
pD_k	Probability of successful detection of attack event D_k
pM_k	Probability of successful mitigation of attack event M_k

ACT FORMALISM

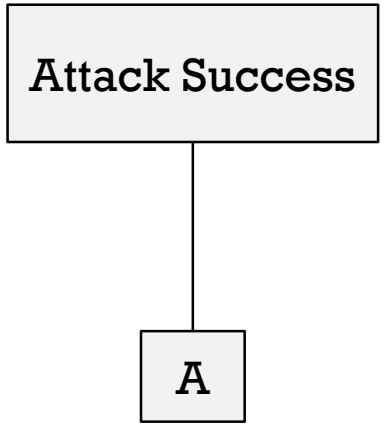
Symbol	Meaning
p_{goal}	Probability of attack success at the ACT goal
p_{UD}	Probability of undetected attack at the ACT goal
p_{DUM}	Probability of detected but unmitigated attack at ACT goal
$I_{A_k}^{ST}$	Structural importance measure of attack event A_k
$I_{A_k}^{B}$	Birnbaum importance measure of attack event A_k
i_{A_k}	Impact of attack event A_k
I_{goal}	Impact at goal node of ACT
cA_k	Cost of attack event A_k
$C_{Attacker}$	Attack cost at goal node of ACT
cCM_k	Security investment cost of countermeasure CM_k

ACT FORMALISM

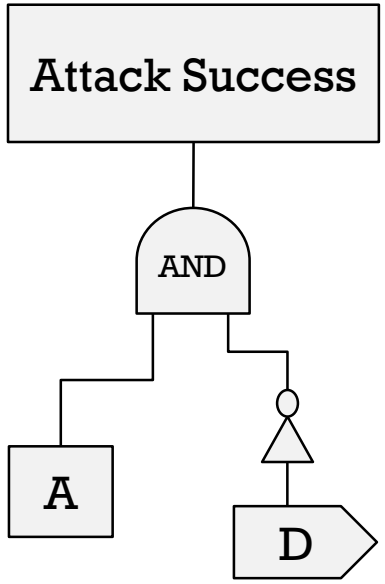
- Attack Event: Installation of key-logger.
- Detection Event: Detect keystroke logger.
- Mitigation Event: Removal of key-logger.



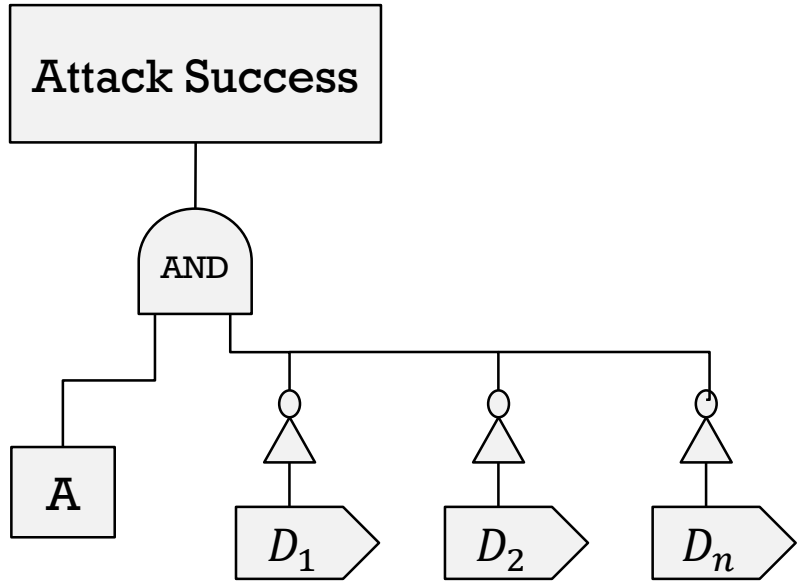
ACT EXAMPLE



$$p_{goal} = p_A$$

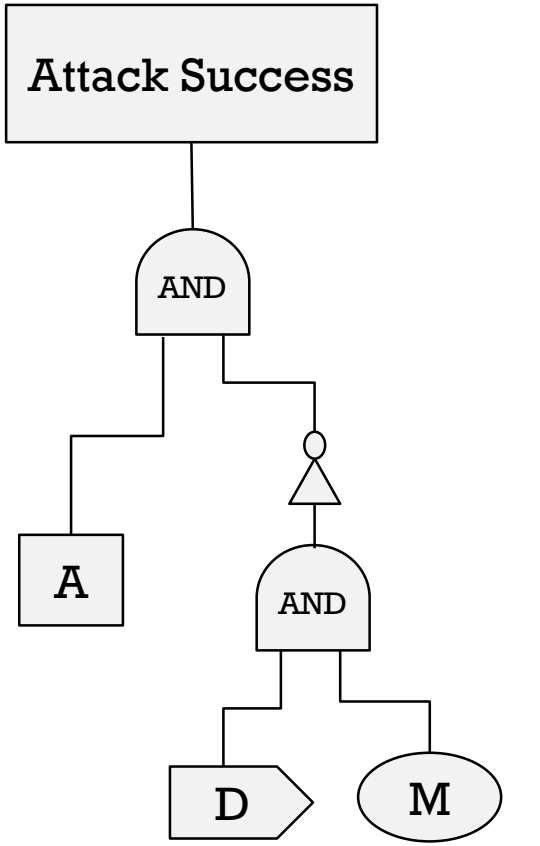


$$p_{goal} = p_A \times (1 - p_D)$$



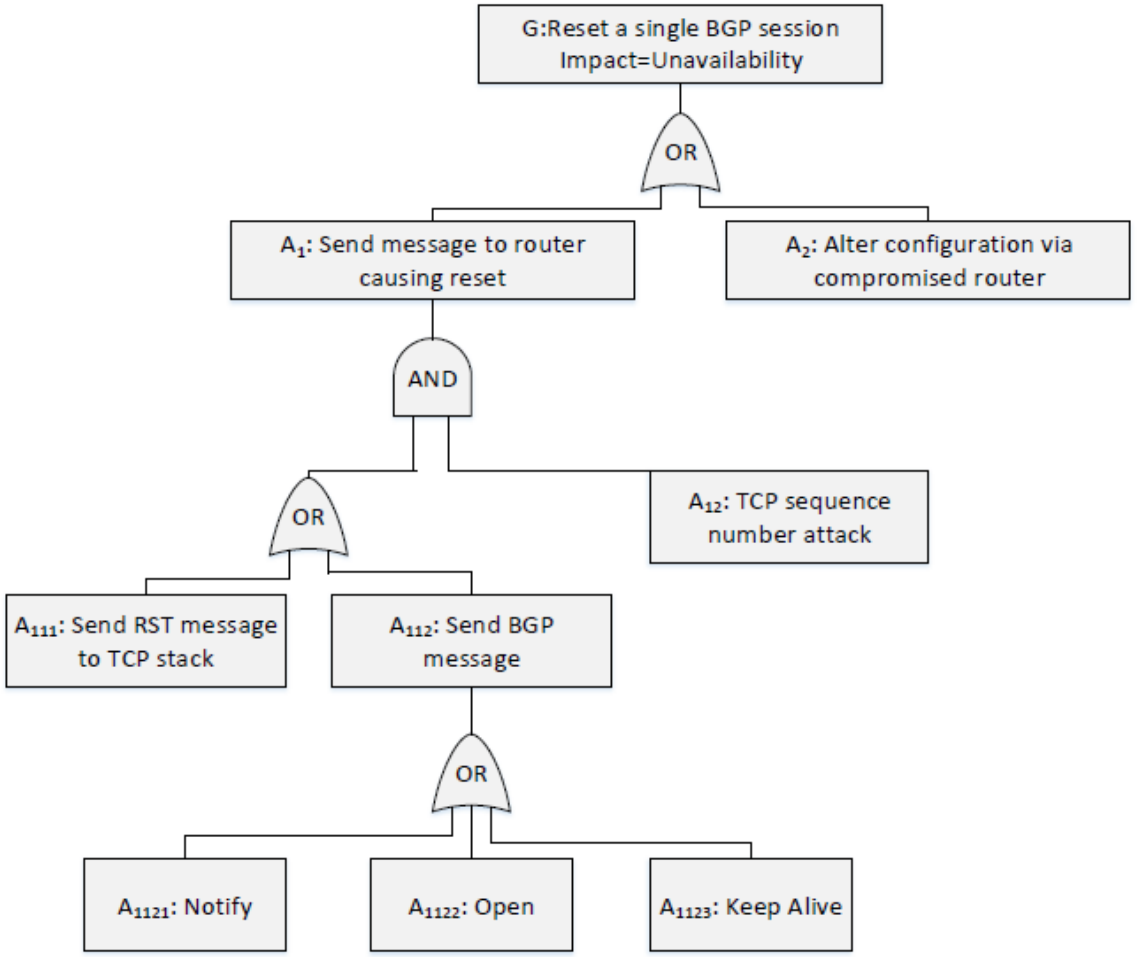
$$p_{goal} = p_A \times (1 - p_{D_1})(1 - p_{D_2})$$

ACT EXAMPLE

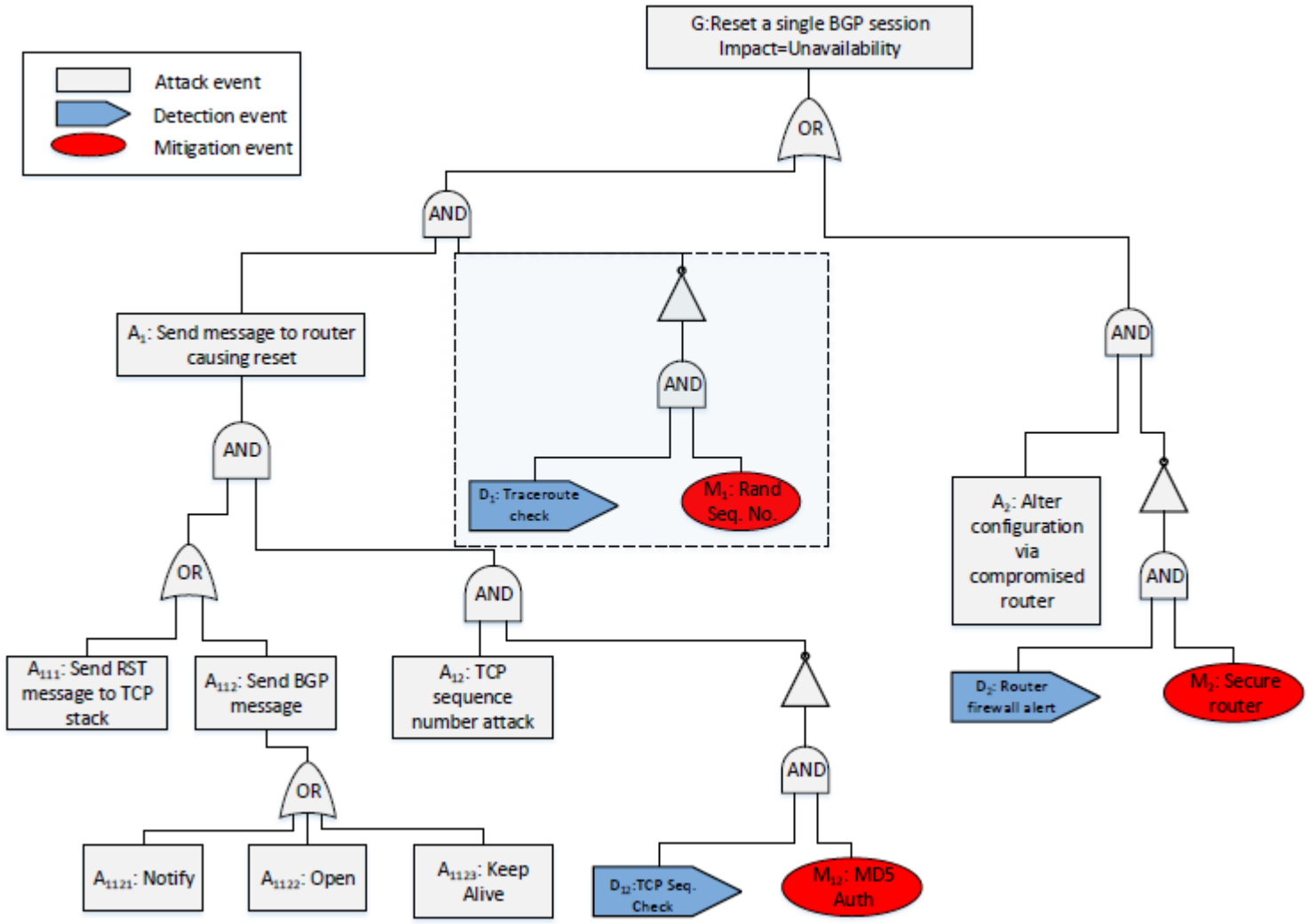


$$p_{goal} = p_A \times (1 - p_D + p_D(1 - p_M))$$

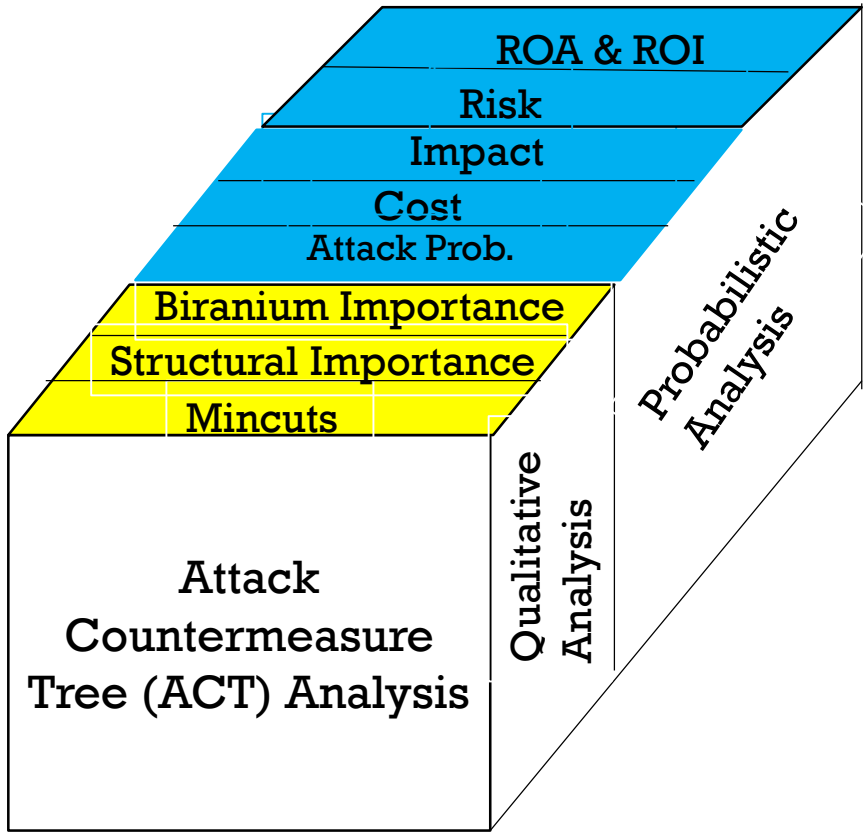
ATTACK TREE BGP REST EXAMPLE



ACT BGP RESET



ACT ANALYSIS



ACT ANALYSIS

- Qualitative Analysis (Metrics)

  1) Mincuts

- 2) Importance Measures
- Semi-quantitative analysis

  1) Some atomic attacks assigned value ‘1’, others ‘0’.

  2) Probabilities assigned to detection and mitigation events.

ACT ANALYSIS

- (Fully) Quantitative Analysis (Metrics)
  1) Probability of Attack
  2) Rate and distribution of time of attack
  3) Adversary viewpoint
     - Attack Cost $C_{attacker}$
     - Return on Attack (ROA)
  4) Defender's viewpoint
     - Attack Impact I_{goal}
     - Security cost
     - Return on Investment (ROI)

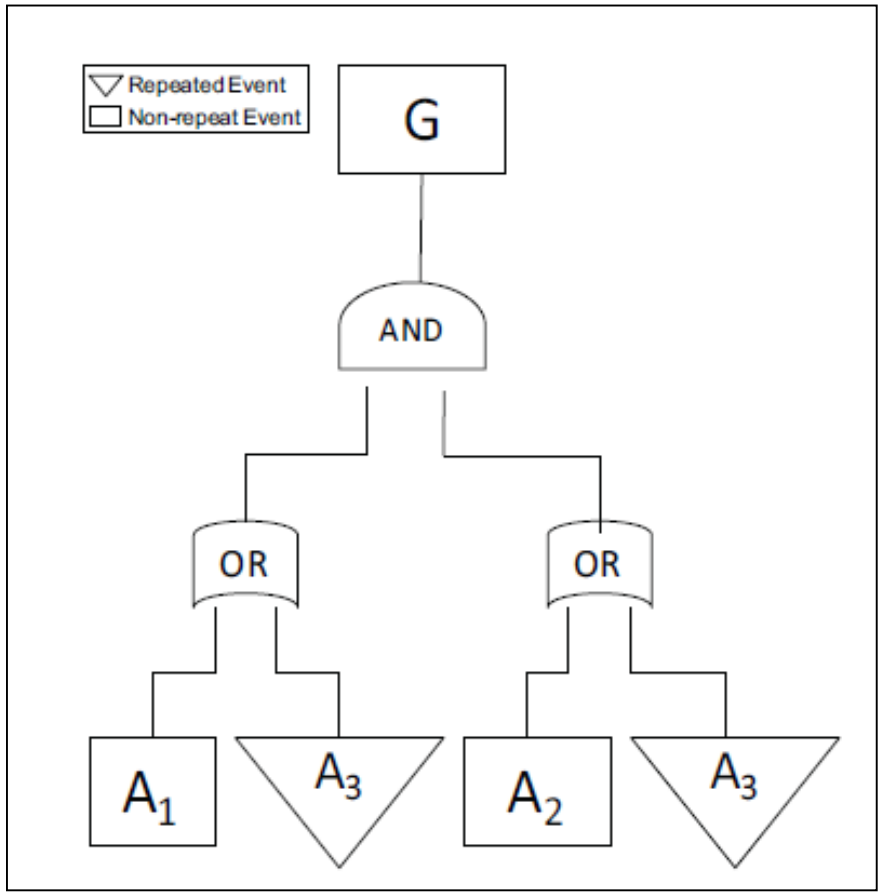
ACT ANALYSIS

Gate Type	Attack Success Prob.	Attack Cost	Impact
AND	$\Pi_{i=1}^{n} p(i)$	$\sum_{i=1}^{n} C_i$	$\sum_{i=1}^{n} I_i$
OR	$1 - \Pi_{i=1}^{n} (1 - p(i))$	$\forall i \min C_i$	$\forall i \max I_i$
k-of-n	$\sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}$	$\sum_{i=1}^{k} C_i$	$\sum_{i=1}^{k} I_i$

ACT ANALYSIS

- For k-of-n gate, it is assumed that cost values $(c_{A_1}, c_{A_2}, \ldots, c_{A_n})$ are sorted in ascending order of their costs.
- The impact values $(I_{A_1}, I_{A_2}, \ldots, I_{A_n})$ are sorted in descending order.
- Without above assumptions for k-of-n gate.
- Attack Cost: $\min_k \sum_{i=1} {C_{A_i}}^k$
- Attack Impact: $\max_k \sum_{i=1} {I_{A_i}}^k$

ATTACK TREE ANALYSIS: A SIMPLE AT



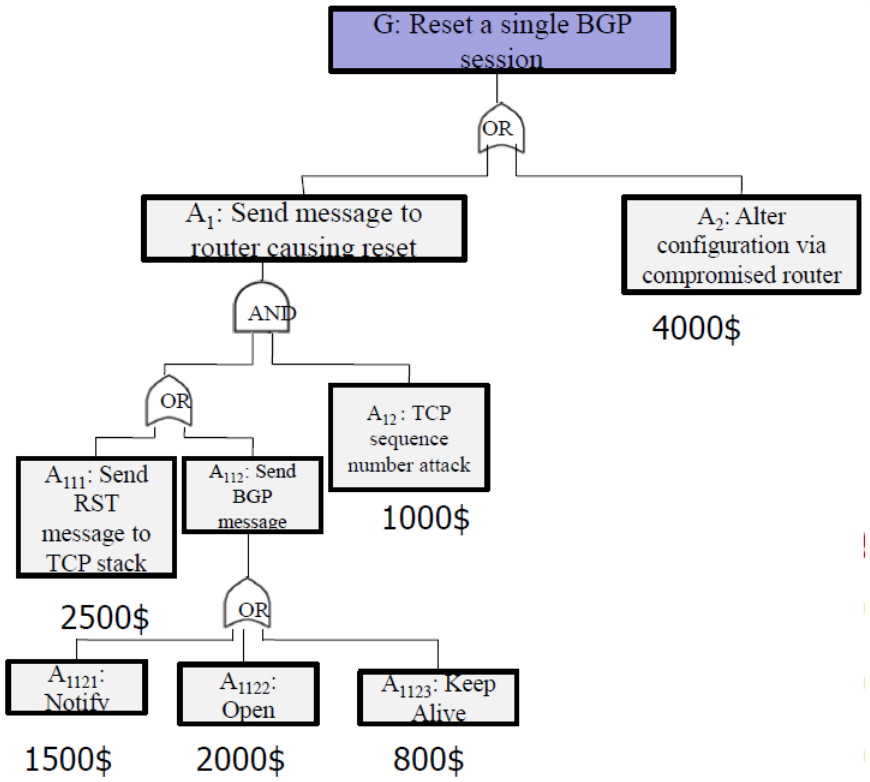
- Mincuts $\{A_1A_2, A_1A_3, A_2A_3, A_3\}$
- Attack Cost

$$C_{attacker} = \min\{C_{A_1} + C_{A_2}, C_{A_3}\}$$

- Attack Impact

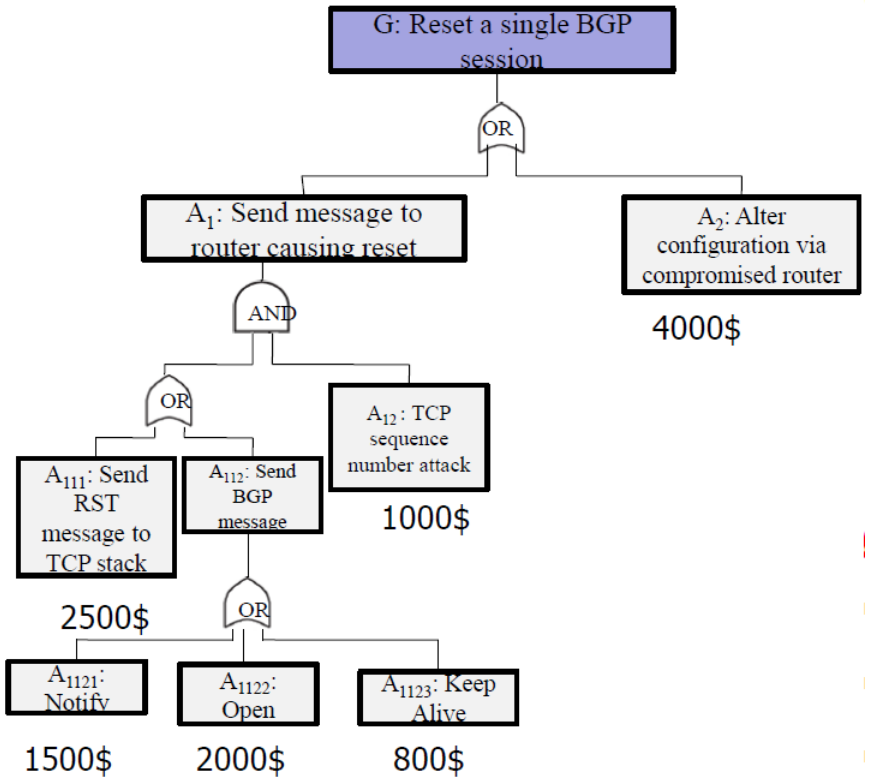
$$C_{goal} = max\{I_{A_1} + I_{A_2}, I_{A_1} + I_{A_3}, I_{A_2} + I_{A_3}, I_{A_3}\}.$$

ATTACK TREE ANALYSIS: ATTACK SCENARIOS



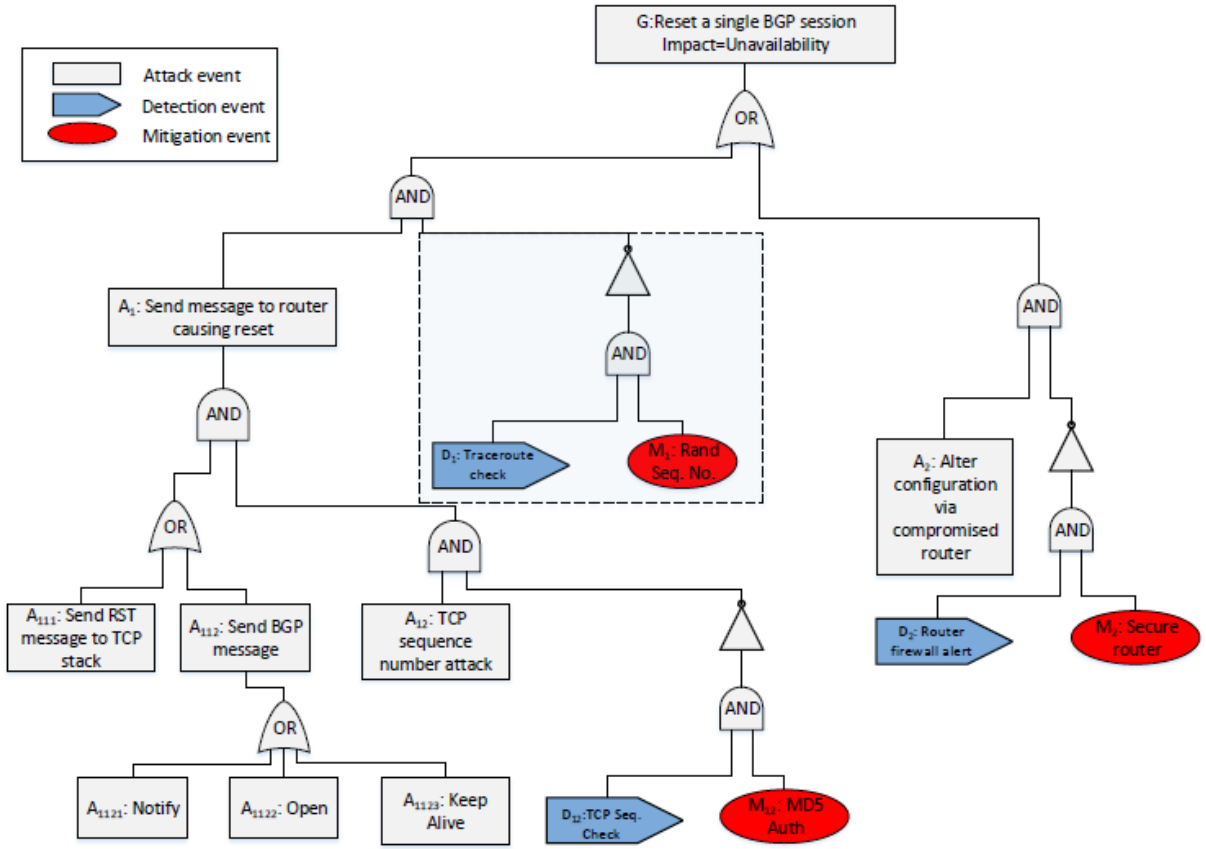
- All attack scenarios
  - $(A_{111}, A_{12})\ C_{attacker} = 3500\$$
  - $(A_{1121}, A_{12})\ C_{attacker} = 2500\$$
  - $(A_{1122}, A_{12})\ C_{attacker} = 3000\$$
  - $(A_{1123}, A_{12})\ C_{attacker} = 1800\$$
  - $(A_2)\ C_{attacker} = 4000\$$

ATTACK TREE ANALYSIS: ATTACK SCENARIOS



- Attack Cost Constraint - 3000$
- Only possible attack scenarios
  - (A_{1121}, A_{12})
  - (A_{1122}, A_{12})
  - (A_{1123}, A_{12})

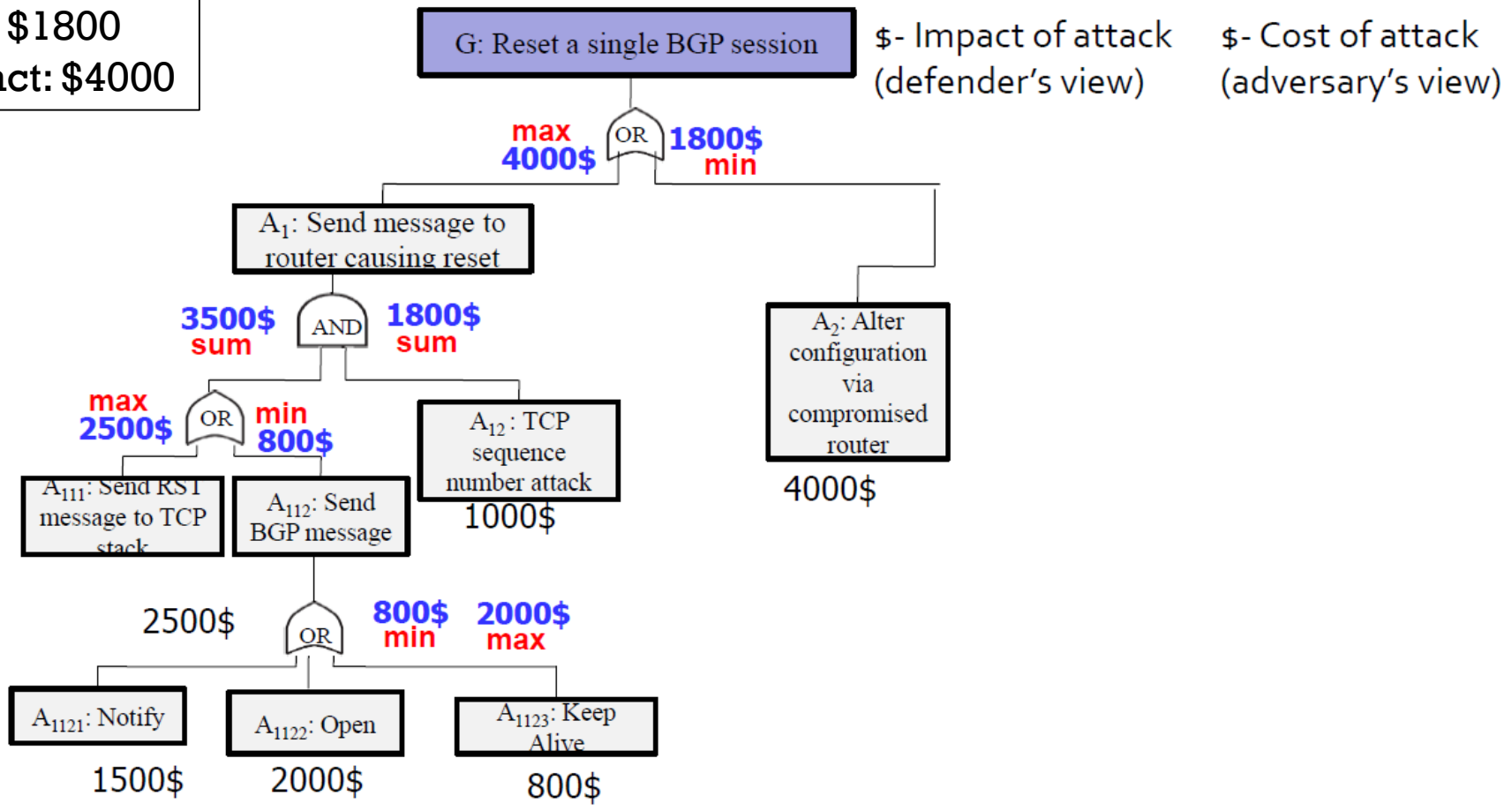
ACT ANALYSIS (QUALITATIVE)



- Mincuts of an attack tree (AT) represents attack scenarios.
- Mincuts of this ACT (represents attack countermeasure scenarios).
- $[(A_2)(D_2M_2)']$, $[(A_{1123})(D_1M_1)'(A_{12})(D_{12}M_{12})']$, $[(A_{1122})(D_1M_1)'(A_{12})(D_{12}M_{12})']$, $[(A_{1121})(D_1M_1)'(A_{12})(D_{12}M_{12})']$, $[(A_{111})(D_1M_1)'(A_{12})(D_{12}M_{12})']$
- Structural importance measures can also be calculated in ACT for sensitivity analysis.

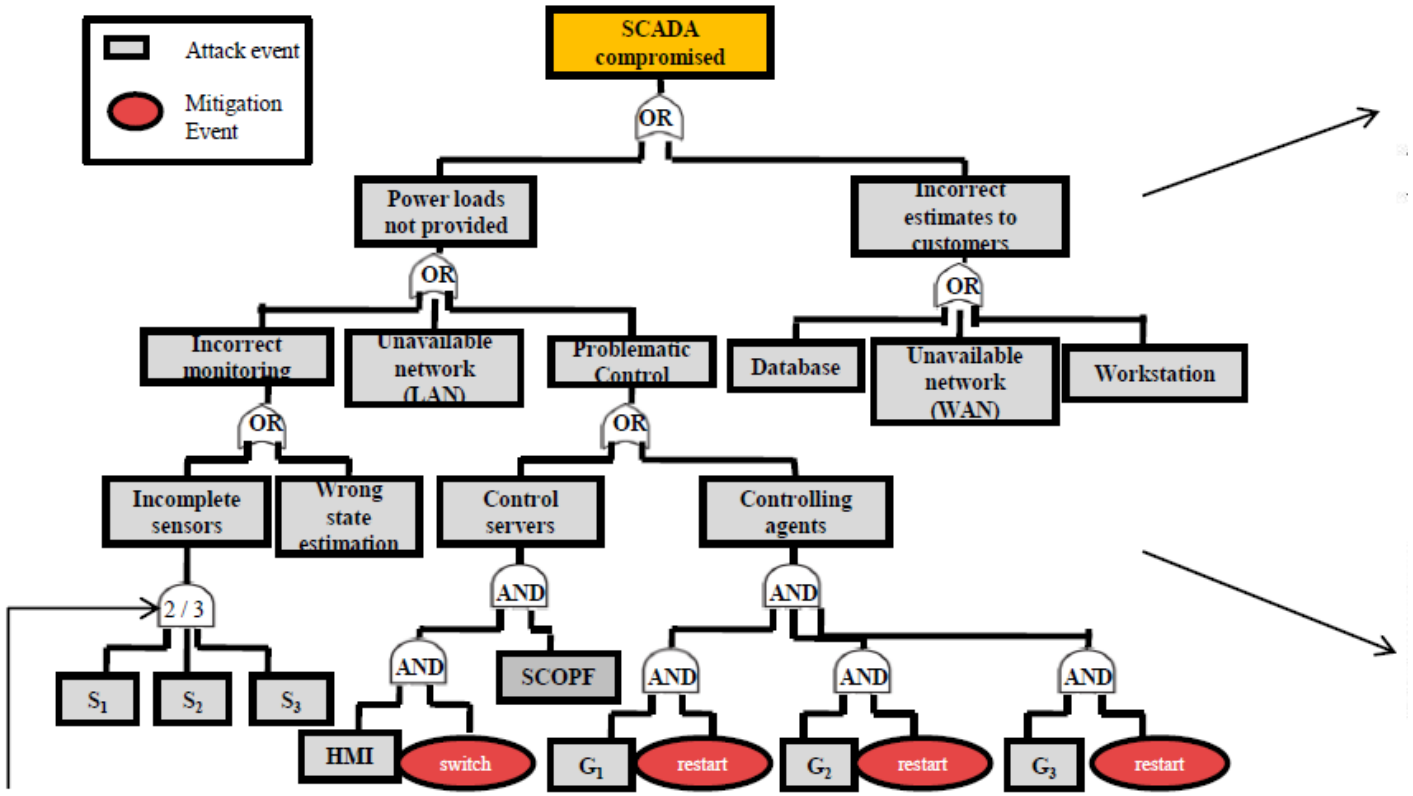
ACT ANALYSIS (QUANTITATIVE)

Attack cost: $1800
Attack impact: $4000



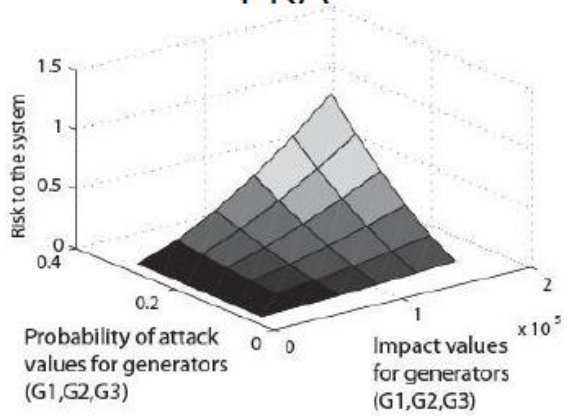
ACT ANALYSIS (QUANTITATIVE)

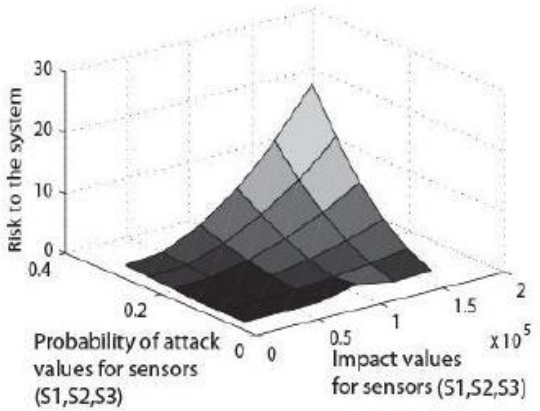
Risk = $Impact \times Prob.Attack = I_{goal} \times p_{goal}$



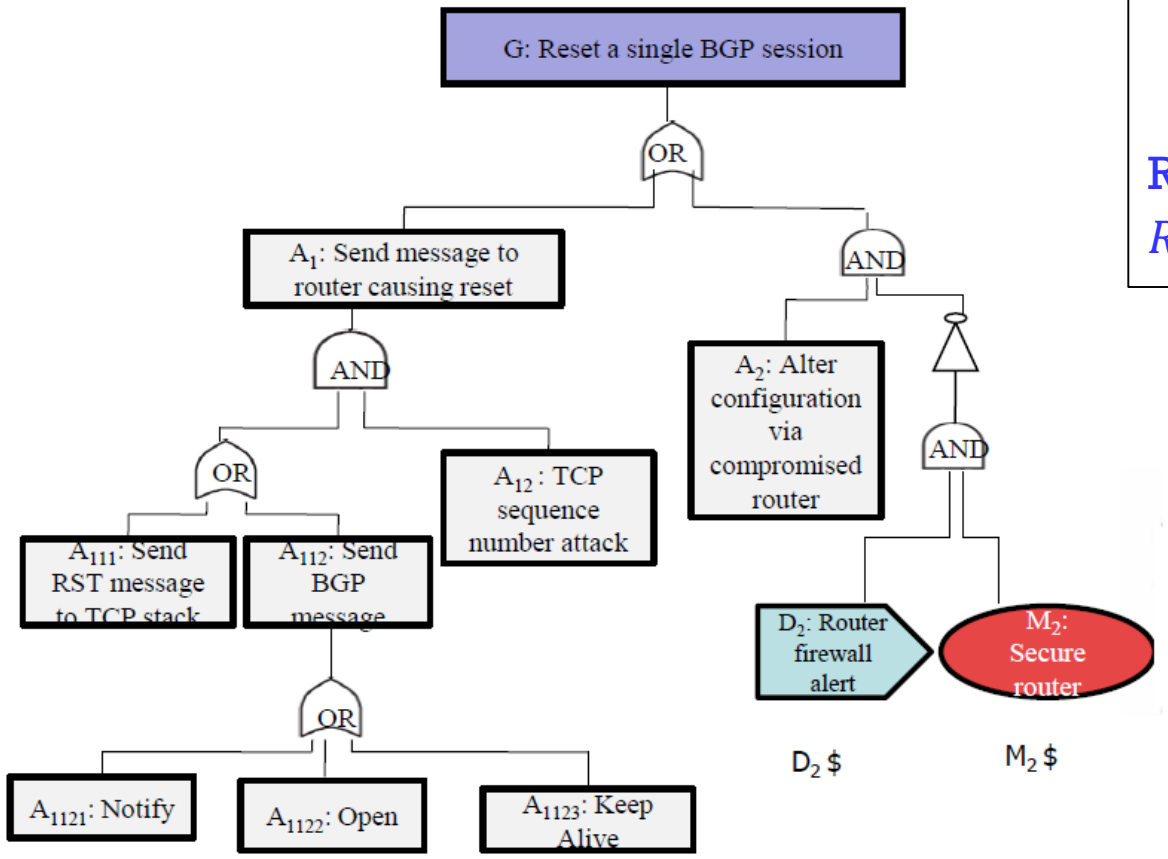
k-of-n gate

PRA





ACT ANALYSIS (QUANTITATIVE)

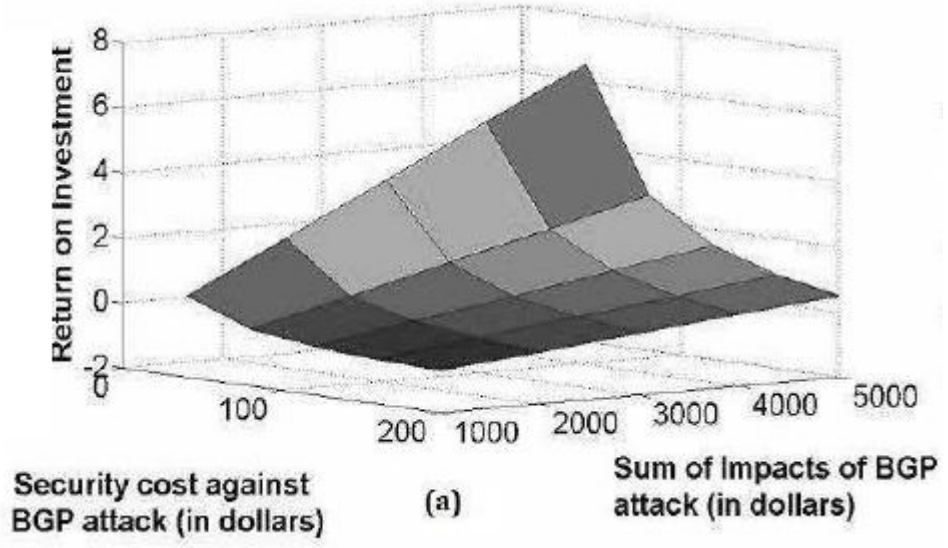


$$ROI = \frac{Investment\ Gain\ - Investment\ Cost}{Investment\ Cost}$$

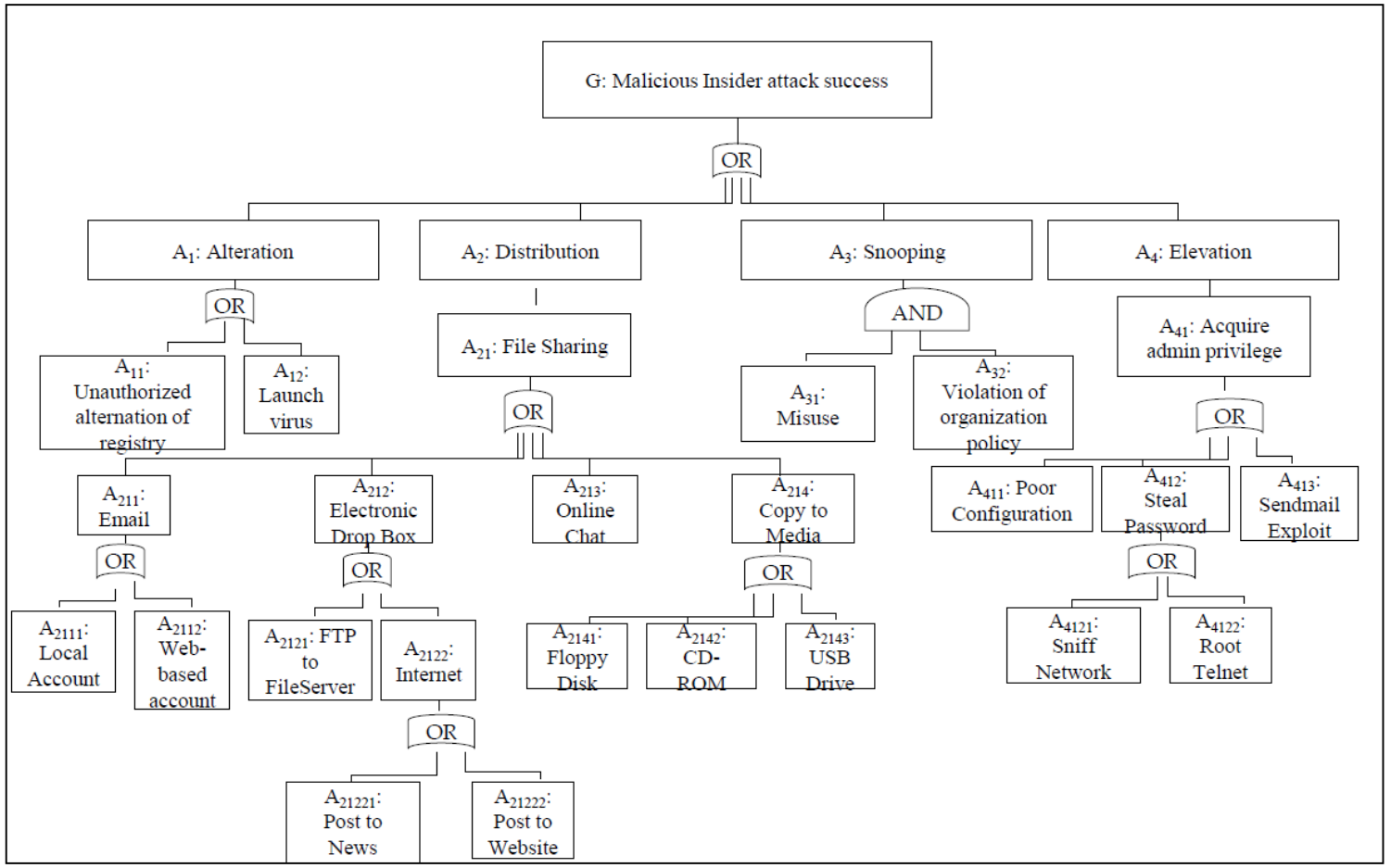
$$ROI = \frac{I_{goal} \times \Delta P_{goal_{CM_i}} - C_{CM_i}}{C_{CM_i}}$$

Range of $C_{CM_2} = (D_2M_2): 0 - 200\$$

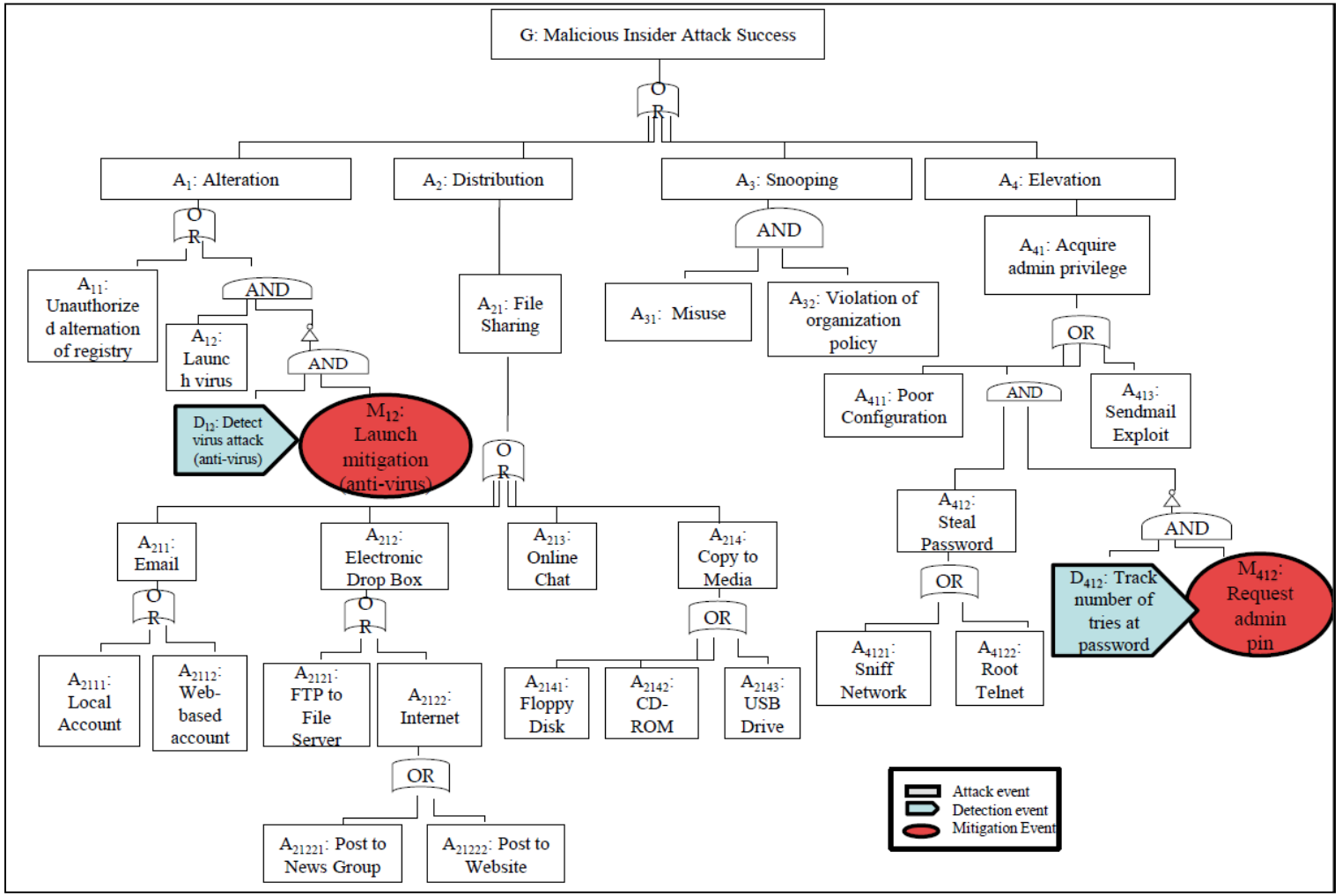
$ROI_{D_2M_2}$ (defender's view)



MALICIOUS INSIDER ATTACK TREE



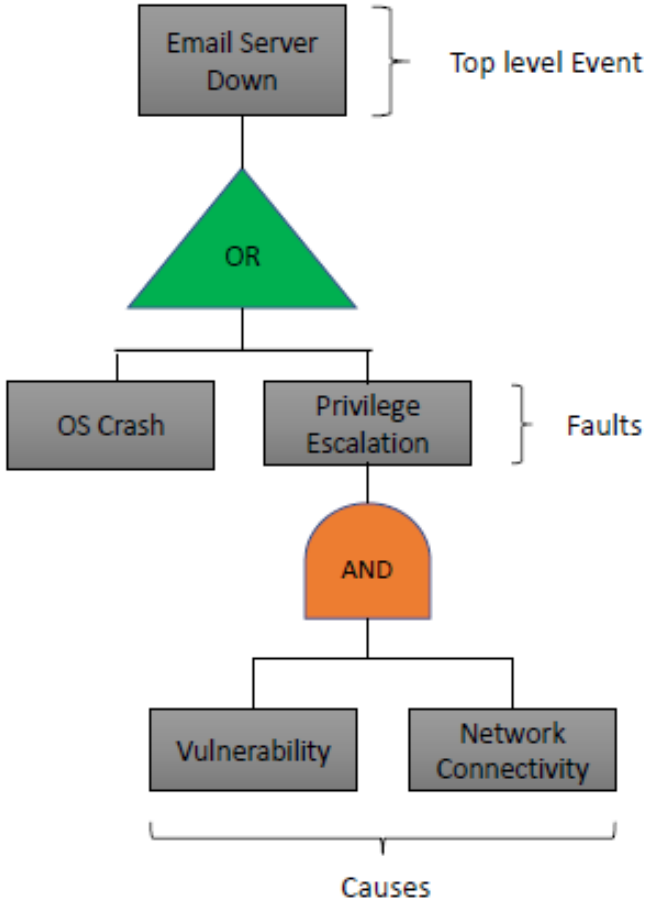
MALICIOUS INSIDER ACT



OTHER ATTACK REPRESENTATION METHODS

Fault Tree, Event Tree, Hierarchical Attack Representation Model (HARM)

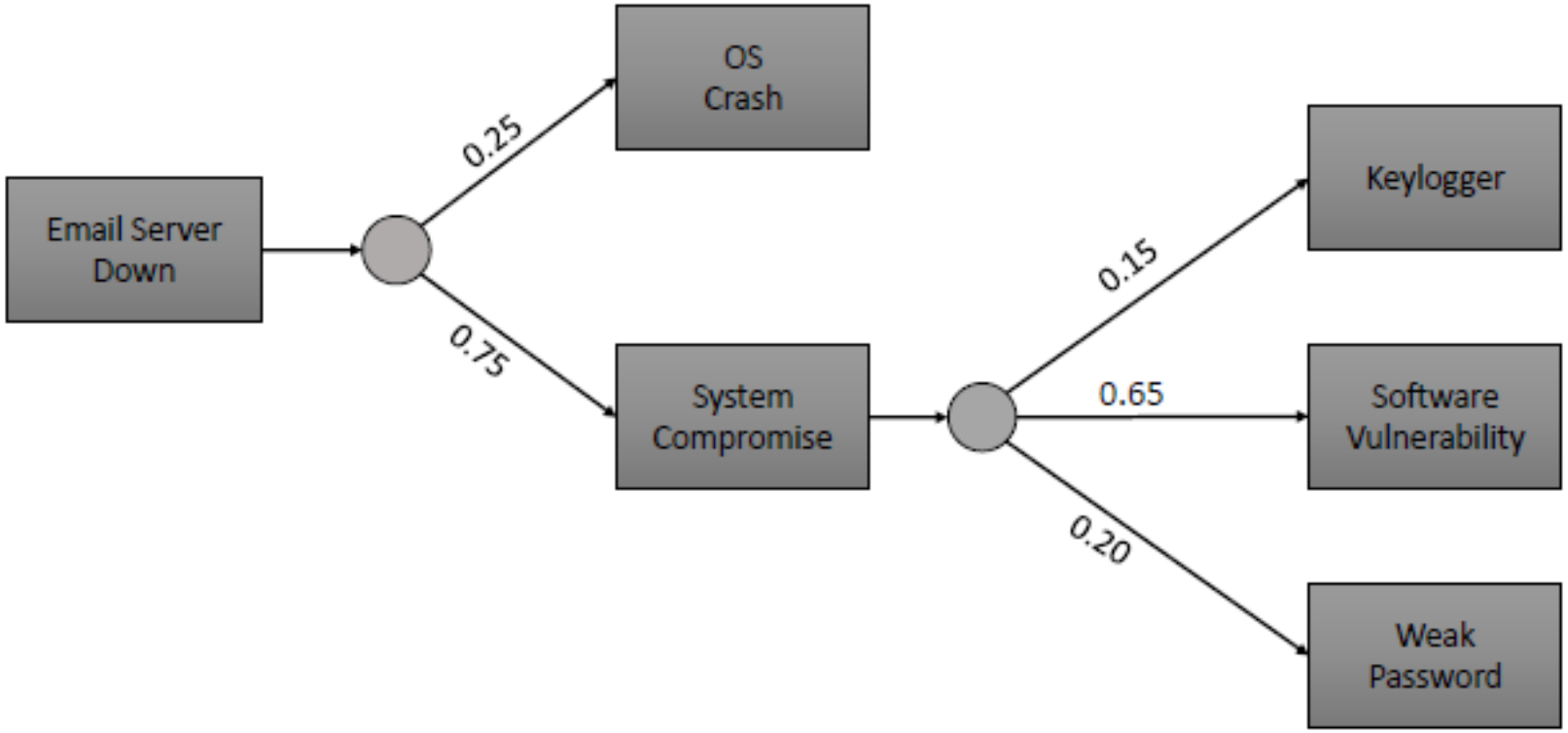
FAULT TREE


Email Server Down
Top level Event
OR
OS Crash
Privilege Escalation
Faults
AND
Vulnerability
Network Connectivity
Causes

FAULT TREE

- Analytic model used to check the state of the system.
- The root node of the tree is specified and system is analyzed for undesired operations and events with leaf nodes that contribute to the event.
- Fault Trees are linked by AND, OR logic gates.

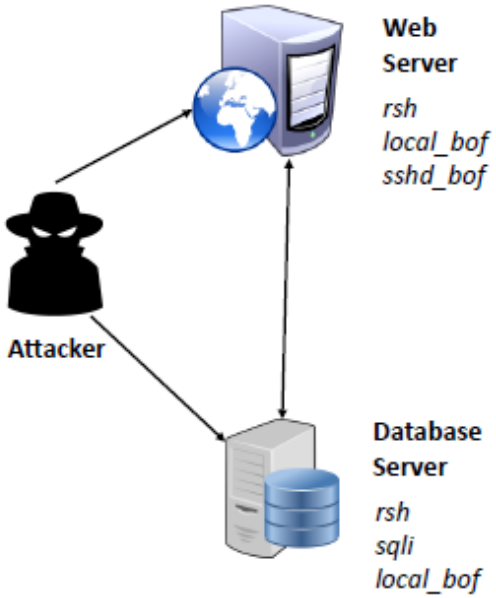
EVENT TREE


Email Server Down
0.25
OS Crash
0.75
System Compromise
0.15
Keylogger
0.65
Software Vulnerability
0.20
Weak Password

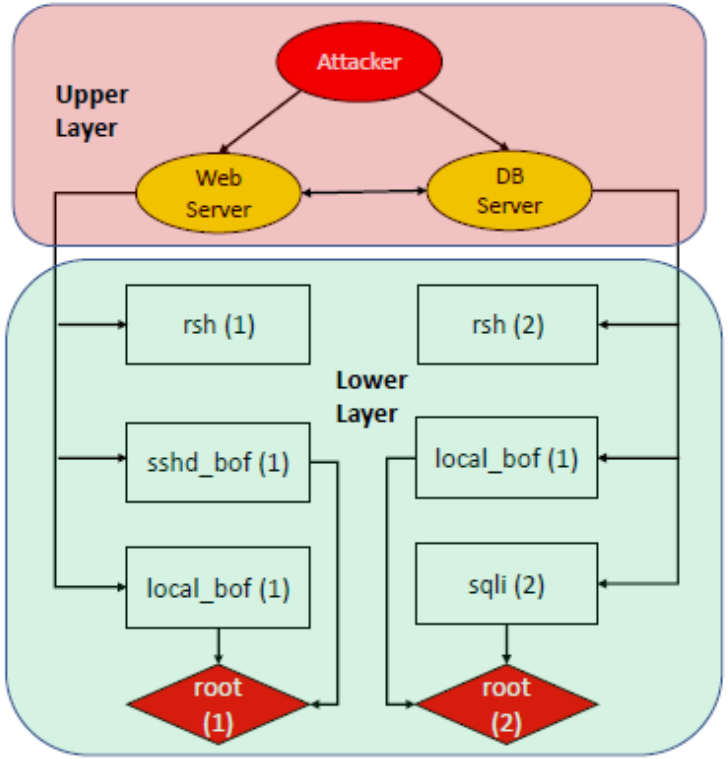
EVENT TREE

- Event Tree is used to identify the security violations in a network.
- Event Tree doesn't use the decision points requiring logical operators AND and OR.
- The probabilistic values of occurrence of various events are captured in the event tree.
- The cumulative and conditional probability values of occurrence of each event can be calculated at each level of the event tree.

HIERARCHICAL ATTACK REPRESENTATION MODEL (HARM)



a) Attack Scenario



b) HARM

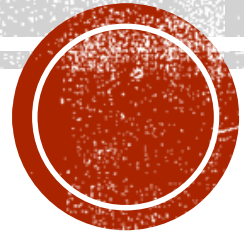
HIERARCHICAL ATTACK REPRESENTATION MODEL (HARM)

- Attack Graph and Tree based representation models have their limitations.
- Attack Graph suffers from state-space explosion issue.
- Attack Tree fails to capture the attack path information accurately.
- Hard to keep track of network topology and service vulnerability information using single attack representation method (ARM).

HIERARCHICAL ATTACK REPRESENTATION MODEL (HARM)

- HARM utilizes layered approach to improve scalability and expressiveness of ARMs.
- Upper layer represents network connectivity and lower layer represents network vulnerabilities.
- Security and performance analysis on each layer in HARM can be performed separately.

LIMITATIONS OF ATTACK REPRESENTATION METHODS



CHALLENGES IN RISK ANALYSIS

- How to calculate probabilities in an attack graph with shared dependencies and cycles.
- Bayesian Network – *Frigault, et al, 2008*.
- Assuming Independence among attack paths – *Wang, et al., 2008*.
- Customized data-flow algorithm with dynamic programming *Homer, et al., 2009*.

SCALABILITY ISSUES IN ARM

- Formal Models – Amman et al. face state-space explosion problem.
- MulVAL has complexity $O(N^2) - O(N^3)$.
- MP-Graph Ingols et al. scales poorly on large network size.
- Scalable attack graph generation methods is an active area of research.

COMPLEXITY OF ATTACK GRAPH GENERATION TOOLS

Name	Developers	open source	Accessible	Attack Graph Type	Scalability	Intuitive level
Attack Graph Toolkit	Carnegie Mellon University	Yes	Free	State graph	Poor, construction time exponentially	Good. vertices are state node, edges are the state transition.
MulVAL	Kansas State University	Yes	Free	Logical attack graph (attribute attack graph)	Polynomial: $O(N^2) \sim O(N^4)$	Good
TVA	George Mason University	No	not open, difficult to obtain	Penetration dependency graph, aggregation attack graph	Polynomial: $O(N^2)$	Better. Vertex is a host or host group
Cauldron	PROINFO Company, George Mason University	Commercial Software	pay	Penetration dependency graph, aggregation attack graph	Polynomial: $O(N^2)$	Better. Vertex is a host or host group.
NetSPA	Massachusetts Institute of Technology	No	not open, difficult to obtain	MP (Multiple-Prerequisite) graph	O(N lgN)	Good
FireMon	FireMon Corporate, Massachusetts Institute of Technology	Commercial Software	pay	MP (Multiple-Prerequisite) graph	O(N lgN)	Better
Skybox View	Skybox Security, Inc.	Commercial Software	pay	unknown	Polynomial $O(N^2)$	Better. Vertex is the host.

REFERENCES

- https://pdfs.semanticscholar.org/4fcb/5b590326ad63eaadb04d5c0f24eacc5fff67.pdf
- https://www.usenix.org/legacy/events/sec05/tech/full_papers/ou/ou.pdf
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6419708

CITE THIS WORK

@book{huang2018software,
title={Software-Defined Networking and Security: From Theory to Practice},
author={Huang, Dijiang and Chowdhary, Ankur and Pisharody, Sandeep},
year={2018},
publisher={CRC Press}}