Quantifying the Security Effectiveness of Firewalls and DMZs

Huashan Chen[1], Jin-Hee Cho[2], Shouhuai Xu[1]

[1]The University of Texas at San Antonio
[2]US Army Research Lab

HotSoS 2018

Outline

❑ Introduction

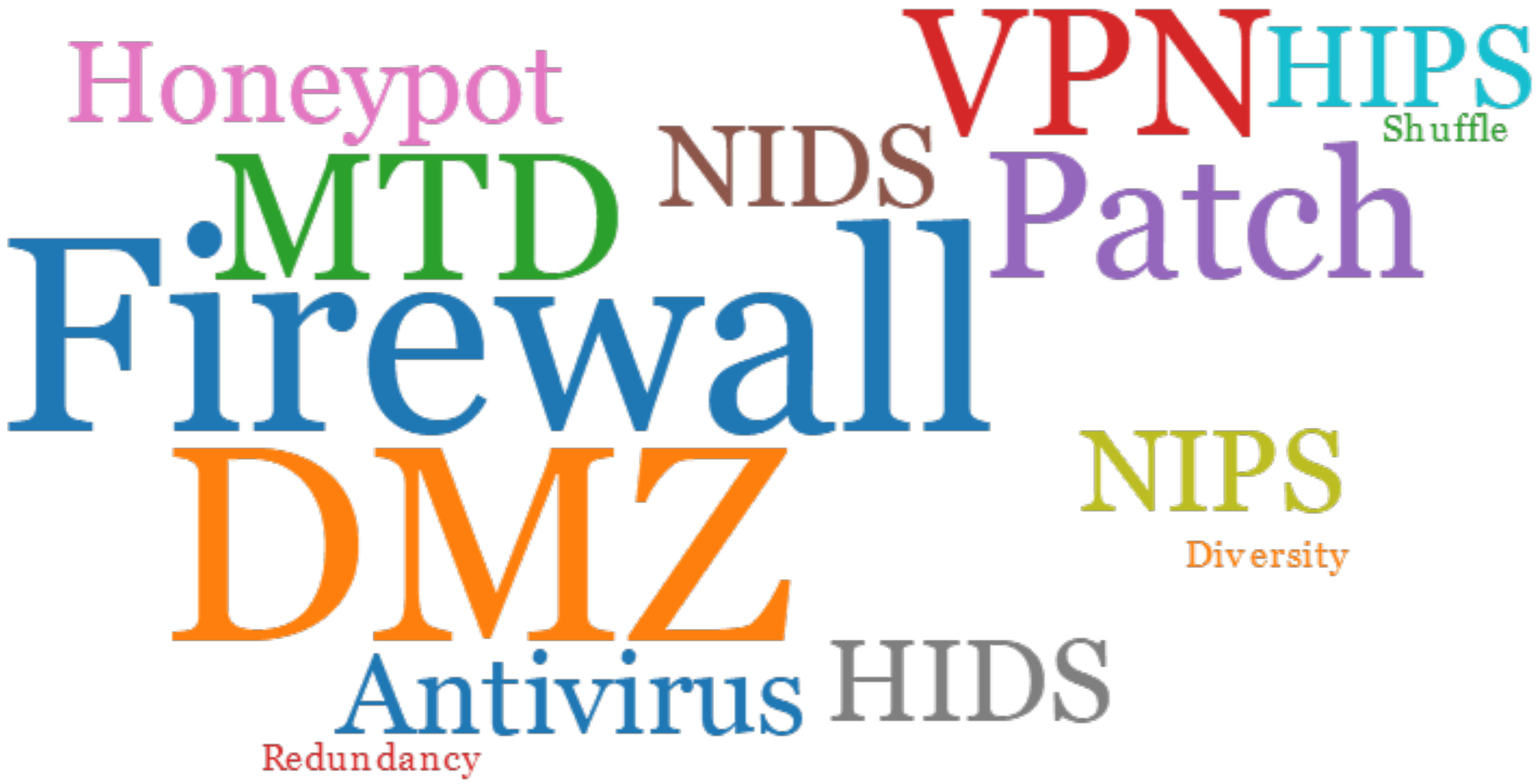
❑ A systematic framework

❑ Simulation experiments & results

❑ Related work

❑ Conclusion

The Problem: Quantitative Analysis of Security Mechanisms in *Networked Systems*



□ One of the most fundamental open problems, and remains open.

□ Very few (even early stage) results: extremely difficult in both modeling and analysis.

□ But, we have to tackle it!

Cybersecurity Dynamics [Xu HotSoS 2014]:
A Framework for Modeling and Analyzing Cybersecurity

□ Using *attack-defense* structure to capture the (attacker, victim) relation.

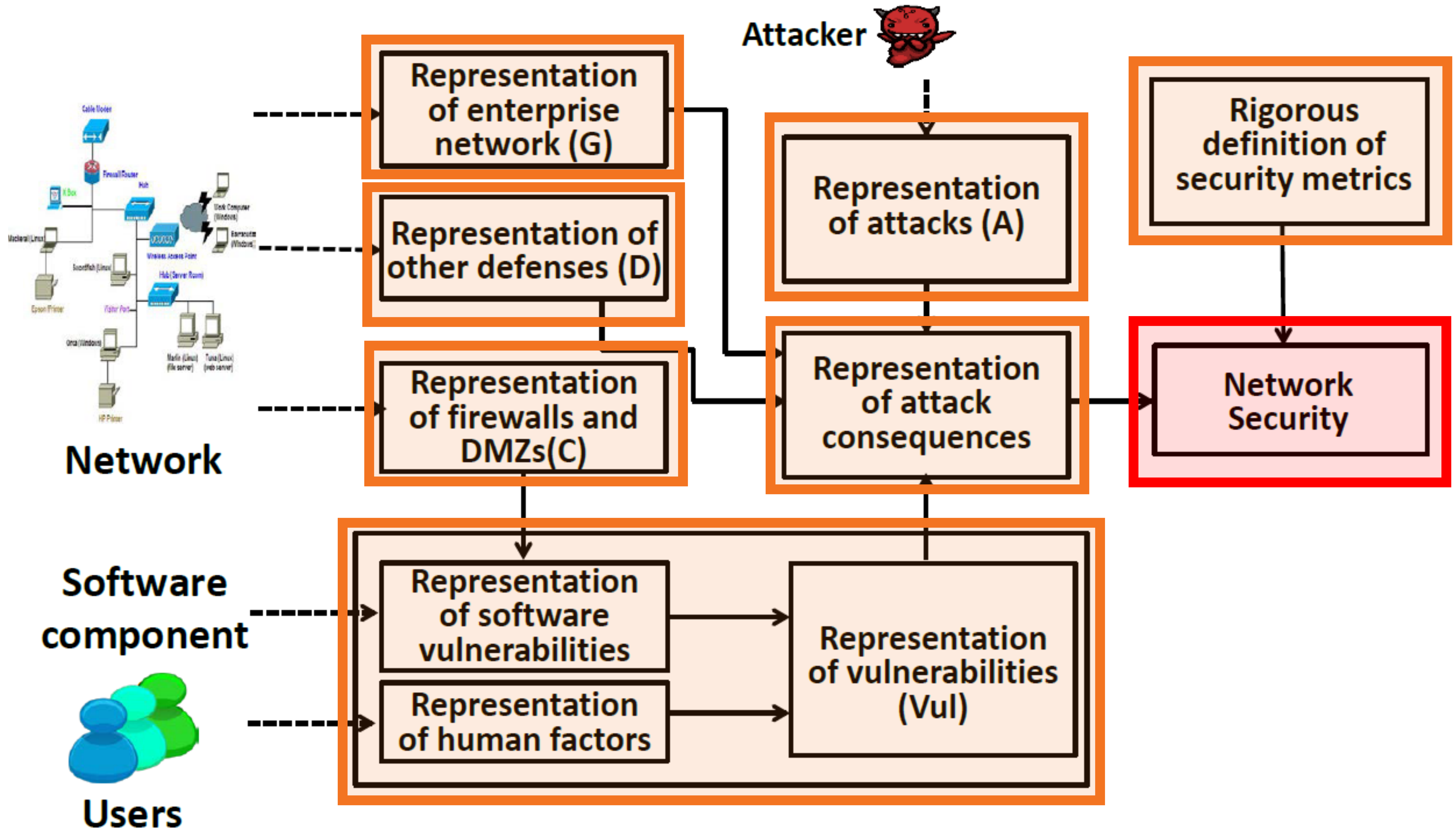
□ Using *parameters* to capture attack and defense capabilities, software vulnerabilities, etc.

□ Using evolution of *global security state* to describe the outcome of attack-defense interactions.

Our Contributions

- □ A systematic, *fine-grained* framework for modeling firewalls and DMZs by treating an entire enterprise network as a whole.
  - ◆ Fine-grained: Treating individual applications and operating system functions as “atomic” entities.
  - ◆ Dependence: No *independence* assumption between the attack events.
  - ◆ Realistic threat model: Accommodating realistic, APT-like attacks.
- □ A set of security metrics that can be objectively evaluated.
- □ A simulation system for evaluating security gain of firewalls and DMZs.

The Framework



Attacker

Network

Software component

Users

Representation of enterprise network (G)

Representation of other defenses (D)

Representation of firewalls and DMZs(C)

Representation of attacks (A)

Representation of attack consequences

Rigorous definition of security metrics

Network Security

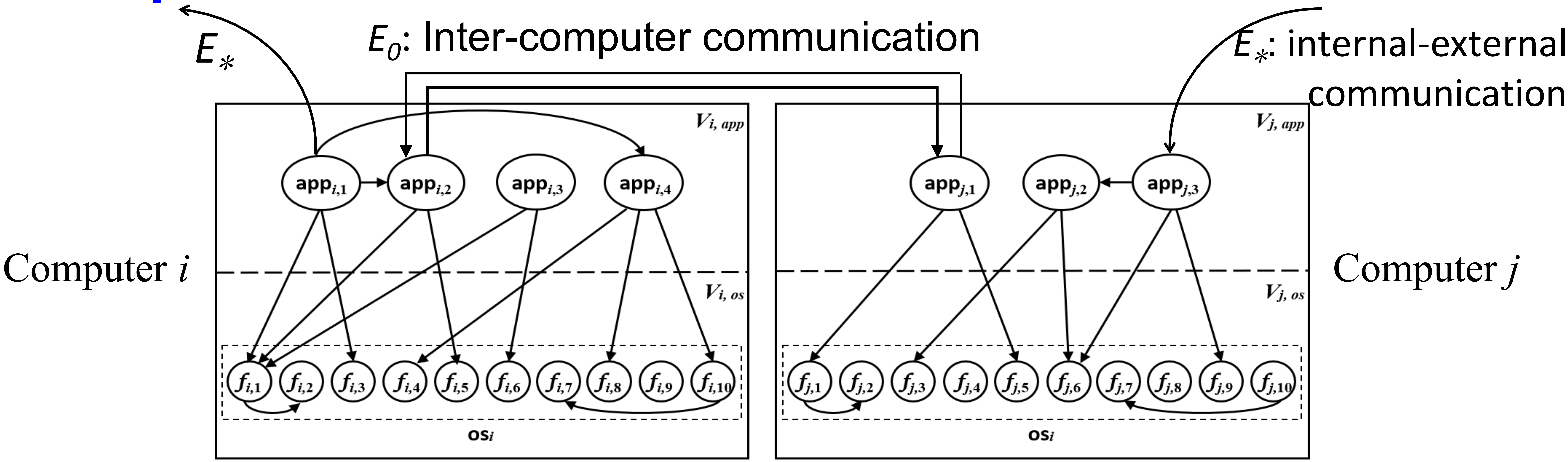
Representation of software vulnerabilities

Representation of human factors

Representation of vulnerabilities (Vul)

Legend: ------> Abstraction ⟶ Control / instruction flow

Representation of Networks in the Framework



E_*
E_0: Inter-computer communication
E_*: internal-external communication
$V_{i,\,app}$
$V_{i,\,os}$
$V_{j,\,app}$
$V_{j,\,os}$
Computer i
Computer j
app$_{i,1}$
app$_{i,2}$
app$_{i,3}$
app$_{i,4}$
app$_{j,1}$
app$_{j,2}$
app$_{j,3}$
$f_{i,1}$ $f_{i,2}$ $f_{i,3}$ $f_{i,4}$ $f_{i,5}$ $f_{i,6}$ $f_{i,7}$ $f_{i,8}$ $f_{i,9}$ $f_{i,10}$
$f_{j,1}$ $f_{j,2}$ $f_{j,3}$ $f_{j,4}$ $f_{j,5}$ $f_{j,6}$ $f_{j,7}$ $f_{j,8}$ $f_{j,9}$ $f_{j,10}$
OS$_i$
OS$_i$

□ **$G_i = (V_i, E_i)$: represents a computer**

Node set V_i: applications, OS functions

Arc set E_i: app-app communication, app-func, func-func dependency

□ **$G = (V, E)$: represents a network**

V = {app} ∪ {OS functions}, $E = E_1 \cup \ldots \cup E_n \cup E_0 \cup E_*$

Representation of Vulnerabilities in the Framework

□ Software Vulnerabilities

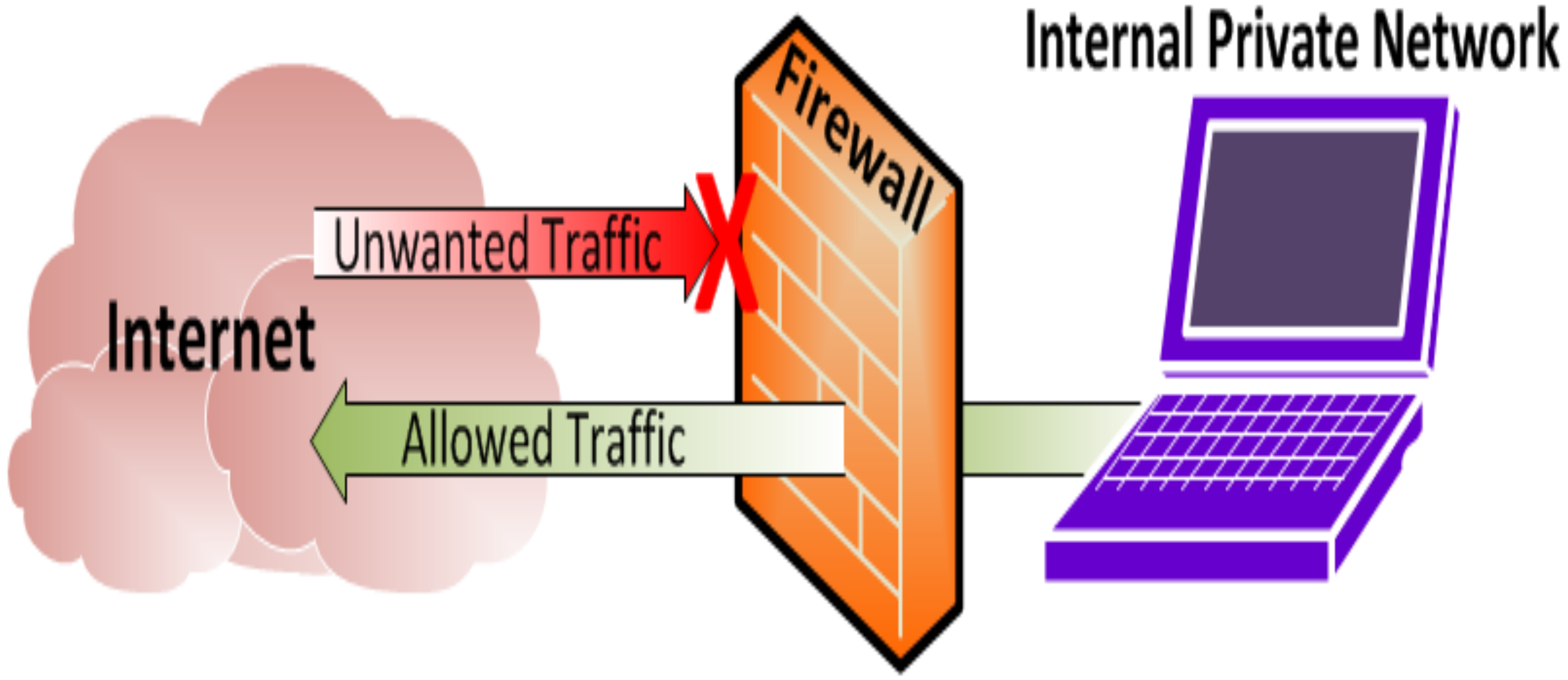
- ◆Access required (loc):
  - loc(vul)=0: require local access
  - loc(vul)=1: otherwise
- ◆Zero-day (zd):
  - zd(vul)=0: known
  - zd(vul)=1: zero-day
- ◆Privilege escalation (priv):
  - priv(vul)=0: user
  - priv(vul)=1: root

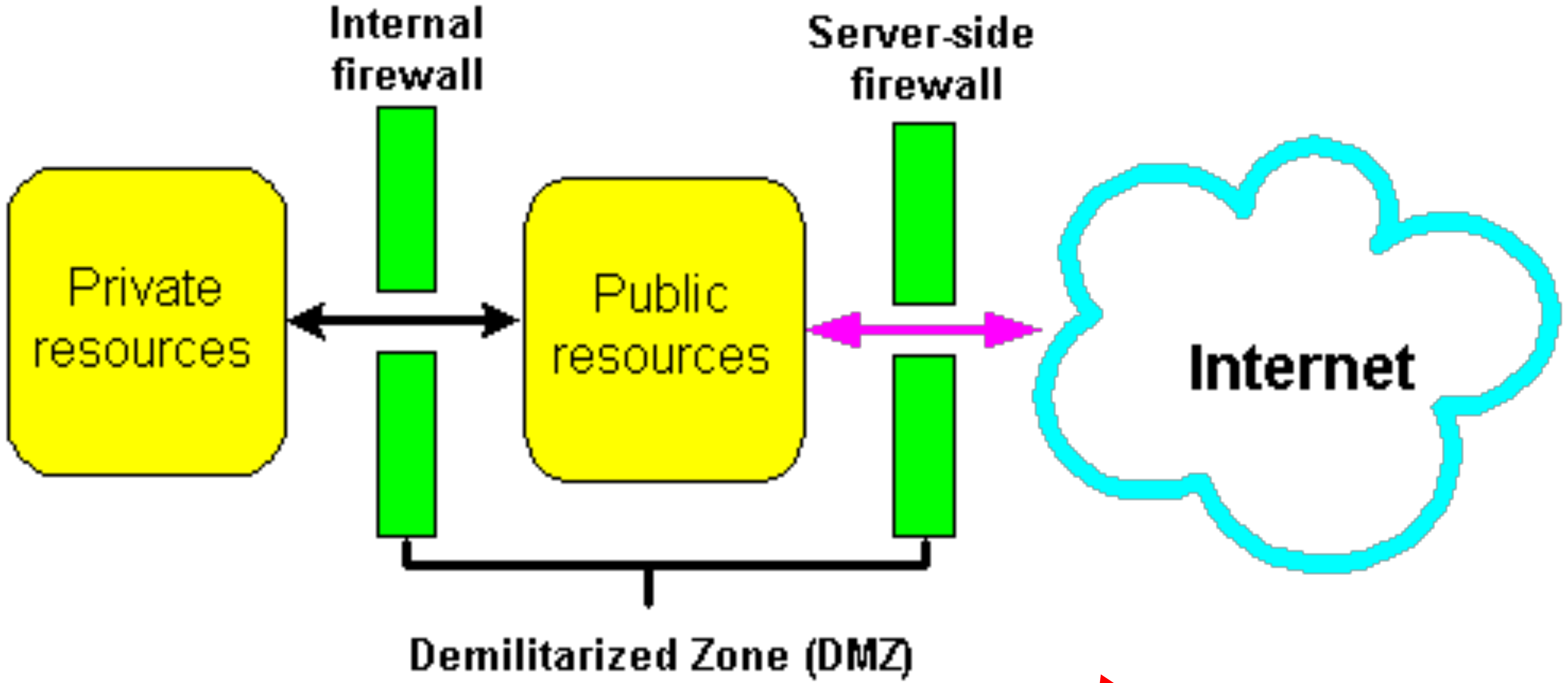
□ Human Vulnerabilities

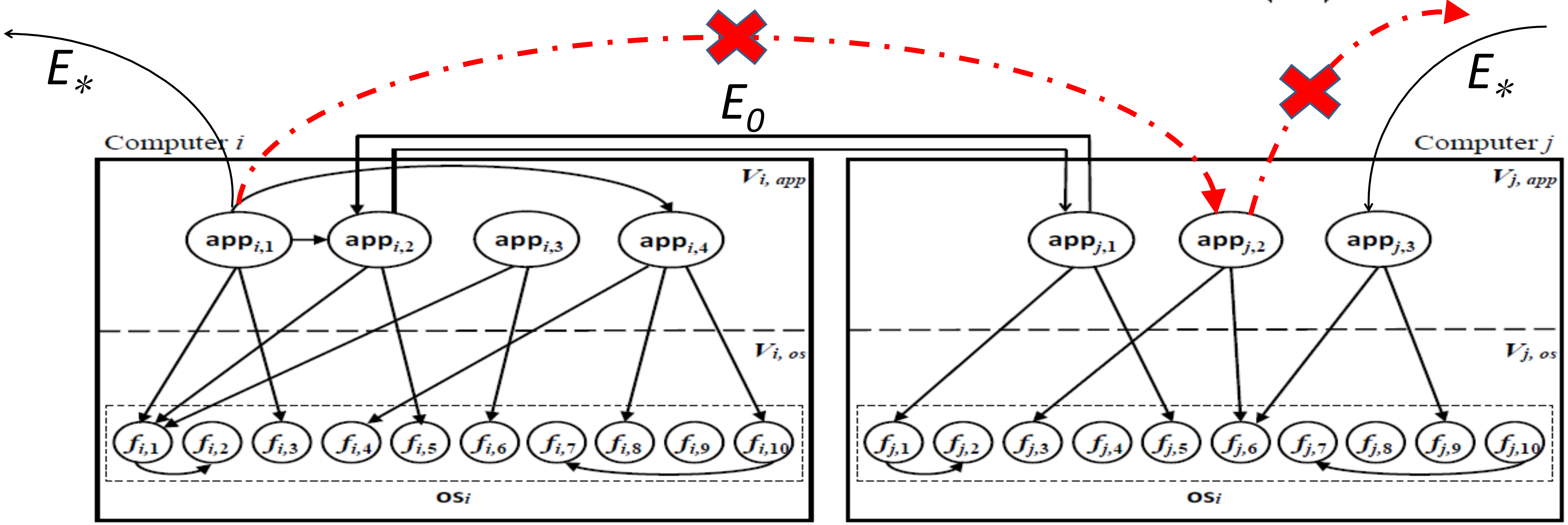
- ◆Probability a user is vulnerable to social engineering attack

$$\psi : V \rightarrow [0, 1]$$

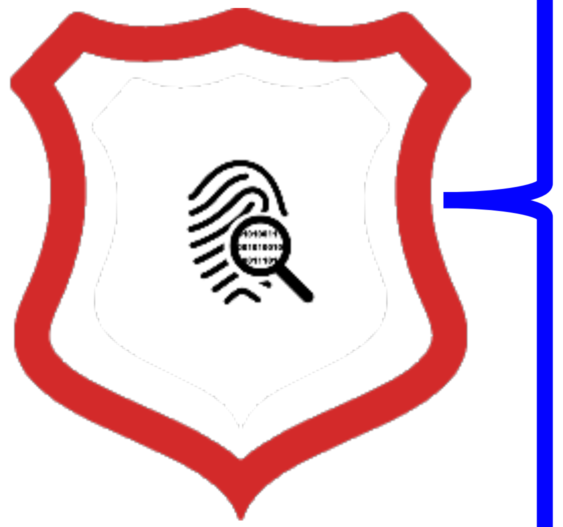
Representation of Firewalls and DMZs in the Framework






E_*
E_0
E_*
Computer i
Computer j
$V_{i,\,app}$
$V_{i,\,os}$
$V_{j,\,app}$
$V_{j,\,os}$
app$_{i,1}$
app$_{i,2}$
app$_{i,3}$
app$_{i,4}$
app$_{j,1}$
app$_{j,2}$
app$_{j,3}$
$f_{i,1}$ $f_{i,2}$ $f_{i,3}$ $f_{i,4}$ $f_{i,5}$ $f_{i,6}$ $f_{i,7}$ $f_{i,8}$ $f_{i,9}$ $f_{i,10}$
$f_{j,1}$ $f_{j,2}$ $f_{j,3}$ $f_{j,4}$ $f_{j,5}$ $f_{j,6}$ $f_{j,7}$ $f_{j,8}$ $f_{j,9}$ $f_{j,10}$
OS$_i$
OS$_i$

Representation of Other Defenses in the Framework

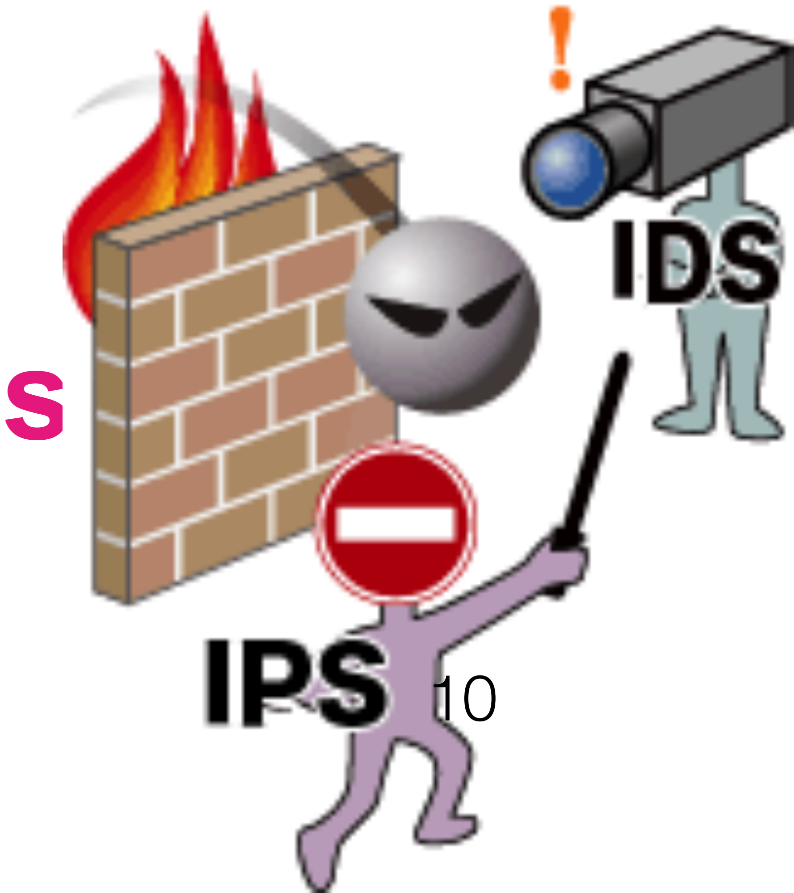


Host-based IPS

- ◆Policy
  - Tight: enforce strict preventive defense (e.g., whitelist)
  - Loose: do not enforce strict preventive defense
- ◆Capability
  - ζ : probability in blocking privilege escalation
  - α : probability in blocking other attacks

Network-based IPS

- ◆Capability: Blocking k fraction of inter-computer attacks

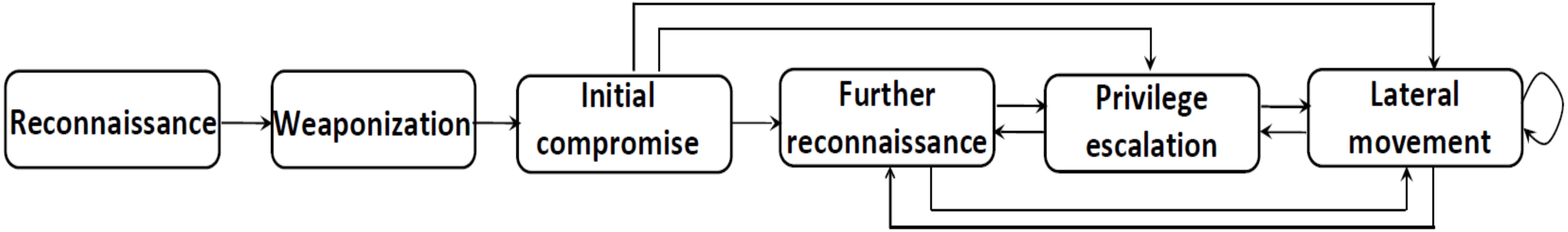


Representation of Attacks in the Framework

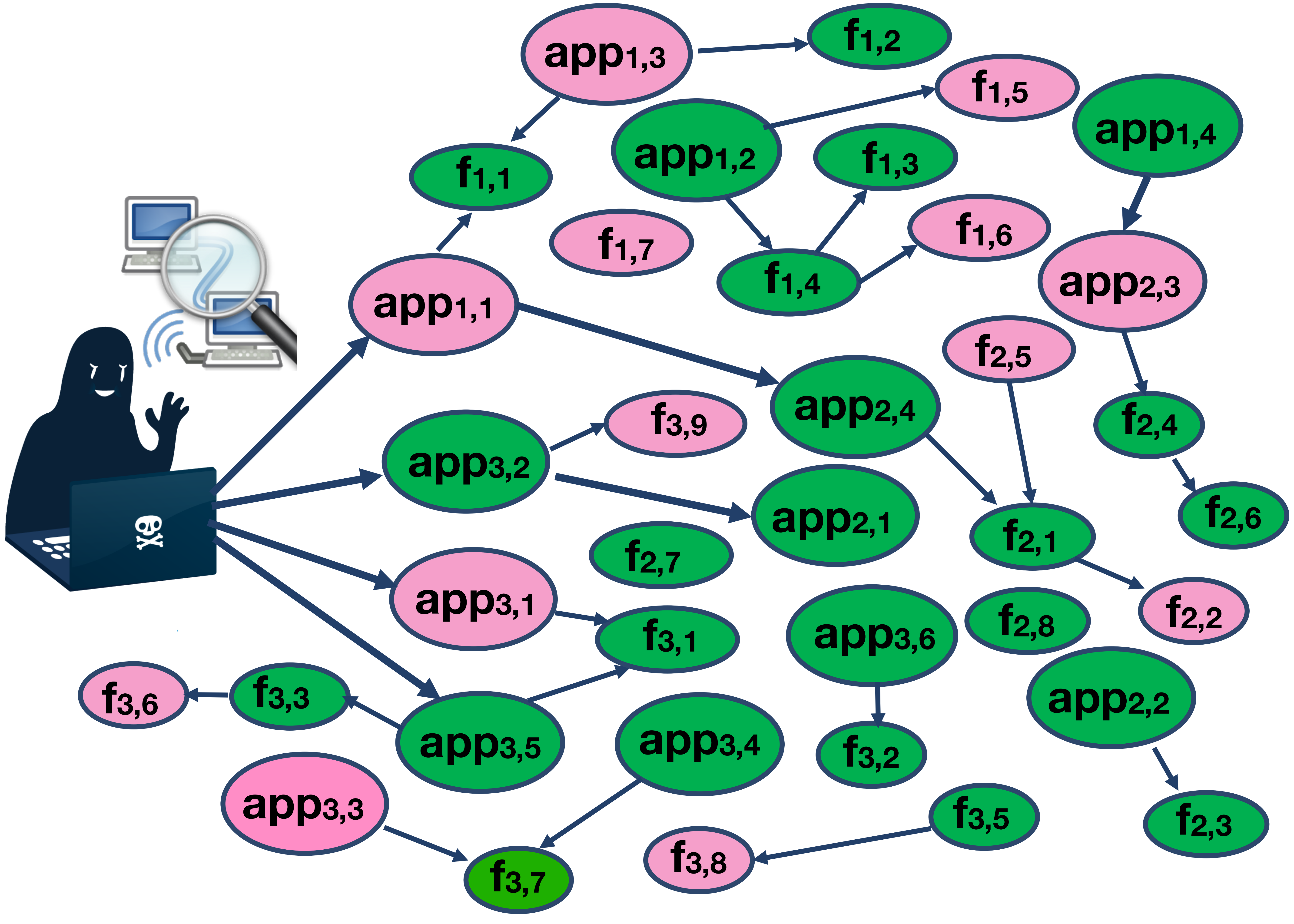
□ Type of attacks

- ◆ Remote-To-User attack (e.g., CVE 2009-1535)
- ◆ Remote-To-Root attack (e.g., CVE 2009-0015)
- ◆ User-To-Root attack (e.g., CVE 2008-4050)

□ Attack strategy: Adapted from Lockheed Martin's Cyber Kill Chain



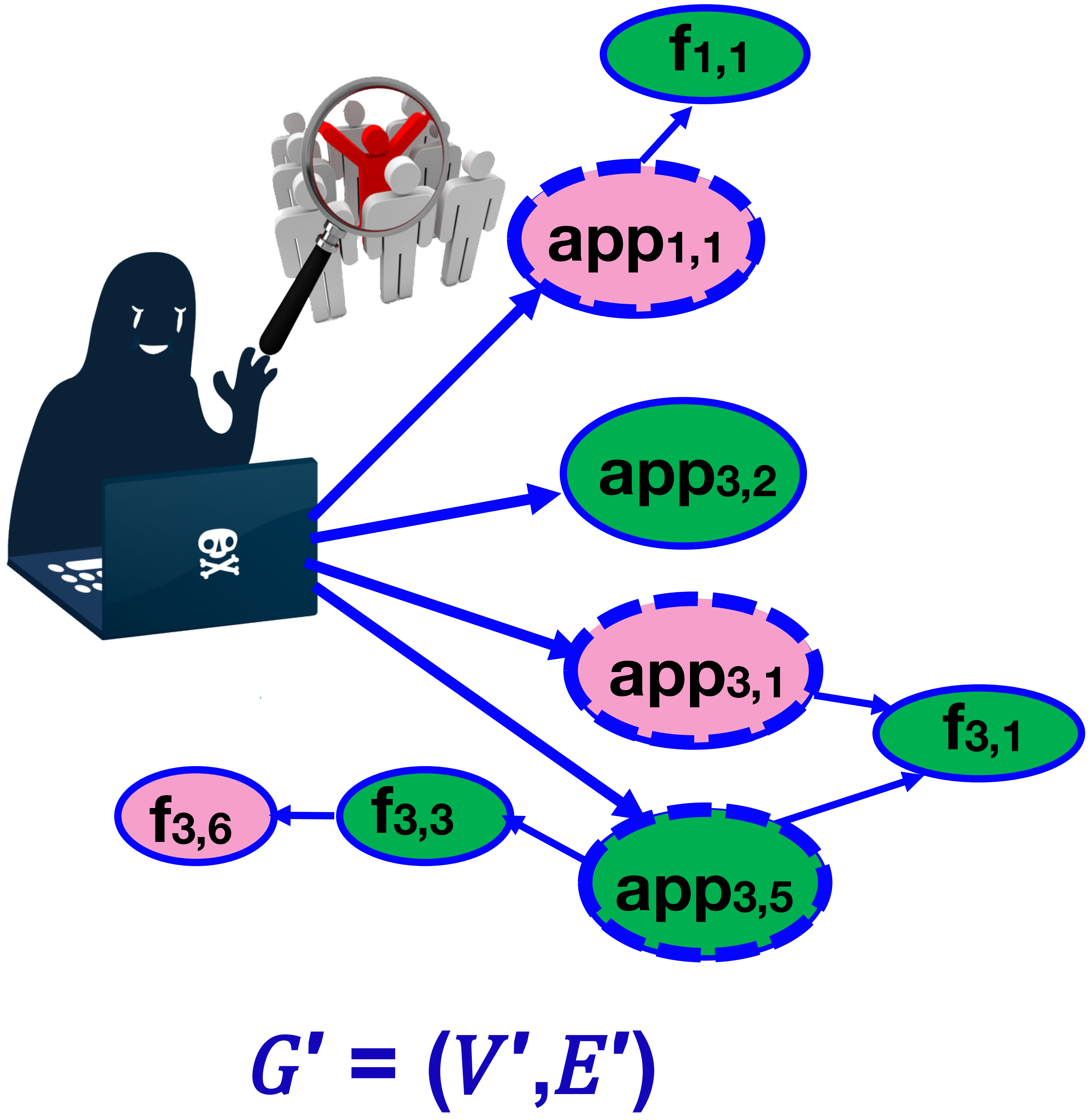
Modeling Attack Strategy Phase 1: Reconnaissance



G = (*V*, *E*)

- □ Gathering information about a target network (e.g., topology, vulnerabilities)
- □ Examples: Ping Sweeps, Port Scanning, Fingerprinting ….
- □ Output: Attacker’s view of target network $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$.

Modeling Attack Strategy Phase 2: Weaponization (1)



□ Given graph G' = (V',E') and the attacker's exploits X, attacker determines nodes $v \in V'$ suitable for targets.

□ A candidate app should satisfy

- ◆ Involved in internal-external communication E_*
- ◆ App contains a software vulnerability or there exists an access path from app to a vulnerable OS function

□ Client application vs. Server application

Modeling Attack Strategy Phase 2: Weaponization (2)

□ A candidate client application for initial compromise

$$(\exists \mathrm{vul} \in \varphi(v), \exists x \in \boxed{X : \psi(v)} = 1 \wedge \rho(x, \mathrm{vul}) > 0) \vee (\exists \mathrm{vul} \in \varphi(u), \exists x \in X : (u \in V_{i,os}) \wedge (v \in V_{i,app}) \wedge \mathrm{dep\_path}\boxed{(v, u) \wedge} \psi(u) = 1 \wedge \rho(x, \mathrm{vul}) > 0)). \quad (1)$$

□ The set of candidate client applications for initial compromise

$$\mathrm{Weapon}_0 = \{v \in (V' \cap V_{i,app}) : \eta(v) = 0 \wedge (((v, *) \in E_{*,io} \cap E') \vee ((*, v) \in E_{*,oi} \cap E')) \wedge \text{condition (1) holds}\}.$$

□ A candidate server application for initial compromise

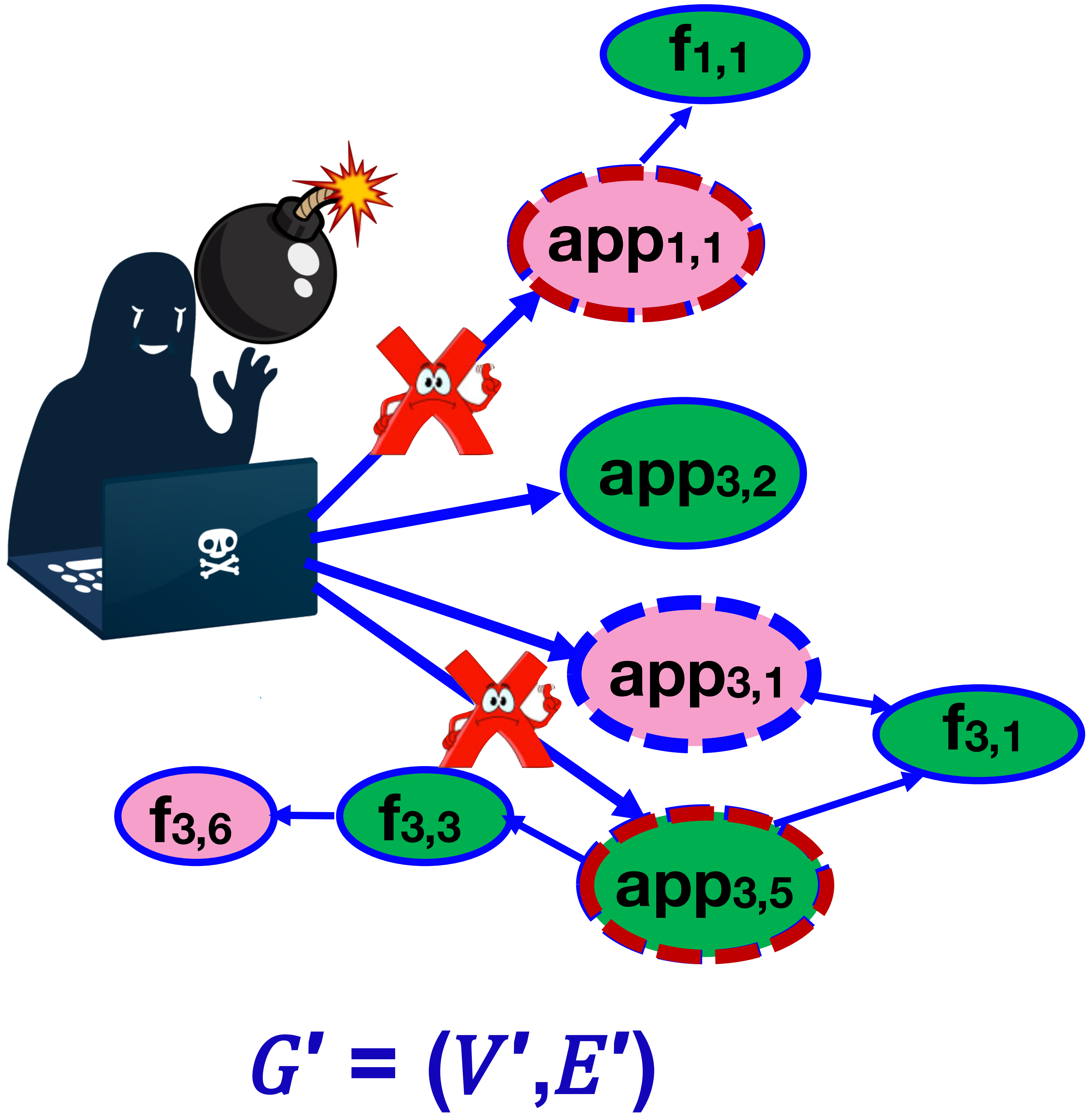
$$(\exists \mathrm{vul} \in \varphi(v), \exists x \in \boxed{X : \mathrm{loc(vul)}} = 1 \wedge \rho(x, \mathrm{vul}) > 0) \vee (\exists \mathrm{vul} \in \varphi(u), \exists x \in X : (u \in V_{i,os}) \wedge (v \in V_{i,app}) \wedge \mathrm{dep\_}\boxed{\mathrm{path}(v, u) \wedge} \mathrm{loc(vul)} = 1 \wedge \rho(x, \mathrm{vul}) > 0)). \quad (2)$$

□ The set of candidate server applications for initial compromise

$$\mathrm{Weapon}_1 = \{v \in V' \cap V_{i,app} : \eta(v) = 1 \wedge (*, v) \in (E_{*,oi} \cap E') \wedge \text{condition (2) holds}\}.$$

$\mathrm{Weapon} = \mathrm{Weapon}_0 \cup \mathrm{Weapon}_1$

Modeling Attack Strategy Phase 3: Initial compromise



□ Strategy to select a subset of Weapon for initial compromise

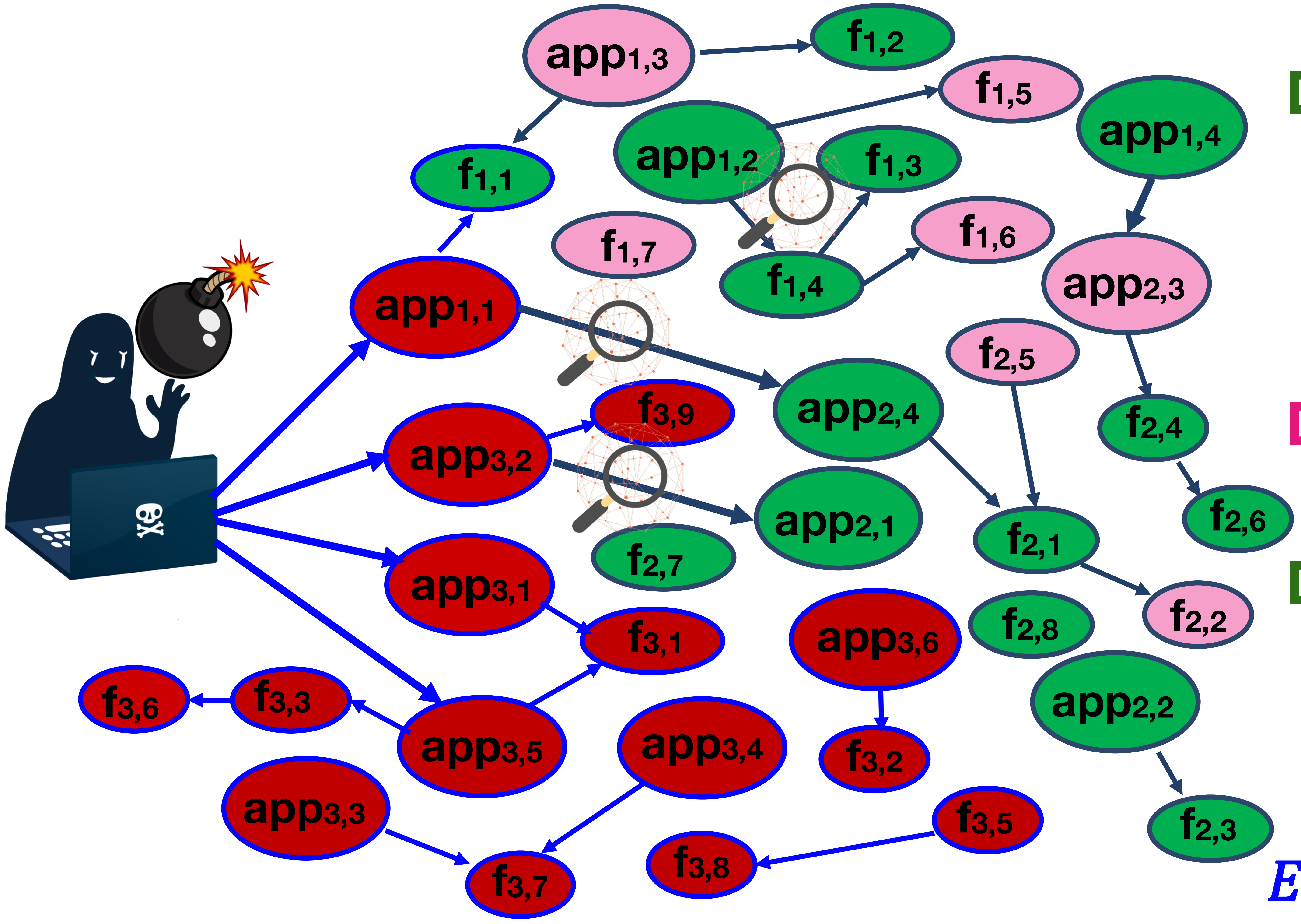
- ◆ Zero-day vulnerabilities first
- ◆ Compromise the OSes whenever possible
- ◆ Otherwise compromise all of the vulnerable apps

□ IniComp = { $\text{app}_{1,1}$, $\text{app}_{3,5}$ }

Remote-To-User attack

Remote-To-Root attack

Modeling Attack Strategy Phase 4: Further reconnaissance



$G' = (V', E')$

□ Once compromises a computer, attacker attempts to obtain information about sub-graph $G - G'$.

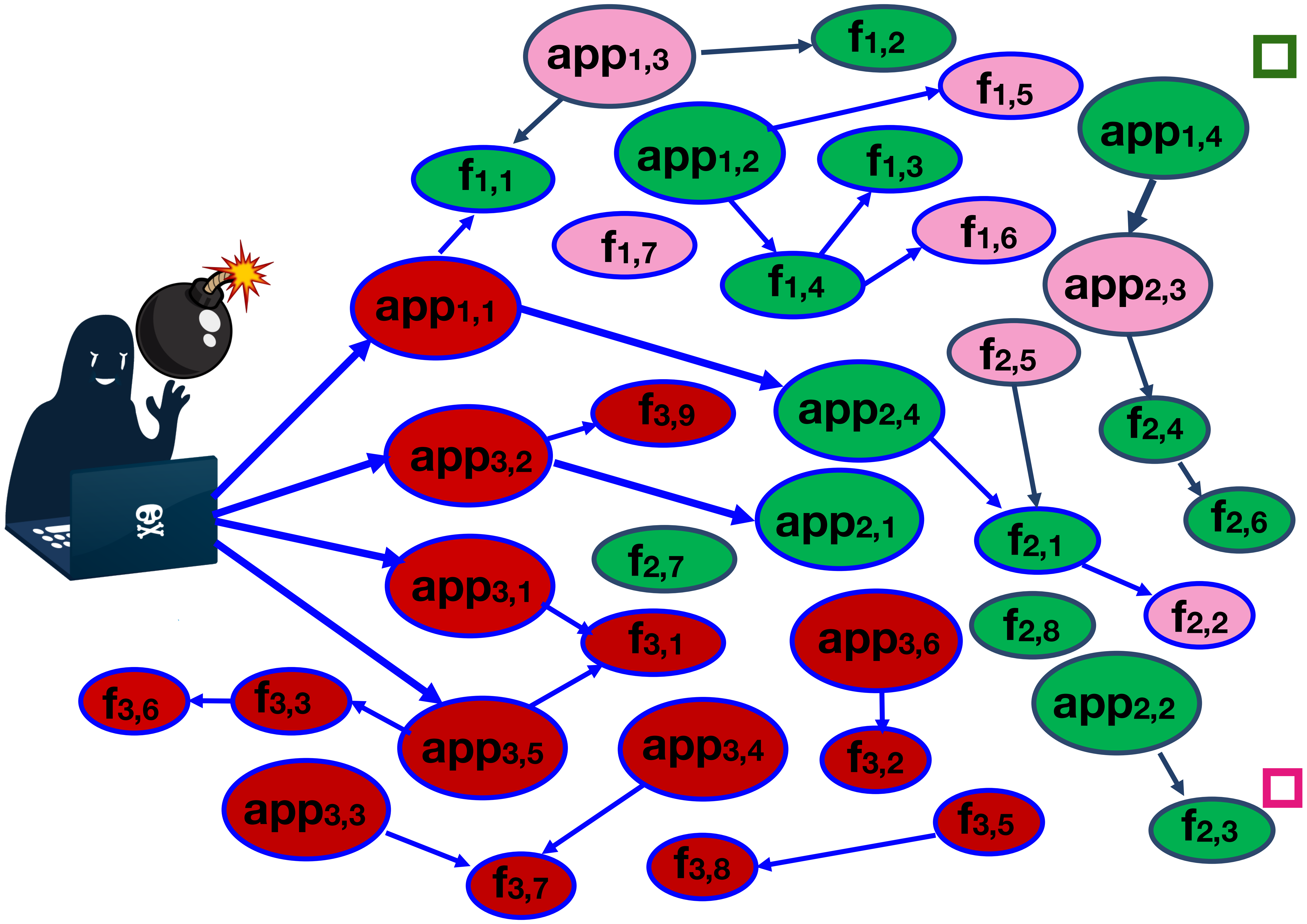
□ Can be conducted recursively

□ Attacker will update information about the enterprise network as

$$V' = V' \cup \{\text{app}_{2,1}, \text{app}_{2,4}, \text{app}_{1,2}, f_{2,1}, f_{2,2}\ldots$$

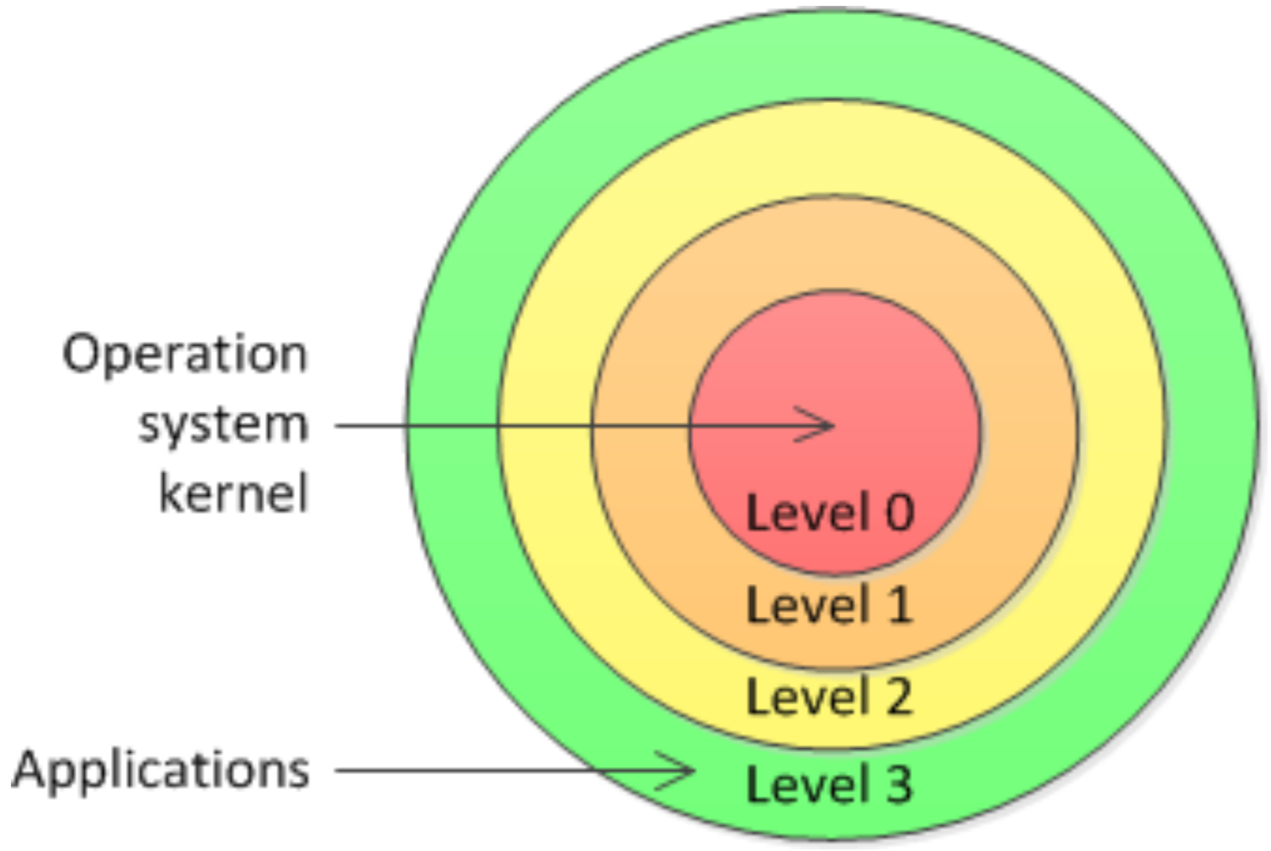
$$E' = E' \cup \{(\text{app}_{1,1}, \text{app}_{2,4}), (\text{app}_{3,2}, \text{app}_{2,1}), ..$$

Modeling Attack Strategy Phase 5: Privilege escalation



$G' = (V', E')$

□ After compromising an app but not OS, attacker attempts to compromise some vulnerable OS functions.



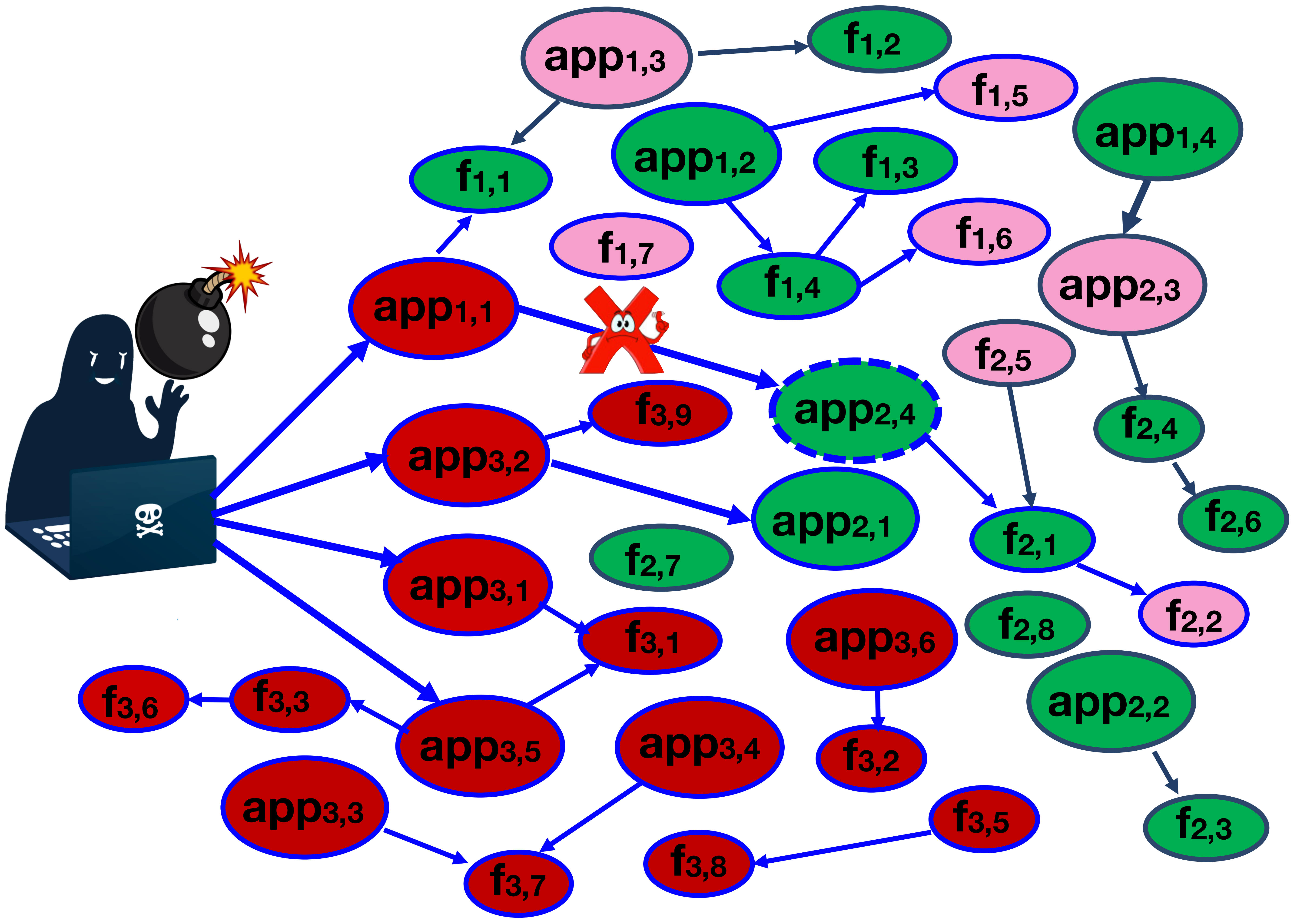
User-To-Root attack

□ tight vs. loose HIPS policy

$\exists v \in V_{i,app},\ \exists u \in V_{i,os},\ \exists \text{vul} \in \varphi(u),\ \exists x \in$

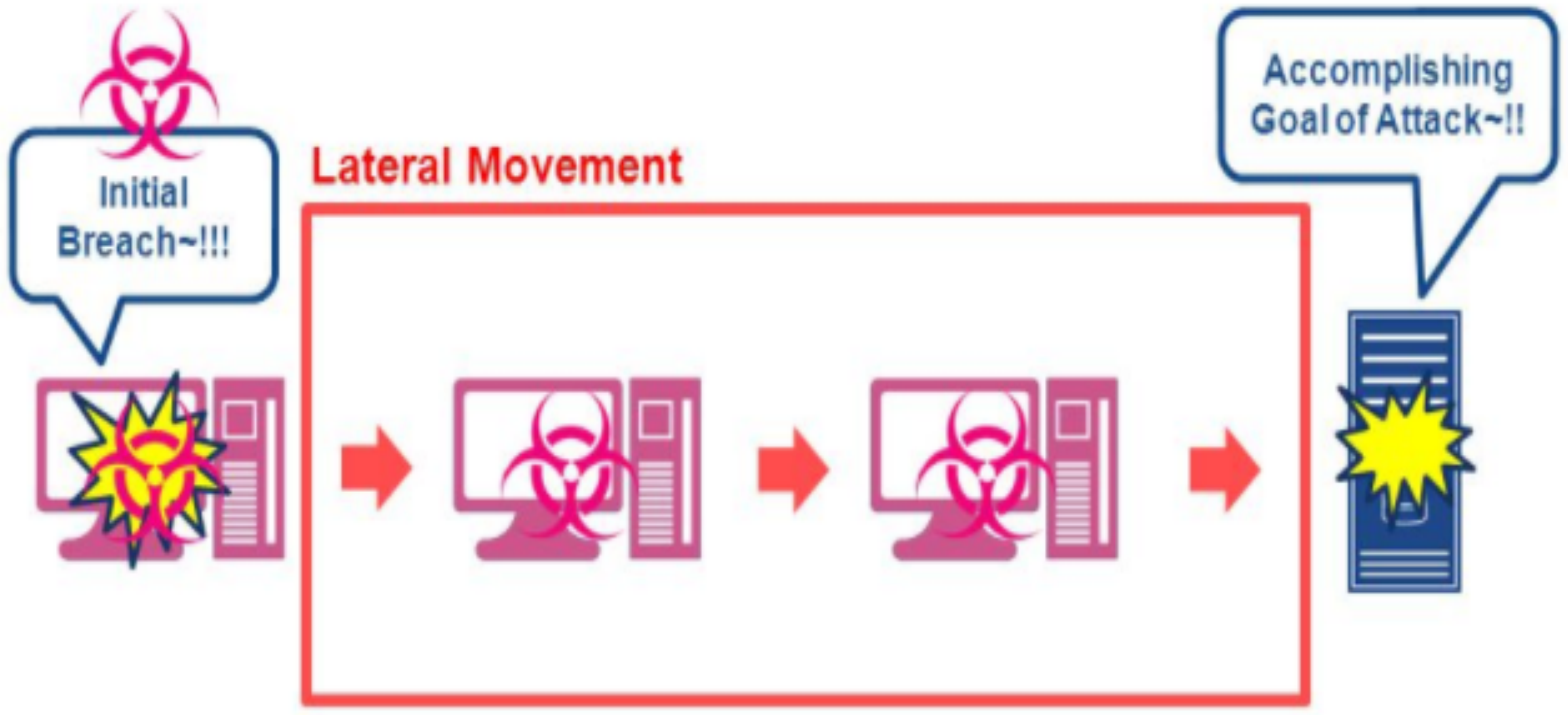
$\text{state}(v, t) = 1 \wedge \boxed{\text{dep\_path}(v, u)} \wedge \rho(x, \text{vul}) >$

Modeling Attack Strategy Phase 6: Lateral movement (1)



$G' = (V', E')$

- □ After penetrating into the network, attacker can leverage inter-computer communication $e \in E'$ to attack other computer.



Security Metrics

- □ **Percentage of compromised applications (pca) at time t**

$$\text{pca}(t) = |\{v \in V_{(app)} : \text{state}(v, t) = 1\}| / |V_{(app)}|$$

- □ **Percentage of compromised server applications (pcsa) at time t**

$$\text{pcsa}(t) = \frac{|\{v \in V_{(app)} \wedge \eta(v) \neq 0 : \text{state}(v, t) = 1\}|}{|\{v \in V_{(app)} \wedge \eta(v) \neq 0\}|}$$

- □ **Percentage of compromised OSes (pcos) at time t**

$$\text{pcos}(t) = |\{v \in V_{(os)} : \text{state}(v, t) = 1\}| / |V_{(os)}|$$

Simulation Setting and Methodology (1)

- □ Synthetic enterprise network
  - ◆ Computers
    - ✓ 1,000 desktops, 5 servers, OS={Windows}
    - ✓ Client APP = {browser, email client, IM, word processor, FTP client, database client}
    - ✓ Server APP= {web server, email server, DNS server, FTP server, database server}
    - ✓ Each OS function is called, directly or indirectly, by each app with probability δ.
  - ◆ Inter-computer communication E_0
    - ✓ See details in the paper
  - ◆ Internal-external communication E_*
    - ✓ See details in the paper

Simulation Setting and Methodology (2)

□ Vulnerabilities

- ✓ β: probability that each application contains a vulnerability
- ✓ ϑ: probability a vulnerability can be exploited remotely
- ✓ τ: probability that a vulnerability is zero-day
- ✓ $\psi(v) \in$ [0, 1]: the probability that a client app is vulnerable to social engineering attacks

□ Defenses

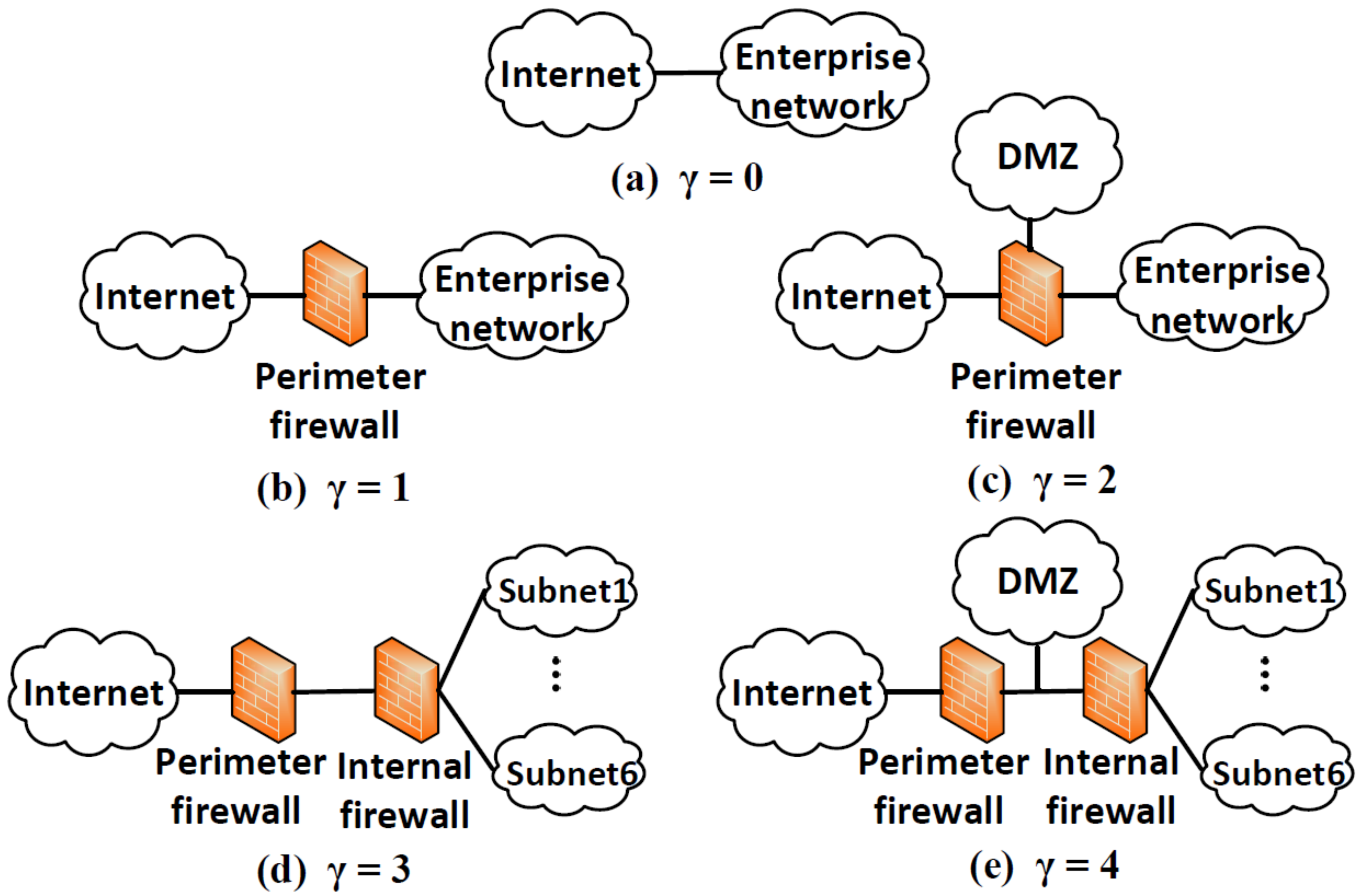
- ✓ Five combinations of firewalls and DMZ employment (identified by γ = 0, 1, 2, 3,4).
- ✓ k: fraction of known vulnerabilities can be prevented from being exploited by NIPS
- ✓ ζ : probability privilege escalation attempts are blocked by HIPS
- ✓ α : probability a social engineering attack is blocked

□ Attacks

- ✓ a: percentage of zero-day vulnerabilities that can be exploited by the attacker
- ✓ b: percentage of known vulnerabilities can be exploited by attacker but will are blocked
- ✓ c: percentage of known vulnerabilities can be exploited by attacker without being blocked
- ✓ $\rho(x$, vul) : probability that $x \in X$ successfully exploits a vulnerability vul
- ✓ ω : fraction of nodes that are discovered by attacker's initial reconnaissance

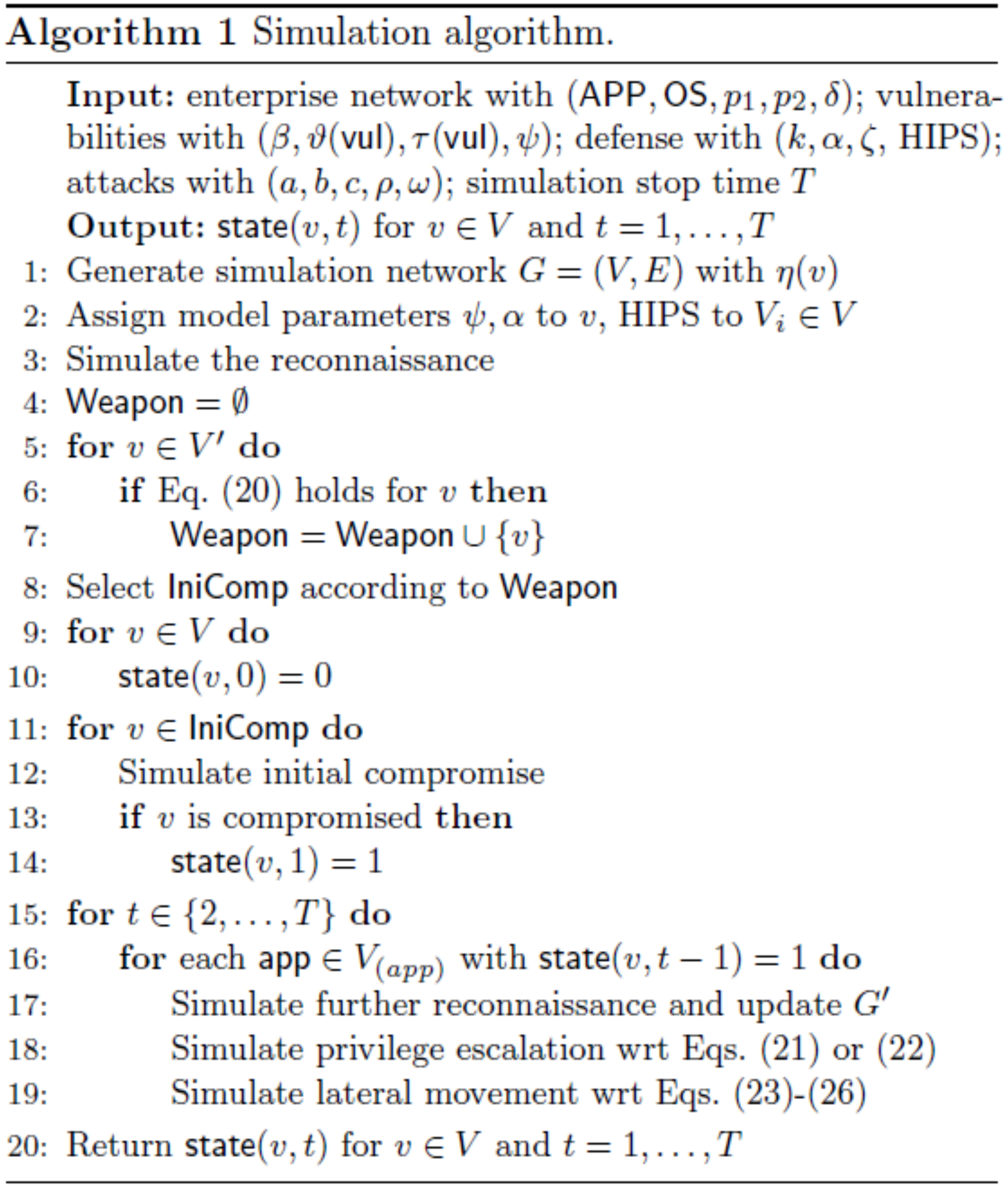
Simulation Setup and Results

Five combinations of firewalls and DMZ employment



- □ Assume the HIPS and NIPS are not effective in blocking attacks.
- □ Assume OSes are not vulnerable, consider other scenarios later.
- □ Network parameters: p_1 = 0.1, p_2 = 0.1, δ = 0.1
- □ Vulnerabilities parameters: $\psi(v)$ = 0.5, ϑ(vul) = 0.5,τ (vul) = 0.5
- □ Other defense parameters: k = 0, α = 0, ζ = 0, HIPS loose policy
- □ Attack parameters: (a, b, c) = (1, 1, 1), $\rho(x$, vul) = 1, ω = 1

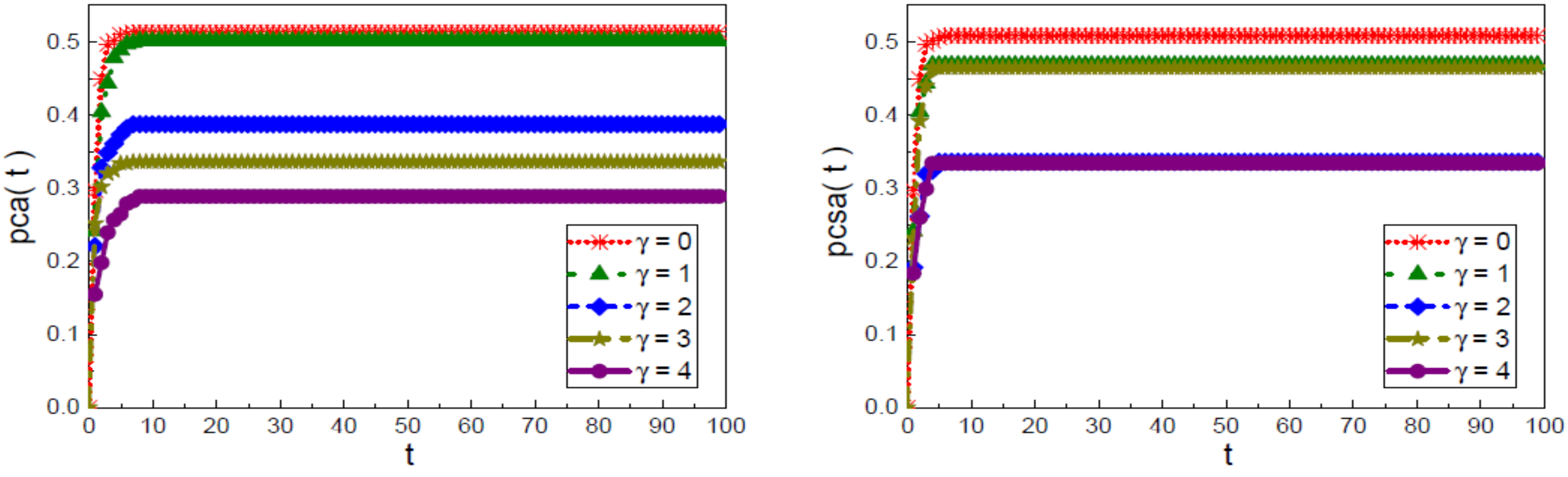
Simulation algorithm



Algorithm 1 Simulation algorithm.

```
Input: enterprise network with $(\mathsf{APP}, \mathsf{OS}, p_1, p_2, \delta)$; vulnerabilities with $(\beta, \vartheta(\mathsf{vul}), \tau(\mathsf{vul}), \psi)$; defense with $(k, \alpha, \zeta, \mathsf{HIPS})$; attacks with $(a, b, c, \rho, \omega)$; simulation stop time $T$
Output: $\mathsf{state}(v,t)$ for $v \in V$ and $t = 1, \ldots, T$
Generate simulation network $G = (V, E)$ with $\eta(v)$
Assign model parameters $\psi, \alpha$ to $v$, HIPS to $V_i \in V$
Simulate the reconnaissance
$\mathsf{Weapon} = \emptyset$
for $v \in V'$ do
    if Eq. (20) holds for $v$ then
        $\mathsf{Weapon} = \mathsf{Weapon} \cup \{v\}$
Select IniComp according to Weapon
for $v \in V$ do
    $\mathsf{state}(v, 0) = 0$
for $v \in \mathsf{IniComp}$ do
    Simulate initial compromise
    if $v$ is compromised then
        $\mathsf{state}(v, 1) = 1$
for $t \in \{2, \ldots, T\}$ do
    for each app $\in V_{(app)}$ with $\mathsf{state}(v, t-1) = 1$ do
        Simulate further reconnaissance and update $G'$
        Simulate privilege escalation wrt Eqs. (21) or (22)
        Simulate lateral movement wrt Eqs. (23)-(26)
Return $\mathsf{state}(v,t)$ for $v \in V$ and $t = 1, \ldots, T$
```

Determining simulation time horizon T



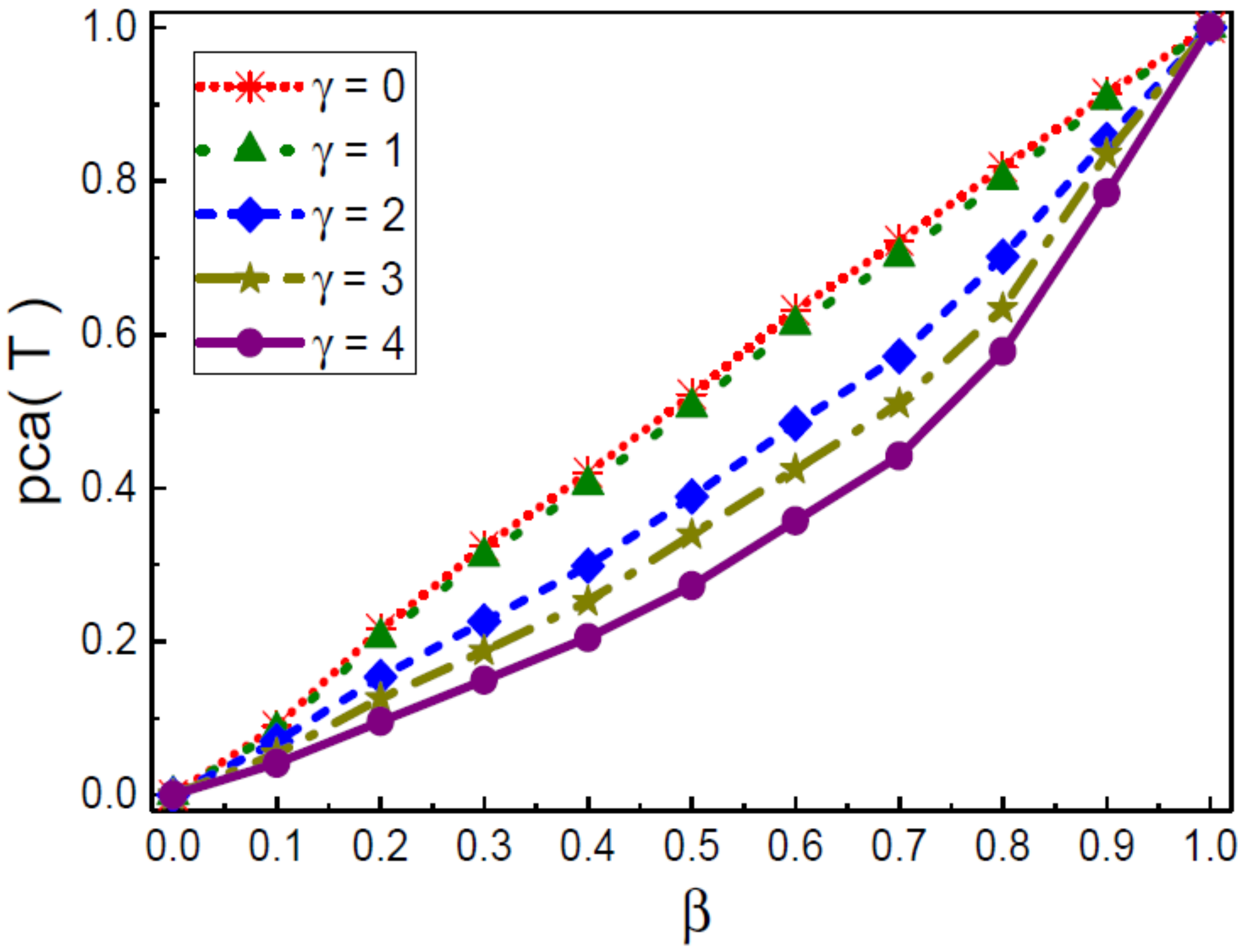
(a) pca(t) (b) pcsa(t)

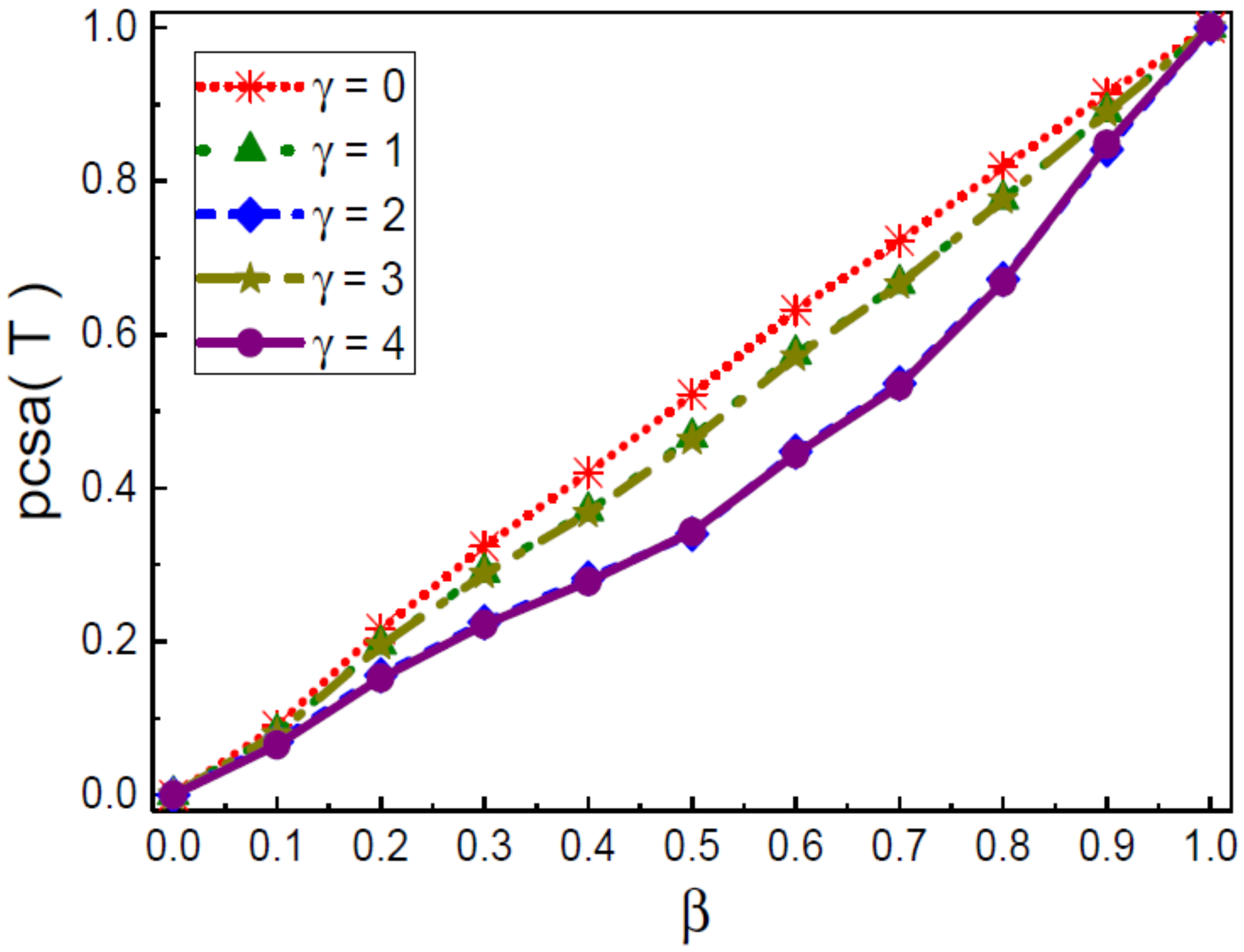
***β* =0.5**

Insight 1.

- **Both pca(t), the *percentage of compromised applications* at time t, and pcsa(t), the *percentage of compromised server applications* at time t, first increase exponentially and then converge to a steady value.**
  - ✓ **Exponential Increase: rich connections (any one can attack any one else)**
  - ✓ **Steady value: Lack of other defenses**

Security effectiveness of firewalls and DMZ (1)

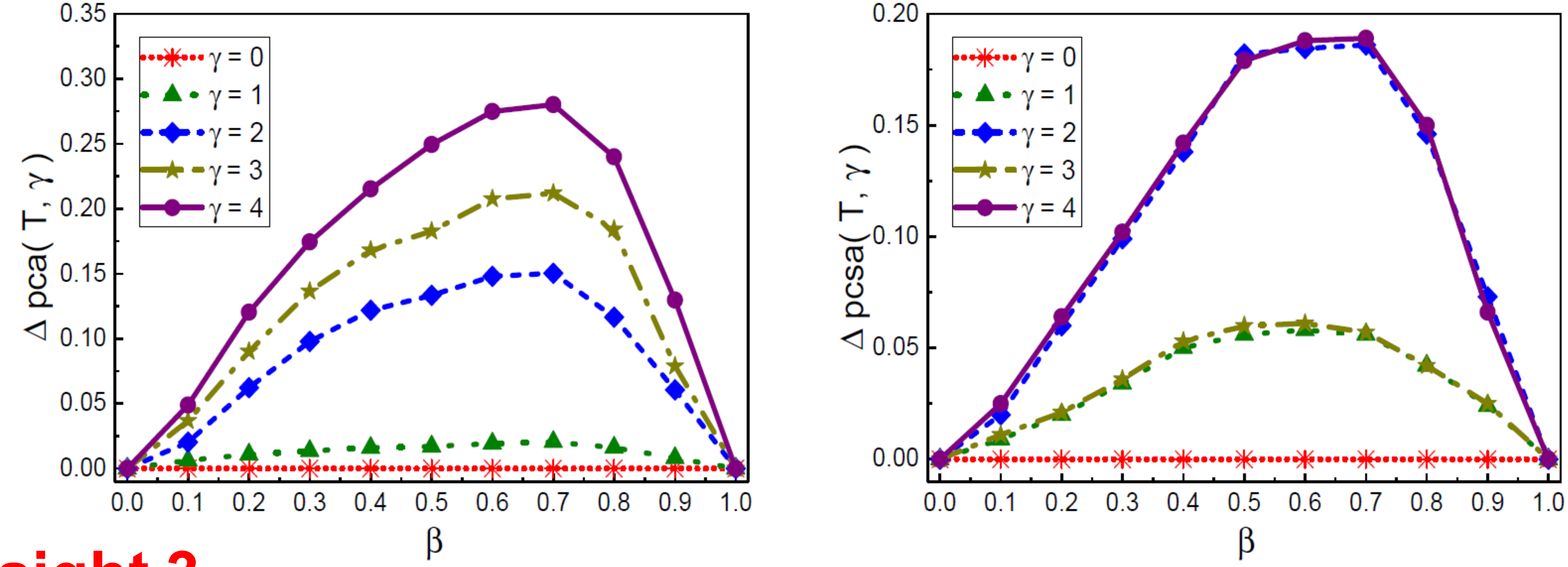




Insight 2.

- When OSes are not vulnerable, security effectiveness of a fixed combination of firewalls and DMZ decreases as fraction of vulnerable applications increases.
- Firewalls and DMZ are not effective when few or most computers are vulnerable.
  - ✓ Caveat: Under the assumption that HIPS and NIPS are not effective

Security effectiveness of firewalls and DMZ (2)



Insight 3.

- Employing perimeter firewall lone has a little security impact.
- Employing a comprehensive use of firewalls and DMZ can substantially increases security when β ∈[0.2,0.9] (probability that each application contains a vulnerability).
- Employing perimeter firewall and DMZ can substantially increase the security of sever applications when β ∈[0.2,0.9].

Related work

- **Epidemic spreading:**
  - Independence assumption
  - Coarse-grained model

- **Cybersecurity Dynamics:**
  - Dependence is partially addressed so far
  - Modeling aggregate effect of vulnerabilities and exploits

- **This paper:**
  - No independence assumption
  - Fine-grained modeling of vulnerabilities and exploits

Ongoing work

- □ More systematic experiments (e.g., HIPS, NIPS are effective): full version is to come

- □ On quantifying the security effectiveness of other preventive defense mechanisms (papers to come)

Conclusion

- First work on quantifying security effectiveness of firewalls and DMZs from a holistic perspective (i.e., global vs. local view).
  - Global view allows us to quantify the network-wide effectiveness of replacing one mechanism with an improved mechanism
- We need many more research on quantifying cybersecurity!!!!!!!