

# **An Analysis of Using Overlay Networks to Resist Distributed Denial-of-Service Attacks**

Ju Wang and Andrew A. Chien

Department of Computer Science and Engineering

University of California, San Diego

{jwang, achien}@cs.ucsd.edu

9500 Gilman Dr, La Jolla, CA 92039

## **Abstract**

Proxy networks based on overlays have been proposed as an architecture to protect Internet applications against denial-of-service attacks, however we know of no formal analysis of such schemes' effectiveness. We describe a framework to analyze a class of these proxy network architectures. Based on an attack model and a system model, we analytically characterize how attacks affect two key factors of such schemes: resource availability and secrecy of applications' locations. Our analytical models are applied to determine appropriate policies for resource recovery and system reconfiguration. Our conclusions show that: 1) intrusion detection-triggered recovery strategy is insufficient to avoid resource depletion, 2) true-positive rates of intrusion detectors have more impact on resource availability than detection speed, 3) simple reconfiguration approaches, such as random proxy migration, can effectively prevent attackers from discovering applications' locations, 4) overlay network topology is critical; richly-connected topologies may reduce a proxy network's effectiveness in resisting attacks.

## **Keywords**

security, availability, Denial-of-Service attack, overlay network, modeling

*Contact Author:* Ju Wang (phone) (858)534-5486, (fax) (858)822-2459

*Submission category:* Regular Papers describing recent research results

*Word count:* 6127

*Declaration:* this material has been cleared through author affiliations

# 1 Introduction

Denial-of-service (DoS) attacks have been a major security threat to Internet applications. Since 1998, there have been several cases of large-scale distributed DoS attacks, during which popular sites such as Yahoo! and Amazon were shut down[1, 2], and an important government website was forced to move to a different location[3]. These attacks have serious economic and political impact, and may even threaten critical infrastructures and national security.

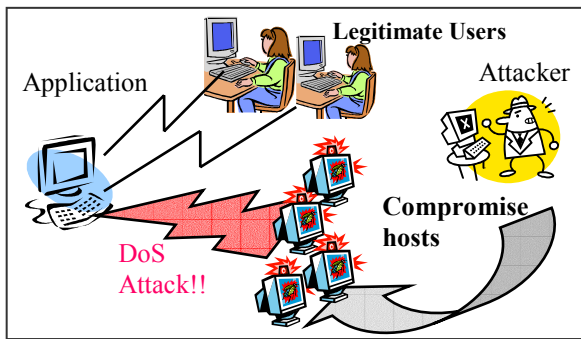


Figure 1 Denial-of-Service Attack

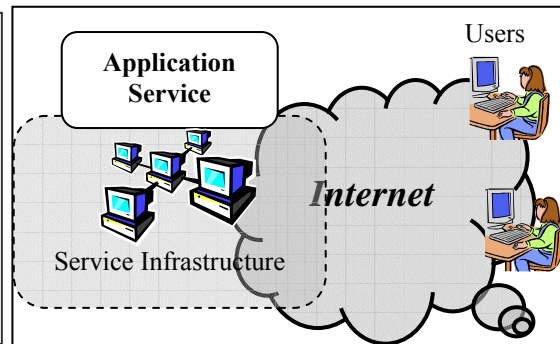


Figure 2 Example of Internet Application

As shown in Figure 1, a typical DoS attack has two stages. In the first stage, attackers compromise many hosts in the Internet and install “zombie” programs. In the second stage, the attackers control these zombie programs to attack the victim. According to the method of attack used in the second stage, DoS attacks can be categorized as *infrastructure level* or *application level attacks*. Figure 2 shows a typical Internet application deployment. The application service runs on a resource pool of interconnected hosts; users access it via the Internet. *Infrastructure level attacks* attack the resource pool directly, for example, by sending packet floods to saturate the victim network. *Application level attacks* cause denial-of-service by requesting significant amount of workload or by exploiting weaknesses on the application.

The fact that most Internet applications are publicly accessible makes them easy targets for *infrastructure level DoS attacks*. In this paper, we focus on *infrastructure level attacks* on publicly accessible applications.

Researchers are studying the use of overlay networks to tolerate DoS attacks on Internet applications[4]. We consider an overlay proxy network approach to tolerate *infrastructure level attacks* on publicly accessible applications. The key idea is to hide the application behind a proxy network (Figure 3), which itself is embedded in a network of a huge number of Internet hosts. The proxy network and applications use the Internet hosts as a resource pool; users access the applications via some edge proxies with known IP addresses. The proxy network can dynamically reconfigure so that attackers cannot easily locate the application, preventing the launch of *infrastructure level* DoS attacks. In addition, the applications are able to be moved amongst the hosts, separating them from dependence on a particular infrastructure, allowing them to tolerate infrastructure attacks. However, we focus on proxy network reconfiguration here as a primary way of system reconfiguration.

In this paper, we build a formal model and use it to study the effectiveness of overlay networks to tolerate DoS attacks. More specifically, subject to the formal model, we characterize how quickly resources can be compromised and the effectiveness of policies such as intrusion detection triggered recovery or a simple periodic system reset. We also characterize the difficulty for attackers to discover the location of the application. Applications of these models to several system scenarios yield the following novel conclusions:

- Intrusion detection-triggered recovery strategy is insufficient to avoid resource depletion.
- True-positive rates of intrusion detectors have more impact on resource availability than detection speed.
- System reconfiguration techniques such as random proxy migration can effectively prevent attackers from discovering applications' locations.
- Overlay network topology is critical; richly connected topology may reduce a proxy network's effectiveness in resisting attacks.

The remainder of this paper is structured as follows. Section 2 formulates the DoS problem.

Theoretical analysis method and important analytical results are presented in Section 3, followed by insights and discussions from those results in section 4. Section 5 addresses some limitations of our work. Section 6 briefly introduces some related work, and then we conclude in Section 7 with a summary of our conclusions and some suggestions for directions for future work.

## 2 Problem

We study the effectiveness of the proxy network as an architecture to resist DoS attacks. We first give an overview of the proxy network scheme, introducing some basic concepts, and explaining how the scheme works. Next we define the classes of attacks and the defensive mechanisms studied in this paper. Finally we define a set of key technical problems precisely.

### 2.1 Proxy Network Scheme Overview

#### ❖ Application and Proxy Network

The applications we study are client-server Internet applications, whose users are distributed over the Internet. As shown in Figure 3, the application<sup>1</sup> is hidden behind a *proxy network*, which is an overlay on a large resource pool of *hosts* (with intrusion detection systems) in an IP network, where *hosts* are accessible via their IP addresses. The application and *proxies* form an overlay network sharing the same resource pool.

#### ❖ Mapping, Location and Overlay Network Topology

*Nodes* in the overlay network (*proxies* and the application) are *mapped* to *hosts* in the resource pool. A *node* is *mapped* to a *host* if it is running on that *host*<sup>2</sup>, whose IP address is called the *location* of the *node*. One (a user, an attacker or another *node* in the overlay network) can

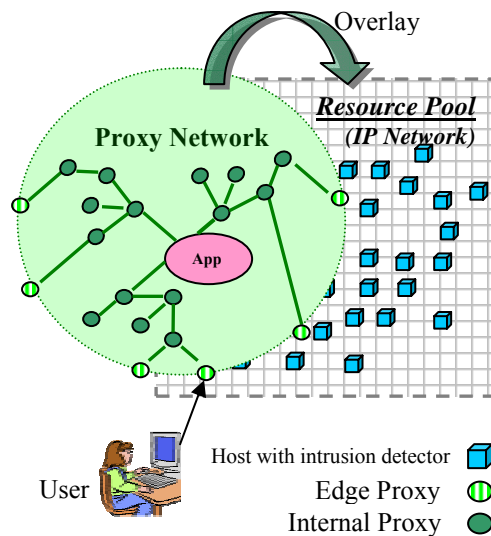


Figure 3 Proxy Overlay Network

<sup>1</sup> Multiple applications can share the same proxy network. For simplicity, we only consider the scenario with one application. The correlation among different applications during attacks is left to future study.

<sup>2</sup> Without loss of generality, we assume a *node* maps to one and only one *host*.

communicate with a *node* if it knows the *location* of that *node*.

Two *nodes* are *adjacent* if they know each other's *location*; therefore *adjacent* nodes can communicate directly. We can draw the overlay network topology, as shown in Figure 3 inside the shaded circle area, each *node* is a vertex in the graph; there is an edge between two vertices if and only if the corresponding *nodes* are *adjacent*. We borrow concepts such as *path* and *distance* from graph theory without elaboration. Two *nodes* can communicate if and only if there is a *path* between them in the topology graph.

❖ How does the proxy network scheme work?

There are three types of *nodes* in the overlay network (Figure 3): the application, *internal proxies* and *edge proxies*. *Location* of the application and *location* of *internal proxies* are hidden from both legitimate users and attackers; *location* of *edge proxies* is publicly known. There are *paths* from the *edge proxies* to the application in the overlay network so that users can access the application via these *edge proxies* without knowledge of the internal structure of the overlay network.

## **2.2 Attacks**

We analyze three classes of attacks: host compromise, denial of service and espionage. They have the following impact:

- *Resource loss*: some hosts in the pool become unusable due to attacks. Such hosts can either behave maliciously (disclosing information they have) or simply stop working.
- *Information disclosure*: critical system information is disclosed to attackers. We focus on *node exposure*, where a *node's location* becomes known to attackers.

### **State of Hosts and Nodes**

A *host* has the three states (population of hosts shown in Table 2-1): *intact*, *compromised* and *dos*. An *intact host* is in a clean state not affected by any attacks; a *compromised host* behaves maliciously; and a *dos host* has failed and stopped, not behaving maliciously.

Notation	Meaning
$\Psi_{\text{intact}}(t)$	Set of <i>intact hosts</i> at time $t$
$\Psi_{\text{com}}(t)$	Set of <i>compromised hosts</i> at time $t$
$\Psi_{\text{dos}}(t)$	Set of <i>dos hosts</i> at time $t$
$\Psi_{\text{target}}(t)$	Set of <i>hosts</i> under attack at time $t$
$\Psi_{\text{hosts}}$	Set of all <i>hosts</i> in the resource pool. $\Psi_{\text{hosts}} = \Psi_{\text{intact}}(t) + \Psi_{\text{com}}(t) + \Psi_{\text{dos}}(t)$

**Table 2-1 Notations of host population**

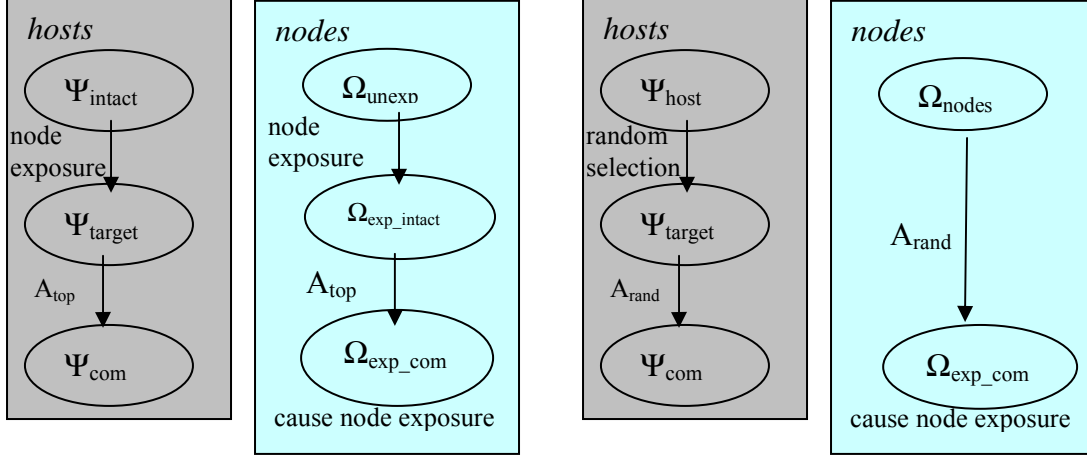
A *node* has three disjoint states (population of nodes shown in Table 2-2): *unexposed*, *exposed-intact* and *exposed-compromised*. An *unexposed node* runs on an *intact host* and is not *exposed*; an *exposed-intact node* is an *exposed node* that runs on an *intact host*; an *exposed-compromised node* is an *exposed node* that runs on a *compromised host*.

Notation	Meaning
$\Omega_{\text{unexp}}(t)$	Set of <i>unexposed nodes</i> at time $t$
$\Omega_{\text{exp\_intact}}(t)$	Set of <i>exposed-intact nodes</i> at time $t$
$\Omega_{\text{exp\_com}}(t)$	Set of <i>exposed-compromised nodes</i> at time $t$
$\Omega_{\text{exp}}(t)$	Set of <i>exposed nodes</i> . $\Omega_{\text{exp}}(t) = \Omega_{\text{exp\_intact}}(t) + \Omega_{\text{exp\_com}}(t)$
$\Omega_{\text{nodes}}$	Set of all <i>nodes</i> in the overlay network. $\Omega_{\text{nodes}} = \Omega_{\text{exp}}(t) + \Omega_{\text{unexp}}(t)$

**Table 2-2 Notations of node population**

❖ *Attack Class I: Host Compromise ( $A_{\text{top}}$  or  $A_{\text{rand}}$ )*

Attackers can attack and compromise *hosts* in the resource pool if they know their IP addresses. When a *host* is compromised, it becomes malicious and information stored on it is disclosed. In particular, when the *host* on which a *node* runs is compromised, the *node*'s *adjacent nodes* are *exposed* to attackers. Therefore host compromise has two impacts: resource loss and information disclosure. In addition, attacks such as eavesdropping, that cause both information disclosure and resource loss (hosts being eavesdropped cannot be used as sound resources) are generalized as host compromise.



**Figure 4**  $A_{top}$  attacks

**Figure 5**  $A_{rand}$  attacks

As shown in Figure 4 and Figure 5, there are two types of host compromise attacks,  $A_{top}$  and  $A_{rand}$ , according to the way targets are chosen. The targets of  $A_{top}$  attacks are the *hosts* on which *exposed nodes* run, therefore  $A_{top}$  attacks proceed along the overlay network topology towards the core. The targets of  $A_{rand}$  attacks are randomly selected from the resource pool. When a *node* gets into the *exposed-compromised* state, its *adjacent nodes* are *exposed*, enabling further penetration of  $A_{top}$  attacks.

❖ Attack Class II: Denial of Service (denoted by  $A_{dos}$ )

Attackers can launch infrastructure level DoS attacks against any *hosts* whose IP addresses are known. These attacks prevent *hosts* from functioning effectively to host *nodes* in the overlay network. Such attacks cause resource loss. Application level attacks such as overloading the application with excessive quantity of requests are not the focus of this paper.

❖ Attack Class III: Espionage (denoted by  $A_{esp}$ )

Attacks that disclose system information (*node locations*) but do not cause resource losses are called espionage. Examples include insiders stealing classified documents, traffic analysis and so on. We focus on the impact not the behavior of such attacks. Information disclosed in espionage along with public information (such as *edge proxy location*) is called prior information.

### 2.3 Defensive Mechanisms

We consider two levels of defensive mechanisms. At the level of resource pool, resource recovery mechanisms reset *compromised hosts* into *intact* state, therefore reverse resource loss; at the level of overlay network, system reconfiguration schemes make the system dynamic, therefore defend against information disclosure.

#### ❖ Resource Recovery

We study two strategies to trigger resource recovery. The recovery mechanisms can be the same in both cases.

- *Intrusion detection triggered recovery*: intrusion detection systems in the resource pool can raise alerts when *hosts* are *compromised*, which trigger recovery of *compromised hosts*.
- *Random reset*: *Hosts* can periodically reset themselves into *intact* state.

#### ❖ System Reconfiguration

We study a simple form of system reconfiguration: *random proxy migration*. *Proxies* periodically change their *location* among hosts in the resource pool. *Random proxy migration* can move *exposed nodes* to *locations* unknown to attackers, therefore removing these *nodes* from  $\Omega_{\text{exp}}$  and disrupting  $A_{\text{top}}$  attacks. We assume the overlay network topology does not change, and there is no application reconfiguration either.

## 2.4 Problem Definition

#### ❖ System Failures

We assume attackers are not able to concurrently DoS attack all the *edge proxies*. The *proxy network* scheme described above may fail due to the following reasons<sup>3</sup>:

- *Resource Depletion*: There are insufficient resources correctly functioning in the resource pool for either the proxy network or application to operate properly.

---

<sup>3</sup> Attackers can concurrently attack a set of (internal or edge) proxies to cause DoS on the application. However the impact of these attacks can be contained by means of proxy network reconfiguration which does not involve application reconfiguration. Here we focus on how securely the proxy network can hide application's location. How effectively the proxy network can contain the impact of such attacks is part of future work.

- *Core Exposure*: The *location* of the application is exposed to attackers, so that attackers can launch infrastructure level denial-of-service attack against it.

❖ Problem

We study the effectiveness of the *proxy network* scheme. With the described defensive mechanisms and attack classes, we develop an analytical model to answer these questions:

- Resource depletion: what percentages of hosts in the resource pool are *intact*?
- Exposure propagation (core exposure I): how far can attackers penetrate the proxy network from a given *exposed node* (the distance to the nodes attackers can expose)?
- Node exposure (core exposure II): what is the probability of a given *node* becoming *exposed*?

### 3 Analytical Model and Results

In this section, we first introduce statistical models to characterize the attacks, the resource recovery mechanisms, and the system reconfiguration schemes discussed in the previous section; then we study the problems defined in Section 2.4. In these models, we focus on characterizing the essence of the problem rather than capturing all the details of the real world. Limitations of these models are discussed in Section 5.

#### 3.1 Analytical Models

##### 3.1.1 Overlay Network

We assume the overlay network topology (defined in Section 2.1) does not change over time.

Each *node* in the overlay has at most  $R$  *adjacent nodes*; i.e.  $R$  is the max fan-out of the topology.

##### 3.1.2 Attack Model

❖ Host Compromise

Attackers can concurrently attack at most  $M_c$  hosts. We model occurrences of host compromises as a Poisson process  $\text{Prob}\{h \in \Psi_{\text{com}}(t) \mid h \in \Psi_{\text{intact}}(0) \cap \Psi_{\text{target}}(0)\} = 1 - e^{-\lambda t}$  ( $t \geq 0$ ). For simplicity, we assume all hosts have the same attributes. The probability of an intact host being compromised in time  $t$  follows exponential distribution with rate  $\lambda$ .

❖ Denial of Service

Attackers can concurrently DoS attack at most  $M_d$  hosts;  $|\Psi_{dos}(t)| \leq M_d$ . We assume impact of DoS attacks is instantaneous and transient, namely hosts that are subject to DoS attacks are in *dos* state for the duration of the attack, and become *intact* when the attack ends.

❖ Espionage

Espionage attacks can expose *nodes* in the overlay network. Our model does not describe how *nodes* are *exposed*; we only study the impact of node exposure caused by such attacks.

### 3.1.3 Defensive Mechanisms

❖ Resource Recovery

Resource recovery converts *compromised hosts* back to *intact* state. We model the two recovery triggering strategies discussed in Section 2.3 as follows.

- *Intrusion detection triggered recovery*: Intrusion detectors have a true positive rate<sup>4</sup> of  $\rho$  and only detected *compromised hosts* can be recovered. We model occurrences of *host* recoveries as a Poisson process, therefore  $\text{Prob}\{h \in \Psi_{intact}(t) | h \in \Psi_{com}(0)\} = \rho(1 - e^{-\mu_d t})(t \geq 0)$ . The probability of a *compromised host* being recovered within time  $t$  follows a scaled exponential distribution.  $\mu_d$  reflects the speed of the detectors, and  $\rho$  is the true positive rate of the intrusion detectors. Since false positives do not affect our analysis, our model does not include them.

- *Random reset*: Every host resets itself into *intact* state once in a while. We model occurrences of such random resets as a Poisson process with rate  $\mu_s$ , namely  $\text{Prob}\{h \in \Psi_{intact}(t) | h \in \Psi_{com}(0)\} = 1 - e^{-\mu_s t}(t \geq 0)$ .

❖ System Reconfiguration

We model occurrences of random *proxy* migration as a Poisson process with rate  $\mu_r$ . The probability of a *proxy*  $n$  changing its *location* within time  $t$  follows exponential distribution with rate  $\mu_r$ .  $\text{Prob}\{n \in \Omega_{unexp}(t) | n \in \Omega_{exp}(0)\} = 1 - e^{-\mu_r t}(t \geq 0)$ .

---

<sup>4</sup> The true positive rate of an intrusion detector is the ratio of the number of intrusions it can detect (eventually) over the total number of intrusions actually occurred.

Notations used in our analytical models are summarized in Table 3-1 for reference.

Notation	Meaning
R	Max fan-out of the overlay network topology
$M_c$	Max number of concurrent <i>host compromise</i> attacks
$M_d$	Max number of concurrent DoS attacks
$\lambda$	Rate of <i>host compromise</i>
$\mu_d$	Rate of intrusion detection triggered recovery
$\rho$	True positive rate of intrusion detectors
$\mu_s$	Rate of random reset
$\mu_r$	Rate of random proxy migration

**Table 3-1 Notations in Analytical Models**

### 3.2 Analysis I: Resource Depletion

Both DoS and host compromises cause resource depletion. We can cleanly separate the impact of DoS attacks and host compromises, therefore we study them separately.

#### ❖ Impact of Denial of Service

According to our attack model, there are  $|\Psi_{dos}|$  *dos hosts* in the resource pool due to DoS attacks at any time, and  $|\Psi_{dos}| \leq M_d$ . Therefore, at any time there are at least  $N = |\Psi_{hosts}| - M_d$  hosts not being affected by DoS attacks. We call  $N = |\Psi_{hosts}| - M_d$  the *effective size* of the resource pool.

#### ❖ Impact of Host Compromise

As an approximation, we separate the impact of DoS attacks by considering a resource pool with its *effective size*  $N$ . We study a lower bound of the *intact host* percentage in the resource pool for any combination of *host compromise* attacks ( $A_{top}$  and  $A_{rand}$ ).

❖ **Claim I:** *If initially all hosts are intact, and  $0 \leq m \leq C_1$ , then for  $t > 0$   $f(t) \geq C_2 \geq C_1$ . Otherwise, if*

*$m > C_1$ , then  $\lim_{t \rightarrow \infty} f(t) = C_1 > C_2$ . Where  $f(t) = \frac{|\Psi_{intact}(t)|}{N}$  is the expected percentage of intact*

*hosts over the effective size of the resource pool,  $m = \frac{M_c}{N}$  is attackers' relative capability,*

$C_1 = (1 + \frac{\lambda\rho}{\mu_{ds}} + \frac{\lambda(1-\rho)}{\mu_s})^{-1}$  and  $C_2 = 1 - \frac{\lambda\rho}{\mu_{ds}}m - \frac{\lambda(1-\rho)}{\mu_s}m$ ,  $\mu_{ds}$  is a constant defined by the

*recovery triggering strategies and  $\max(\mu_{ds}, \mu_s) \leq \mu_{ds} \leq \mu_d + \mu_s$ .*

$C_1$  is a threshold defined by properties of the resource recovery triggering strategies. This claim says that when attackers' relative capability  $m \leq C_1$ , there is a provable lower bound  $C_2 \geq C_1$  for the expected percentage of *intact* hosts in the resource pool. When attackers' relative capability is greater than the threshold  $C_1$ , there is only a weaker claim that the expected percentage of intact hosts will eventually stabilize at  $C_1$ . (Proof is in the Appendix)

### 3.3 Analysis II: Core Exposure

Here we study the last two problems discussed in Section 2.4: exposure propagation and node exposure. We focus on the impact of host compromise and espionage attacks in this section. Impact of DoS attacks is addressed in Section 5.

#### 3.3.1 Exposure Propagation

Suppose a node  $n_e$  is initially exposed, we study the probability of  $A_{\text{top}}$  attackers penetrating  $d$  hops into the overlay network from  $n_e$ . Let  $\Omega(n_e, d)$  denote the set of *nodes*  $d$  hops away from  $n_e$ . Formally, we study  $\text{Prob}\{\Omega_{\text{exp}}(t) \cap \Omega(n_e, d) \neq \emptyset | n_e \in \Omega_{\text{exp}}(0)\}$ , denoted by  $\pi_{n_e}(d, t)$  or  $\pi(d, t)$ .

❖ **Definition:**  $\xi(s) = q + (1 - q)s^{R-1}$ ,  $|s| \leq 1$ , where  $s$  is a complex variable,  $q = \frac{\mu_r}{\lambda + \mu_r}$ , and  $R$

(see Section 3.1.1) is the fan-out of the overlay topology graph. The *iterates* of the  $\xi(s)$  are  $\xi_0(s) = s$ ,  $\xi_1(s) = \xi(s)$ ,  $\xi_{n+1}(s) = \xi[\xi_n(s)]$ .  $n = 1, 2, \dots$

❖ **Claim II:** *If  $(1 - q)(R - 1) < 1$ , then  $\exists D > 0$  such that for  $\forall d \geq D$  and  $\forall t$ ,  $\pi(d, t) = 0$ .*

Claim II says with certain overlay topology and migration rate, the depth of  $A_{\text{top}}$  attackers' penetration is limited.

❖ **Claim III:**  $\forall t, \pi(d, t) \leq 1 - \xi_d(0)$ .

It can be proved that  $\pi(d, t)$  decreases when  $d$  increases. Therefore in practice, we can use Claim III to compute the *safety distance*  $D_\delta$ , such that  $\forall d \geq D_\delta, \forall t, \pi(d, t) \leq \delta$  for small constant  $0 \leq \delta \ll 1$ . Namely with probability  $(1 - \delta)$ , attackers cannot penetrate more than  $D_\delta$  hops from an *exposed*

*node*. Proofs are in the Appendix.

### 3.3.2 Node Exposure

For any *node*  $n_0$  in the overlay network, we study the probability of exposure, formally  $\text{Prob}\{\exists t, n_0 \in \Omega_{\text{exp}}(t)\}$ , denoted by  $\eta(n_0)$ . We first study the problem without prior information (espionage attacks and exposed edge proxies, defined in Section 2.2) and then show how to combine the impact of prior information.

❖ **Claim IV:** *Suppose initially no nodes are exposed and there are no espionage attacks, if  $N_i$  is a non-decreasing series, then  $\eta(n_0) \leq 1 - \left( \sum_{i=0}^{\infty} \frac{N_i}{N} \xi_i(0) \right)^{M_c}$ .  $N = |\Psi_{\text{hosts}}| - M_d$  is the effective size of the resource pool;  $N_i = |\Omega(n_0, i)|$  for  $i < \infty$ , and  $N_{\infty} \equiv N - \sum_{i < \infty} N_i$ .  $N_i$  is the number of nodes  $i$  hops away from  $n_0$  in the overlay topology.*

Claim IV shows an upper bound of the impact of host compromise attacks without prior information. This upper bound is defined by the overlay network topology, rate of random proxy migration, size of resource pool and capability of attackers. The proof is in the Appendix.

❖ **Claim V:** *Consider a *node*  $n_k$   $k$  hops away from  $n_0$ ,  $\eta^*(n_0) = \text{Prob}\{\exists t, n_0 \in \Omega_{\text{exp}}(t) | n_k \in \Omega_{\text{exp}}(0)\}$  and  $\eta(n_0) = \text{Prob}\{\exists t, n_0 \in \Omega_{\text{exp}}(t) | n_k \in \Omega_{\text{unexp}}(0)\}$ , assume all espionage attacks happen before time 0, then  $\eta^*(n_0) < 1 - (1 - \eta(n_0)) \xi_k(0)$ .*

Claim V shows the impact of prior information acquired via espionage attacks or given publicly such as *exposed edge proxies*. The correctness of Claim V follows directly from Claim III.

Using Claim IV and V we can compute an upper-bound for  $\eta(n_0)$  in general.

In this section, we study the three problems introduced in Section 2.4 and have the following results. First, there is a threshold  $C_1$  defined by properties of the resource recovery triggering strategies; when attackers' relative capability is lower than  $C_1$ , there are always a certain percentage of *intact hosts* in the resource pool. Second, when the overlay network topology and the rate of random proxy migration meet certain requirement, the depth of  $A_{\text{top}}$  penetration can be limited; Claim III shows how to compute the depth of  $A_{\text{top}}$  penetration. Third, there is an upper

bound on the probability of attackers exposing a given *node*; this upper bound is defined by the overlay network topology, proxy migration rate, resource pool size and the capability of attackers.

## 4 Parametric Analysis

Using the models derived in Section 3, we study the following questions in this section.

- I. *To avoid resource depletion, what resource recovery policies are necessary?*
- II. *How to choose intrusion detectors to more effectively avoid resource depletion?*
- III. *Are simple schemes like random proxy migration effective to avoid core exposure?*
- IV. *What type of overlay network topology can effectively avoid core exposure?*

### 4.1 Effectiveness of Resource Recovery Policies

❖ *An intrusion detection-triggered recovery strategy is insufficient to avoid resource depletion; schemes like random reset are necessary.*

When there are only intrusion detection-triggered recoveries, formally  $\mu_s=0$ , we know from Claim I that  $\lim_{t \rightarrow \infty} f(t) = C_1 = 0$  as long as the true positive rate  $\rho < 1$ . This implies that all hosts in the resource pool are eventually compromised. Intuitively, because of imperfect detectors in that they cannot detect all intrusions, this recovery triggering strategy cannot fully recover all the compromised hosts.

On the other hand, with random reset, namely  $\mu_s > 0$ , we know from Claim I that  $\lim_{t \rightarrow \infty} f(t) \geq C_1 > 0$ . In other words, we can assure that a certain percentage of *hosts* are *intact*. Furthermore, with appropriate choices of reset rate  $\mu_s$  to make  $C_1$  large enough such that

attackers' relative capability  $m = \frac{M_c}{|\Psi_{hosts}| - M_d} < C_1$ , then from Claim I we know that the

expected percentage of the *intact* hosts is no less than  $C_2 = 1 - \frac{\lambda \rho}{\mu_{ds}} m - \frac{\lambda(1-\rho)}{\mu_s} m > C_1 > m$ .

This means that appropriate random reset schemes can effectively avoid resource depletion. The impact of reset rate can be clearly seen in Figure 6 and Figure 7.

## 4.2 Choice of Intrusion Detectors

❖ *True positive rates have more impact on resource availability than detection speed*

Figure 6 and Figure 7 show how true positive rate, detection speed and random reset rate affect resource availability. We can see that resource availability is more sensitive to the true positive rate than the detection speed. The difference is very significant when the reset rate is low. Intuitively, when reset rate is low, the amount of residue left behind from intrusion alert-based recovery becomes a significant factor to resource availability. The true positive rate decides the amount of residue, but detection speed has no impact on that. Normally, random resets involve overheads; lower reset rates are preferable. This makes true positive rate the most important factor in choosing an intrusion detector.

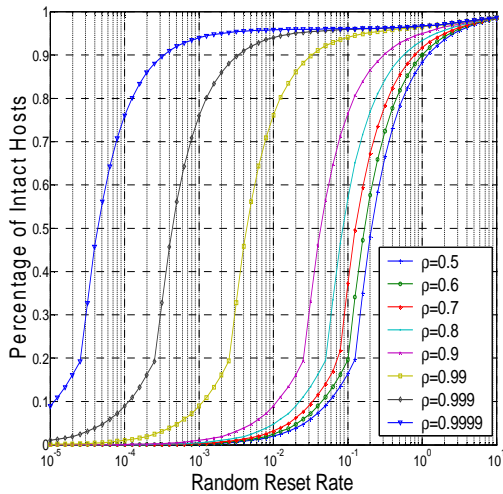


Figure 6 Availability vs. True Positive Rates

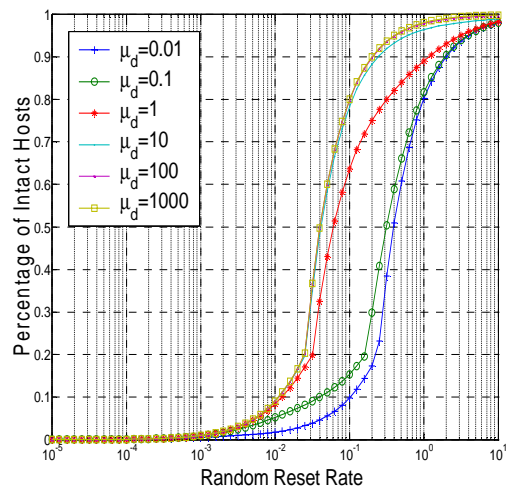
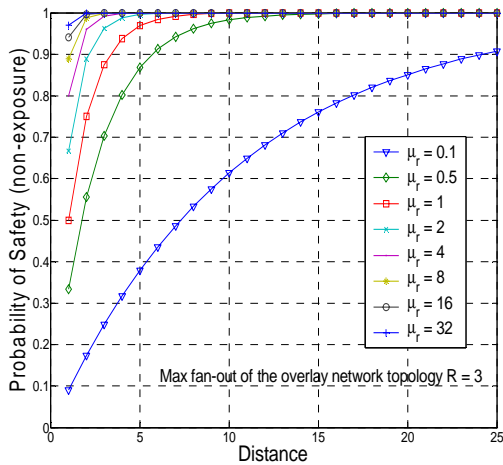


Figure 7 Availability vs. Detection Speed

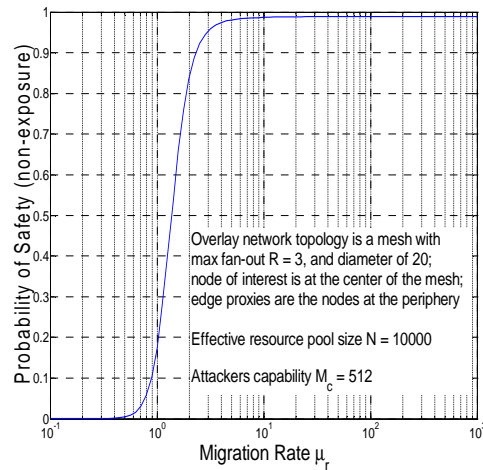
## 4.3 Effectiveness of Random Proxy Migration

❖ *Random proxy migration effectively avoiding core exposure*

Assuming that proxies randomly migrate among hosts in the resource pool, we fix the overlay network topology, resource pool size and capability of attackers, vary the migration rate (taking compromise rate as baseline, for example, “migration rate=0.1” means the migration rate is 10 times slower than compromise rate), and study how well random proxy migration scheme can stop exposure propagation and avoid node exposure.



**Figure 8 Exposure Propagation vs. migration rate  $\mu_r$**



**Figure 9 Node Exposure**

Figure 8 shows that random proxy migration can effectively stop exposure propagation. For example, when the migration rate is twice as fast as the compromise rate, it is virtually impossible for attackers to penetrate more than 5 hops from an exposed node. The scheme can effectively contain the impact of espionage attacks (or initial nodes exposure, such as edge proxies). Distance is an effective insulation for the critical nodes.

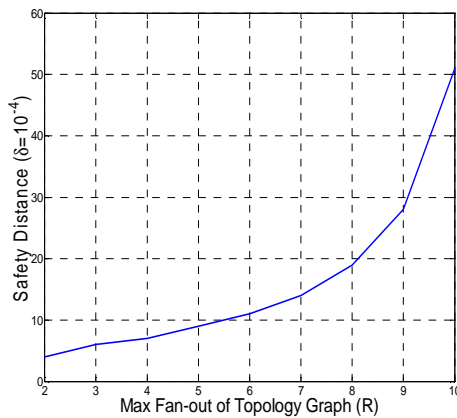
Figure 9 plots the probability of a specific node remaining unexposed in an overlay network. It shows that the higher the migration rate is, the more security we can have. For example, when the migration rate is 10 times faster than the compromise rate, the probability of the node being unexposed is about 0.99. It proves that the random proxy migration is an effective scheme to avoid node exposure.

#### 4.4 Overlay Network Topology

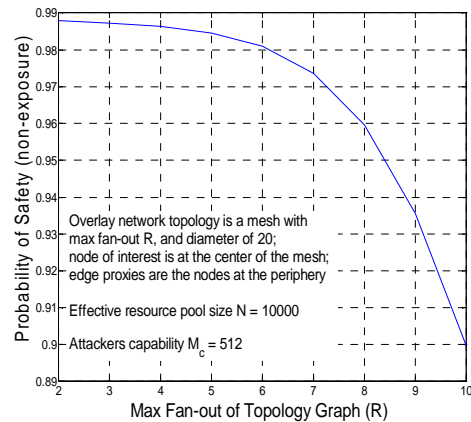
❖ Rich connectivity reduces the effectiveness of proxy network scheme to avoid core exposure

We fix the proxy migration rate and vary the max fan-out of the overlay network topology, and study how it affects the effectiveness of the scheme to stop exposure propagation and avoid node exposure. Claim II states that when  $(1-q)(R-1) < 1$  exposure propagation can be stopped. If that condition is not met, we may not be able to stop exposure propagation. Therefore R (max fan-out of the topology graph) should not be too big. Figure 10 and Figure 11 show that when R

increases the overall security gets worse. Intuitively,  $R$  represents the amount of information stored on each node in the overlay network. A larger  $R$  implies that one compromise may expose more information. Furthermore, rich connectivity implies shorter distances from the periphery (edge proxies) of the overlay network to the center (node of interest). Since distance is an effective insulation for the critical nodes, a large  $R$  jeopardizes the effectiveness of the proxy network scheme. A complete graph is not a good topology, and even  $\log N$  maximum degree networks[8] may not be good candidates.



**Figure 10 Exposure Propagation vs. Max Fan-out**



**Figure 11 Node Exposure**

## 5 Discussion

There are a few issues one should consider when applying our models.

First, all our analysis is based on the assumption that all compromises are independent. When hosts share same vulnerabilities, or flaws are not timely patched due to poor administration, attackers will effectively have a higher capability  $M_c$  or a higher compromise rate  $\lambda$ , or the Poisson model for host compromises is no longer appropriate.

Second, we did not consider the impact of DoS attacks on core exposure, assuming DoS attacks do not cause information disclosure. However in some DoS attacks, attackers may strategically flood certain parts of the network and reveal system information by observing the impact of such flooding. Our analysis does not address this attack.

Third, the size of the resource pool we use in the analysis is actually the number of potential hosts

we can use (observed by attackers). Having a large resource pool can effectively mitigate attack impact. By keeping the host IP addresses secret, we can effectively enlarge the pool size.

## **6 Related Works**

How to effectively resist denial-of-service attacks is still an open problem. There are many on-going studies in this area, which can be categorized into two approaches: preventive and tolerance approaches. Preventive approaches try to stop or deter attacks from the source, which include Intrusion detection systems[6, 7, 9-12], network ingress filtering[13] and IP trace-back schemes[14-17]. Tolerance approaches focus on mitigating the attack impact on the victim by means of system reconfiguration[18], resource isolation[18, 19] or load balance[20-23].

Many researchers are exploring the use of overlay networks to tolerate or avoid DoS attacks. The Secure Overlay Services (SOS) project[4] in Columbia University is one of them. They use Chord[8] in the overlay network to provide some amount of anonymity to hide the location of secret “servlets”. There are primitive analytical results about the system security under simple attack models such as DoS attack on individual hosts. However, the analysis is tied to their Chord-based SOS design and their attack models do not cover host compromise, which is a main threat. To our knowledge, our work is the first attempt of a thorough analysis in this area.

Here we study how to hide the application location. Interestingly a complementary problem, hiding the identity of the users, has been well studied since the early eighties. The solutions range from the early mix email server[24], to distributed Onion Routing schemes[25], and to the more recent Peer-to-Peer schemes such as Tarzan[26] and Pasta[27]. A key difference between the two problems is that there are many users in the system while there are only a handful of applications. Most of the schemes are based on the idea of mixing all input from all users so that an outsider cannot associate a particular message to a particular user. Another key difference is that user initiates the communication. In some schemes, such as Onion Routing, senders need to construct a route to the receiver before hand. These key differences make the two problems entirely different, and solutions in that area do not apply in our study.

## 7 Conclusion and Future Work

In this paper, we built a formal framework to rigorously study the properties of the system. Based on our analytical models, we have the following results. 1) intrusion detection triggered recovery strategy is insufficient to avoid resource depletion, 2) true-positive rates of intrusion detectors have more impact on resource availability than detection speed, 3) simple reconfiguration approaches, such as random proxy migration, can effectively prevent attackers from discovering applications' location, 4) overlay network topology is critical, choosing richly-connected topology may undermine proxy network's effectiveness to resist attacks.

**Future Directions:** There are at least three interesting directions. 1) How other (more sophisticated) forms of system reconfigurations affect system security and how they compare to the simple one studied here? 2) Attackers may attack a subset of nodes in the overlay network to cause denial-of-service to the application. How to reason about system properties against such attacks? 3) It is also worthwhile to study the correlation among failures. For example, how (DoS or compromise) attacks on one host affects other hosts in the resource pool.

### Acknowledgement

Supported in part by the Defense Advanced Research Projects Administration through United States Air Force Rome Laboratory Contracts AFRL F30602-99-1-0534 and the National Science Foundation thruNSF EIA-99-75020 Grads and NSF Cooperative Agreement ANI-0225642 (OptIPuter) to the University of California, San Diego Support from Hewlett-Packard is gratefully acknowledged.

### References

1. Fonseca, B., *Yahoo outage raises Web concerns*. 2000.
2. Williams, M., *EBay, Amazon, Buy.com hit by attacks*. 2000.
3. CERT, *"Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL*. 2001.
4. Keromytis, A.D., V. Misra, and D. Rubenstein. *SOS: Secure Overlay Services*. in *ACM SIGCOMM'02*. 2002. Pittsburgh, PA: ACM.
5. Harris, T.E., *The Theory of Branching Processes*. 1963: Prentice-Hall Inc.
6. Cowan, C., et al. *Automatic Detection and Prevention of Buffer-Overflow Attacks*. in *the 7th USENIX Security Symposium*. 1998. San Antonio, TX.
7. Wagner, D. and D. Dean. *Intrusion detection via static analysis*. in *2001 IEEE Symposium on Security and Privacy*. 2001. Oakland, CA, United States: Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy 2001..
8. Stoica, I., et al. *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*. in *ACM*

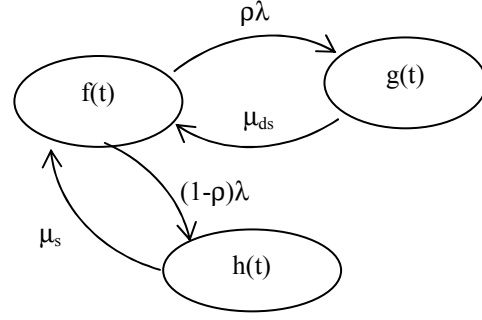
SIGCOMM'01. 2001.

9. Vigna, G. and R.A. Kemmerer, *NetSTAT: a network-based intrusion detection system*. Journal of Computer Security, 1999. 7(1): p. 37-71.
10. Axelsson, S., *Intrusion Detection Systems: A Survey and Taxonomy*. 2000, Chalmers University of Technology: Goteborg, Sweden.
11. Kim, G.H. and E.H. Spafford, *Experiences with Tripwire: Using Integrity Checkers for Intrusion Detection*. 1995, Purdue University.
12. Kumar, S. and E.H. Spafford. *A Pattern Matching Model For Misuse Intrusion Detection*. in *Proceedings of the 17th National Computer Security Conference*. 1994.
13. Ferguson, P. and D. Senie, *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. The Internet Society, 1998.
14. Snoeren, A.C., et al. *Hash-based IP traceback*. in *ACM SIGCOMM 2001- Applications, Technologies, Architectures, and Protocols for Computers Communications-*. 2001. San Diego, CA, United States: Computer Communication Review. v 31 n 4 2001.
15. Song, D.X. and A. Perrig. *Advanced and authenticated marking schemes for IP traceback*. in *20th Annual Joint Conference of the IEEE Computer and Communications Societies*. 2001. Anchorage, AK, United States: Proceedings - IEEE INFOCOM. v 2 2001.
16. Stone, R. *An IP Overlay Network for Tracking DoS Floods*. in *the 2000 USENIX Security Symposium*. 2000. Denver, CO.
17. Savage, S., et al., *Practical network support for IP traceback*. Computer Communication Review, 2000. 30(4): p. 295-306.
18. *Mutable Services*, New York University.
19. Spascheck, O. and L.L. Peterson. *Defending Against Denial of Service Attacks in Scout*. in *The 3rd symposium on operating systems design and implementation*. 1999.
20. Welsh, M., D. Culler, and E. Brewer. *SEDA: An Architecture for Well-Conditioned, Scalable Internet Services*. in *The 18th symposium on Operating Systems Principles*. 2001.
21. *Robust Networks*, Princeton University.
22. *Websphere Edge Services Architecture*, IBM.
23. *Network Load Balancing Technical Overview -- Microsoft Application Center*, Microsoft Corporation.
24. Chaum, D.L., *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*. Communications of the ACM, 1981. 24(2): p. 84-90.
25. Reed, M.G., P.F. Syverson, and D.M. Goldschlag, *Anonymous Connections and Onion Routing*. IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection, 1998.
26. Freedman, M.J., et al. *Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer*. in *1st International Workshop in Peer-to-Peer Systems (IPTPS'02)*. 2002. Cambridge, Massachusetts.
27. Elnikety, S., et al., *Pasta: Anonymous Peer-to-Peer Email System*. 2002, Rice University.

## Appendix

### ❖ Proof of Claim I:

It can be proved that the lower bound we study occurs when only  $A_{\text{rand}}$  attacks are present. Therefore, we only study the case where attacks are randomly and uniformly distributed. Figure 12



**Figure 12 host population**

summarizes the statistical models described in 3.1 (notations in Table 3-1).  $f(t)$  denotes the expected percentage of *intact hosts* over effective resource pool size  $N$ ;  $g(t)$  denotes the expected percentage (over  $N$ ) of the *compromised hosts* that can eventually be detected; and  $h(t)$  denotes the expected percentage (over  $N$ ) of the *compromised hosts* that can never be detected.  $\mu_{ds}$  approximates the combined impact of both recovery triggering strategy,  $\max(\mu_d, \mu_s) \leq \mu_{ds} \leq \mu_d + \mu_s$ . There are two cases:  $m > f(t)$ , denoted as world  $\Phi_1$  and  $f(t)$  in this case is denoted by  $f_{\Phi_1}(t)$ ;  $m \leq f(t)$ , denoted as world  $\Phi_2$  and  $f(t)$  in this case is denoted by  $f_{\Phi_2}(t)$ . In world  $\Phi_1$ , attackers can concurrently attack all *intact hosts*; in world  $\Phi_2$ ,  $m$  bounds attackers' capability.

**Lemma 1:**  $\lim_{t \rightarrow \infty} f_{\Phi_1}(t) = C_1$  when  $f(0) + g(0) + h(0) = 1$ , where  $C_1 = \frac{1}{1 + \frac{\lambda\rho}{\mu_{ds}} + \frac{\lambda(1-\rho)}{\mu_s}}$ .

Proof: From the Figure 12, we can get the following differential equations:

$$\begin{cases} \frac{df(t)}{dt} = -\lambda f(t) + \mu_{ds} g(t) + \mu_s h(t) \\ \frac{dg(t)}{dt} = \rho\lambda f(t) - \mu_{ds} g(t) \\ h(t) = 1 - f(t) - g(t) \end{cases} \quad \text{Solve them, and we can get the following result:}$$

$f_{\Phi_1}(t) = A_1 e^{\varphi_1 t} + B_1 e^{\varphi_2 t} + C_1$ , where  $\varphi_1 < 0$  and  $\varphi_2 < 0$  and  $C_1 = \frac{1}{1 + \frac{\lambda\rho}{\mu_{ds}} + \frac{\lambda(1-\rho)}{\mu_s}}$ . This result

holds when  $f(0) + g(0) + h(0) = 1$ .  $C_1, \varphi_1, \varphi_2, A_1, B_1$  are all constants, then Lemma 1 follows. ■

**Lemma 2:**  $\lim_{t \rightarrow \infty} f_{\Phi_2}(t) = C_2$ , where  $C_2 = 1 - \frac{\lambda\rho}{\mu_{ds}} m - \frac{\lambda(1-\rho)}{\mu_s} m$ . Furthermore, if  $f_{\Phi_2}(0) = 1$ ,

$g(0)=h(0)=0$ , then for any  $t>0$ ,  $f_{\Phi_2}(t)\geq C_2$ .

Proof: From Figure 12, we can get the following equations:

$$\begin{cases} \frac{df(t)}{dt} = -\lambda m + \mu_{ds} g(t) + \mu_s h(t) \\ \frac{dg(t)}{dt} = \rho \lambda m - \mu_{ds} g(t) \\ h(t) = 1 - f(t) - g(t) \end{cases} \quad \text{Solve them, we have } f_{\Phi_2}(t) = A_2 e^{-\mu_s t} + B_2 e^{-\mu_{ds} t} + C_2, \text{ where}$$

$$\begin{cases} A_2 = \frac{(1-\rho)\lambda}{\mu_s} m - h(0) \\ B_2 = \frac{\rho\lambda}{\mu_{ds}} m - g(0) \\ C_2 = 1 - \frac{(1-\rho)\lambda}{\mu_s} m - \frac{\rho\lambda}{\mu_{ds}} m \end{cases} \quad . \text{ If } g(0)=h(0)=0, \text{ then } A_2>0 \text{ and } B_2>0; \text{ therefore, for any } t>0,$$

$f_{\Phi_2}(t)>C_2$ . ■

Proof of Claim I: Let  $\varpi = \frac{\lambda\rho}{\mu_{ds}} + \frac{\lambda(1-\rho)}{\mu_s}$ , so  $C_1 = \frac{1}{1+\varpi}$  and  $C_2=1-\varpi m$ .

1)  $0\leq m\leq C_1$ , with some algebra we can get  $C_2\geq C_1$ . Because  $m<1$  and  $f(0)=1$ ,  $f(t)$  starts in world 2 and stays there as long as  $f(t)\geq m$ . From Lemma 2,  $\forall t>0$   $f(t)\geq C_2\geq C_1\geq m$ , it stays in  $\Phi_2$ . The first part of the claim is proved.

2)  $m>C_1$ , similarly we can get  $m>C_1>C_2$ . From Lemma 1, 2 we know that  $\exists t^*$  such that  $\forall t>t^*$   $f(t)=f_{\Phi_1}(t)$ . Therefore  $\lim_{t\rightarrow\infty} f(t) = \lim_{t\rightarrow\infty} f_{\Phi_1}(t) = C_1$ . ■

### ❖ Proof of Claim II and Claim III:

**Lemma 3:** *The probability of a node  $n$  being exposed-compromised,*

$$Prob_{exp\_com}(t) = Prob\{n \in \Omega_{exp\_com}(t) | n \in \Omega_{exp\_intact}(0)\} = (1-q)(1 - e^{-(\lambda+\mu_r)t}), \text{ where } q = \frac{\mu_r}{\lambda + \mu_r}.^5$$

$$\text{Proof: } 1 - Prob_{exp\_com}(t) = \int_0^t (1 - e^{-\mu_r x}) \lambda e^{-\lambda x} dx + \int_t^\infty \lambda e^{-\lambda x} dx = 1 - \frac{\lambda}{\lambda + \mu_r} (1 - e^{-(\lambda+\mu_r)t}).$$

Therefore,  $Prob_{exp\_com}(t) = (1-q)(1 - e^{-(\lambda+\mu_r)t})$ . Also  $\forall t<\infty$ ,  $1 - Prob_{exp\_com}(t) > q$ . In other words,  $q$  is the probability of  $n$  never being caught. ■

It can be proved that the asymptotic case of the exposure propagation problem is a Galton-

<sup>5</sup> Here we implicitly assume that the resource pool has sufficient amount of intact resource available. Results from section 3.2 address the validity of this assumption.

Watson branching process<sup>6</sup>, formulated as follows:

Let  $Z_d = |\Omega(n_e, d) \cap \Omega_{\text{exp}}(\infty)|$  ( $d = 0, 1, 2, \dots$ ) denote the number of (eventually) exposed nodes  $d$  hops away from  $n_e$ . From Lemma 3, the corresponding Galton-Watson branching process has a probability generating function  $\xi(s) = q + (1-q)s^{R-1}$ ,  $R$  is the max fan-out of the overlay network topology. Claim II and III directly follow from properties of branching process. Proofs and analysis of branching processes can be found in [5].

❖ **Proof of Claim IV:**

We consider the asymptotic case where attackers know the overlay network topology. There are  $M_c$  distinct attackers, each attacking one host. Consider the scenario that attackers start from random targets and each attacker does  $A_{\text{top}}$  attack independently. It can be shown that if  $N_i$  is a non-decreasing series, then this is an upper bound scenario for attackers' impact. Consider a specific attacker  $X$ , the probability of  $n_0$  not being exposed by  $X$  is  $\sum_{i=0}^{\infty} \frac{N_i}{N} \xi_i(0)$  (from Claim III).

Therefore, the probability of  $n_0$  not being exposed by any of the  $M_c$  independent attackers is at

$$\text{least} \left( \sum_{i=0}^{\infty} \frac{N_i}{N} \xi_i(0) \right)^{M_c}. \quad \blacksquare$$

---

<sup>6</sup> Francis Galton first introduced Galton-Watson branching process. He formulated the problem of the extinction of families as follows:

*“Let  $p_0, p_1, p_2, \dots$  be the respective probabilities that a man has 0, 1, 2, ... sons, let each son have the same probability for sons of his own, and so on. What is the probability that the male line is extinct after  $r$  generations, and more generally what is the probability for any given number of descendants in the male line in any given generation?”*

Additionally, in this model, different individual reproduce independently of one another. As a convention we denote by  $Z_0, Z_1, Z_2, \dots$  the number of individuals in the 0<sup>th</sup>, first, second, ... generations.

The asymptotic case of our overlay network topology is an  $R$ -ary tree with the root node initially exposed, and we want to study the number of exposed nodes at each level of the tree.