Analyzing Secure Overlay Forwarding Systems under Intelligent DDoS Attacks

Xun Wang, Sriram Chellappan, Phillip Boyer and Dong Xuan

Abstract

In the framework of a set of clients communicating with a critical server over the Internet, a recent approach to protect communication from Distributed Denial of Service (DDoS) attacks involves the construction of overlay systems. SOS, MAYDAY and I3 are such systems. The overlay system serves as an intermediate forwarding system between the clients and the server, which typically have fixed architectures that employ a set of overlay nodes arranged in different layers that control access to the server. Although such systems perform well under simple congestion-based DDoS attacks, it is questionable whether they are resilient to intelligent DDoS attacks which aim to infer architectures of the systems to launch more efficient attacks. In this paper, we define several intelligent DDoS attack models and develop an analytical approach to study the impacts of architectural design features on the system performance in terms of path availability. Our data clearly demonstrate that the system performance is indeed sensitive to the architectural features and the different features interact with each other to impact overall system performance under intelligent DDoS attacks. Our observations provide important guidelines in design of such secure overlay systems.

Index Terms

Secure Overlay Forwarding System, DDoS attacks

Xun Wang, Sriram Chellappan, Corey Boyer and Dong Xuan are with the Department of Computer Science and Engineering, The Ohio-State University, Columbus, OH 43210. E-mail: {wangxu, chellapp, boyerp, xuan}@cse.ohio-state.edu.

An earlier version of this work was published in the *Proceedings of IEEE ICDCS*, Tokyo, Japan, March 2004. This work was partially supported by NSF under grant No. ACI-0329155. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

I. Introduction

Distributed Denial of Service (DDoS) attacks are currently major threats to communication in the Internet. Current level of sophistication in system resilience to Distributed Denial of Services (DDoS) attacks is far from definite. Tremendous amount of research is being done in order to improve the system security under DDoS attacks [1], [2], [3], [4]. The reliability of communication over the Internet is not only important but also required. Typical examples of such applications are emergency, medical, and other related services. The system needs to be resilient to attacks from malicious users within and outside of the system that aim to disrupt communication. It is also very likely that there are certain special nodes or hot spots in such systems, and attacks on such nodes could prove to be catastrophic to the systems.

A recent body of work in the realm of protecting communication between a set of clients and a server employs the use of proactive defense mechanisms using overlay based structures. Typically a set of nodes that make up the overlay act as intermediate forwarders between the clients and the server. The design rationale in all these systems is to ensure; the server is effectively hidden from external clients, the presence of multiple paths to improve reliability, access control to prevent illegitimate users from being serviced, effectively dropping attack traffic far away from the server. The final objective though is to ensure that there are high degrees of path availabilities from clients to the server even when malicious entities attempt to compromise the communication [1]. Popular systems using the overlay based structure to defend communications against DDoS attack include SOS [5], Mayday [6] and I3 [7] etc. Such systems perform quite well in terms of path availabilities under random congestion based DDoS attacks. However, on a critical note, the following questions are naturally raised while analyzing the systems.

- The system can be targeted by *intelligent* attackers. That is, the attacker can *know* or *infer* the structure already present in the system. Knowledge of the existing structure can be taken advantage of by the attacker causing extensive damage to the system. An intelligent attacker can have the ability to traverse the communication chain between clients and the server. A sequence of such executions

[1]Throughout this paper this will be our performance metric. A formal definition of the performance metric is given in Section III.

can eventually disclose the server. Apart from this the attacker can also have the ability to congest nodes strategically, rather than just randomly. The question here is; how sensitive is the performance of such overlay based systems when targeted by intelligent DDoS attacks that take advantage of the structure already present?

- The structure of such overlay based system consists of certain design features [2]. A careful study of each system reveals that the design features can be generalized as the layering (number of layers), mapping degree (number of next layer neighbors per node, which can be also called connectivity degree), node distribution (number of nodes in a layer). The sensitivity of system performance to system structure is typically the sensitivity of performance to these design features under attacks. In SOS [5], the number of layers is set as 3. During analysis, the authors assume the neighbors of a node are all the nodes at the next layer. Are these the best choices? More interestingly, how do these design features interact with each other to impact system performance under different intensities of intelligent DDoS attacks?

In this paper, we aim to address the above issues. That is, our objective is to study the impact of the design features of overlay based intermediate forwarding systems on system performance under intelligent DDoS attacks. Towards this extent, we first generalize the deign features of the above overlay forwarding systems. We denote our generalized system as Secure Overlay Forwarding Systems (SOFS). We then define two classes of intelligent DDoS attack models. Our attack models take advantage of system structure while executing the attacks. We develop an analytical approach in the absence of system repair, and use simulations in the presence of system repair to analyze the sensitivity of system performance to each design feature of the SOFS system under these attack models. Our results show that the system performance is indeed very sensitive to the design features under intelligent DDoS attacks. We observe that the number of layers and number of neighbors per node (the mapping degree) have opposite effects on improving resilience to the attacks. In order to compensate for the effects of attacks, there is a clear trade-off in

[2]We use the term design features, architectural features and structural features interchangeably in this paper.

the layering as well as the mapping degree. We also observe that the system performance is sensitive to the node distribution per layer, particularly when the mapping degree is large. The conclusions we derive from our analysis in the absence of system repair hold true also in the case of system repair demonstrating the sensitiveness of the design features of the system in ensuring good performance. We believe that our findings can provide important guidelines in design of secure overlay forwarding systems.

II. The SOFS Architecture and Intelligent DDoS Attacks

A. The SOFS Architecture

In its most basic version, the SOFS architecture consists of a set of overlay nodes arranged in layers as shown in Fig. 1. The nodes in these layers serve as an intermediary between the clients and the critical target [3]. Such a system has three distinguishable design features. They are Layering, Mapping (Connectivity) degree and Node distribution across layers. A clearer description is given below.

- Number of Layers: The number of layers in the architecture is an estimate of the depth of control during access to the target. If the number of layers is L, then clients must pass through these L layers before communicating with the target. The importance of layering is that if the number of layers is large enough, implicitly it means that the target if better hidden against external clients.
- Mapping (Connectivity) degree: Each node in layer i routes to node(s) in layer $i+1$ towards the target to complete the communication chain. The mapping degree in the SOFS architecture is a measure of the number of neighbors a node in layer i has in layer $i+1$. Typically, the larger the mapping degree is, the more reliable is the communication due to the availability of multiple paths. The largest is actually 1 *to all*, where each node in layer i has all nodes in layer $i+1$ as its neighbors.
- Node Distribution: Node distribution is a measure of the number of nodes in each layer. Intuitively it may seem that uniform node distributions across layers is preferred to ensure a degree of load balancing in the system. However, for a fixed amount of nodes to be distributed across a fixed

[3]We use the term target and server interchangeably in this paper.

number of layers, it may be advisable to deploy more nodes at layers closer to the target to increase defenses in *sensitive* layers nearer the target.

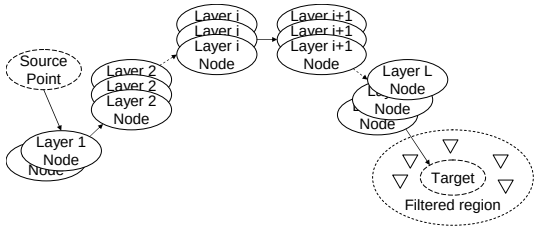


Fig. 1. The generalized SOS architecture.

A client that wishes to communicate with the target first contacts node(s) in the first layer which contact node(s) in the second layer and so on till the traffic reaches the target. In this architecture each node is only aware of nodes in its neighboring layer. A set of filters acts a firewall surrounding the target through which only legitimate traffic is allowed.

B. Intelligent DDoS Attack Models

The attacker in our model is intelligent. It has the ability to break-into nodes in the system with the intention of disclosing the victim's next-layer neighbors in the system. The attacker also has the ability to congest nodes in the overlay thereby denying service to legitimate clients. The attacker may be aware of the identities of a part of the nodes in the system. Since the goal of the attacker is to disclose as many nodes as possible and congest nodes in the overlay, the following are the two types of attacks conducted.

- Break-in Attacks: The attacker has the ability to attempt to break-into nodes in the SOFS system. A successful break-in results in dysfunction of the victim node and disclosure of the neighbors of the victim node. The goal of the attacker here is to disclose as many nodes in the overlay as possible and maybe disclose even the target.
- Congestion Attacks: The attacker has the ability to congest nodes in the SOFS system by bombarding them with traffic such that the node cannot serve legitimate client requests. Typically, nodes whose identities are known to the attacker will be preferred targets for congestion.

The above two attacks can be conducted in several possible ways. However, keeping in mind the above attack types, and the with the intention of maximizing attack impacts, the attacker will usually first conduct break-in attacks to disclose the identities of many nodes. Congestion attacks on the disclosed nodes then follow after the break-in attacks. In this realm, we define two attack models below.

- A discrete round based attack model: In this model, the attacker expends its break-in attack resources in a round by round fashion, with a part of its resources used in each round. The rationale is that, by successively breaking into nodes and locating their neighbors, the attacker can disclose many nodes. We call this model as *discrete* because, here the attacker starts a fresh round only after the results of all attempted break-ins in the current round are available to it. Congestion attacks follow next, and are conducted in one round.
- A continuous attack model: In this model, the attacker attempts to disclose some nodes first, using part of its break-in attack resources. However in this model, the attacker continuously keeps breaking into disclosed nodes as and when they become available. Congestion attacks follow next in a similar fashion. The attack models are described in more detail in our analysis in Sections III and IV.

We wish to emphasize here that the SOFS system also has the repair ability to defend against attacks. However any meaningful execution of the repair mechanism is contingent on the attacks. In some cases, the system may not be able to conduct any effective repair if the attacker can speedily conduct its attack, disrupting system performance for some short duration of time. However, if the attack is slow enough the system can attempt to take effective repair action to restore performance. More details on system repair are given in Section IV.

In this paper, we study the SOFS system performance under discrete round based attacks and continues attacks. We demonstrate that system performance is sensitive to design features and attacks, and the architecture needs to be flexible in order to realize better performance under different attacks.

III. Analysis of the SOFS Architecture under Round based Attacks

In this section we conduct an extensive mathematical analysis on our SOFS architecture under a discrete round based attack model with no system repair. In our analysis, the system we study consists of a total of N overlay nodes. These N nodes can be *active* or *dormant*. By *active* we mean that the nodes are currently in the overlay ready to serve legitimate requests [4]. A *dormant* node is one that is a part of the system but currently is not serving requests. We denote the number of *active* nodes in the system (also called as SOFS nodes) by n which is distributed across L layers. Layer i has n_i nodes and $\sum_i^L n_i = n$. It is reasonable to assume that the attacker resources are limited. We assume that the attacker can launch break-in attacks on N_T nodes and congestion attacks on N_C nodes. With a probability P_B, the attacker can successfully break into a node. In this section, the SOFS system we study either has no mechanism to repair a compromised active node (broken-in or congested), or it has no time or resources to do so because of the quickness of round based attacks and the cost of repair. We define system performance as the probability P_S that a client can find a path to communicate with the target under on-going attacks.

A. Under a One-burst Round Based Attack

1) Attack Model: In this attack model, which is an instance of the discrete round based attack model, the number of rounds is 1. The attacker will spend all the break-in attack resources randomly in one round and then launch the congestion attack. Even though this model may appear simple, in reality such a type of attack is possible when say, the system is in a high state of alert anticipating imminent attacks of which the attacker is aware of and still wishes to proceed with the attack. Here we assume the attacker has no prior knowledge about the SOFS nodes.

2) Analysis: The key defining feature of our analysis is in determining the set [5] of attacked nodes in each layer. The intuitive way to analyze the system is to list all possible combinations of attacked nodes in each layer. The overall system performance can be obtained by calculating the probability of occurrence

[4] In the remaining of the paper, if the context is clear, we will just use 'node' to represent active nodes.

[5] We use the term *set* and *number of nodes in a set* interchangeably.

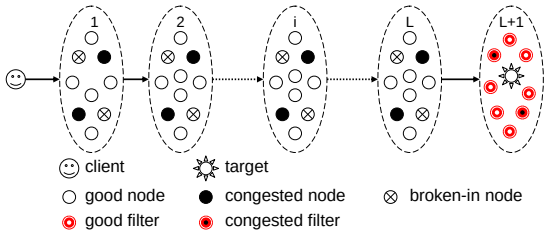


Fig. 2. A Snapshot of the generalized SOFS architecture under the intelligent DDoS attacks.

of each combination and calculating P_S for that combination and appropriately summarizing P_S over all possible combinations. It is easy to see that there could be many such possible combinations. For a system with L layers and n nodes evenly distributed, such combinations will be in $\theta(\frac{n}{L})^{2L}$. For a system 3 layers and 100 SOFS nodes evenly distributed, we have about $1.0 * 10^{10}$ combinations. Practically, it is not scalable to analyze the system in this fashion. To circumvent the scalability problem, we take an alternative approach. Since the system and attack parameters N, n, N_C, N_T are large, based on the weak law of large number, we use the average case analysis approach. We calculate the average number of attacked nodes in each layer to obtain P_S.

In our architecture, a node maintains a neighbor table that consists of nodes in its next higher layer and the number of neighbors is decided by the mapping degree policy. Upon receiving a message, a node in Layer i will contact a node in Layer $i+1$ from its neighbor table and forward the received message to that node. This process repeats till the target is reached via the nodes in successive higher layers. The routing thus takes place in a distributed fashion. We call a *bad* or *compromised* node as one that has either been broken into or is congested and cannot route a message. The other overlay nodes are *good* nodes. The routing table will contain *bad* entries during break-in or congestion attacks that can cause failure of a message being delivered. A snapshot of the system under an on-going attack is shown in Fig. 2. To compute P_S, first we should know the probability P_i that a message can be successfully forwarded from Layer $i-1$ to Layer i ($1 \leq i \leq L+1$). Here Layer $L+1$ refers to the set of filters that encompass the target. In our analysis ,we consider this layer also because it is possible that their identities can be

disclosed during a successful break-in at Layer L. With the property of distributed routing algorithm, we can obtain P_S by direct product of all P_i's, i.e., $P_S = \Pi_{i=1}^{L+1} P_i$. Obviously, P_i depends on the availability of good nodes in Layer i that are in the routing table of nodes in Layer $i-1$. Towards this extent, define $P(x, y, z)$ as the probability that a set of y nodes selected at random from $x > y$ nodes contains a specific subset of z nodes, then $P(x, y, z) = \begin{pmatrix} y \\ z \end{pmatrix} / \begin{pmatrix} x \\ z \end{pmatrix}$ if $y \geq z$, and otherwise $P(x, y, z) = 0$. Define s_i as the number of bad nodes in Layer i. Recall that each node in Layer $i-1$ will have m_i neighbors in Layer i. Then, on an average $P(n_i, s_i, m_i)$ is the probability that all next-hop neighbors in Layer i of an overlay node in Layer $i-1$ are bad nodes. Hence $P_i = 1 - P(n_i, s_i, m_i)$. Thus, the probability P_S that each message will be successfully received by the target can be expressed as follows:

$$P_S = \Pi_{i=1}^{L+1} P_i = \Pi_{i=1}^{L+1} (1 - P(n_i, s_i, m_i)). \tag{1}$$

In (1), only s_i's are undetermined. Recall that a bad node is one that has either been broken-into or is congested. If we define b_i and c_i as the number of nodes that have been broken-into and the number of congested nodes respectively in Layer i, we have $s_i = b_i + c_i$.

The nodes that were broken in will disclose some SOFS nodes. In our model, once a node is broken into, it is compromised and the attacker will not need to congest that node. Thus at the end of the break-in attack phase, there is a set of nodes disclosed, from which we have to discount nodes that have been successfully broken into. The resulting set of nodes is the one the attacker will try to congest first.

We assume the N_T break-in trials are uniformly distributed on the nodes in the system. The average number of broken-in overlay nodes, $N_B = P_B \frac{n}{N} N_T$. We define h_i as the number of nodes on which a break-in attempt has been made in Layer i. For Layer i, $h_i = \frac{n_i}{N}(N_T)$, and $b_i = P_B(\frac{n_i}{N})(N_T)$ for $i = 1, \ldots, L$. We assume here that the filters are special and cannot be broken into. Hence $b_{L+1} = 0$.

At the start of the congestion attack phase, the attacker needs to know the set of nodes disclosed which have not been attempted to break into. We calculate this set as follows. Let $Y_{i,j}$ be a random variable whose value is 1 when the j^{th} node in Layer i is either a disclosed node or one on which a break-in

attempt has been made. Let z_i denote the average number of nodes that have been disclosed or have been tried to be broken into. Thus,

$$z_i = E(\sum_{j=1}^{n_i} Y_{i,j}) = \sum_{j=1}^{n_i} E(Y_{i,j}) = \sum_{j=1}^{n_i} \Pr\{Y_{i,j} = 1\}. \tag{2}$$

The probability that the j^{th} node in Layer i is neither a disclosed node nor one on which a break-in attempt has been made is given by $(1 - \frac{m_i}{n_i})^{b_{i-1}}(1 - \frac{h_i}{n_i})$. The same node can be disclosed by more than one node in the previous layer. The part $(1 - \frac{m_i}{n_i})^{b_{i-1}}$ excludes such overlaps.

$$\Pr\{Y_{i,j} = 1\} = 1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}(1 - \frac{h_i}{n_i}), \tag{3}$$

and then z_i is given by,

$$\begin{aligned} z_i &= \sum_{j=1}^{n_i}(1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}(1 - \frac{h_i}{n_i})) & (4)\\ &= n_i(1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}(1 - \frac{h_i}{n_i})). & (5) \end{aligned}$$

We denote d_i^N the number of nodes which are disclosed but haven't been attempted to break-in:

$$d_i^N = z_i - h_i = n_i(1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}(1 - \frac{h_i}{n_i})) - h_i, \tag{6}$$

for $i = 2, \ldots, L+1$. Apart from d_i^N, there is a set of nodes that have been disclosed on which a break-in attempt has been made un-successfully. This set is denoted by d_i^A and is given by,

$$\begin{aligned} d_i^A &= \sum_{j=1}^{h_i - b_i}(1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}) & (7)\\ &= (h_i - b_i)(1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}). & (8) \end{aligned}$$

Note that nodes in the first layer cannot be disclosed due to a break-in attack and so $d_1^N = d_1^A = 0$.

Thus the attacker will congest nodes in the set d_i^N and d_i^A as their identities have been disclosed and they have not been broken into. Define N_D to be the average total number of nodes that are disclosed but not broken-into successfully in the system, where $N_D = \sum_{i=1}^{L+1}(d_i^N + d_i^A)$. Recall that N_C is the overall number of overlay nodes that the adversary can congest. Considering the attack congestion mechanism, there are two cases:

- $N_C \geq N_D$: In this case, all N_D disclosed nodes will be congested. Since the attacker still has capacity to congest $N_C - N_D$ nodes, it will expend its spare resources randomly. The extra congested nodes will be uniformly randomly chosen from the remaining $N - N_B - (N_D - d_{L+1}^N - d_{L+1}^A)$ good nodes. We emphasize that d_{L+1}^N and d_{L+1}^A are part of the filters and are excluded from N_D to determine the remaining overlay nodes that are targets for random congestion [6]. Therefore, the total number of congested overlay nodes in Layer i is,

$$c_i = \begin{cases} d_i^N + d_i^A + (N_C - N_D) * \\ \quad \frac{n_i - b_i^A - d_i^N - d_i^A}{N - N_B - (N_D - d_{L+1}^N - d_{L+1}^A)}, & i = 1, \ldots, L, \\ d_i^N, & i = L+1. \end{cases} \tag{9}$$

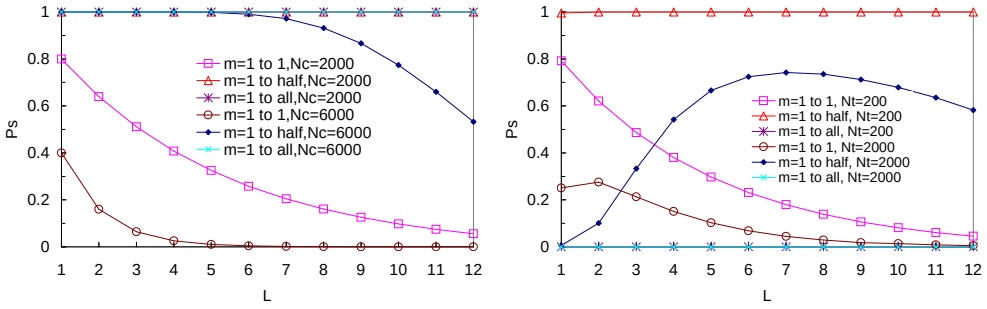
- $N_C < N_D$: The attacker can randomly congest the subset of N_C candidates among N_D disclosed nodes. In this case,

$$c_i = \frac{N_C}{N_D}(d_i^N + d_i^A), \ i = 1, 2, \ldots, L+1 \tag{10}$$

Recall that $s_i = b_i + c_i$ is the set of bad nodes in Layer i. We then use Formula (1) to compute P_S.

3) Numerical Results and Discussion: Fig. 3 shows the relationship between P_S and the layering and mapping degree under different attack intensities. We discuss the issue of node distribution in the successive round-based attack model. The mapping degrees used here are: one to one mapping which means each SOFS node has only one neighbor on the next layer; one to half mapping which means each node has half of all the nodes on the next layer as its neighbors; and one to all mapping which means each node has all the nodes on next layer as its neighbors. Other system and attack configuration parameters are: $N = 10000$, $n = 100$, $P_B = 0.5$, the SOFS nodes are evenly distributed among layers. The number of filters is set as 10. In Fig. 3 (a), N_T is set as 0 and we evaluate performance under two congestion intensities: $N_C = 2000$ and $N_C = 6000$ representing moderate and heavy congestion attacks respectively. In Fig. 3 (b), we fix $N_C = 2000$ and analyze two intensities of break-in: $N_T = 200$ and $N_T = 2000$. We make the following observations;

[6] In our model, the filters are special and can be congested only upon disclosure and not randomly.



(a) (b)

Fig. 3. Sensitivity of P_S to L and m_i under different attack intensities.

- Fig. 3 (a) shows that under the same attack intensities, different layer numbers result in different P_S. When $N_T = 0$ (pure random congestion attack), as L increases, P_S goes down. This is because, there are less nodes per layer, and under random congestion, few nodes per layer are left uncompromised. This behavior is more pronounced when the mapping degree is high. We wish to remind the reader about the SOS architecture, where the number of layers is fixed as 3 and the mapping degree is one to all for defending against random DDoS congestion attacks (same as the attack model we analyze here). From the above discussion we can see that fixing the number of layers as 3 is not the best solution for such a type of attack.
- For any L, a larger mapping degree (more neighbors for each node) means more paths from nodes in one layer to nodes in the next layer, thus increasing P_S in Fig. 3 (a) under the absence of break-in attacks. Under break-in attacks, a high mapping degree is not always good as more nodes are disclosed during break-in attacks. For instance when the mapping is one to all, $P_S = 0$ in Fig. 3 (b). Thus the effect of mapping typically depends on the attack intensities of the break-in and congestion phase.
- Finally we see that an increase in N_C and N_T naturally leads to a decrease in P_S, because more nodes could be congested or broken into.

B. Under a Successive Round Based Attack

1) Attack Model: Our successive round based attack (successive attack in short) model extends from the one-burst attack model in two ways: (1) the attacker has some prior knowledge about the first layer SOFS nodes that it uses to its advantage. Let P_E represent the percentage of nodes at the first layer known to the attacker before an attack, (2) the break-in attack phase is conducted in R rounds ($R > 1$), i.e., the attacker will launch its break-in attacks successively rather than in one burst. In this attack model, more SOFS nodes are disclosed in a round by round fashion thus accentuating the effect of attack.

The strategy of the successive attack is shown in Procedure 1. We denote β to be the available break-in resources at the start of each round and $\beta = N_T$ at the start of round 1. For each round, the attacker will try to break-into a minimum of α nodes and is fixed as $\frac{N_T}{R}$. If the number of disclosed nodes is more than α, the attacker *borrows* resources from β to attack them all. Otherwise it attacks the nodes disclosed and some other randomly chosen nodes to utilize α for that round. The spare break-in attack capacity available keeps decreasing till the attacker has exhausted all of its N_T resources. At any round, if the attacker has discovered more SOFS nodes than its available attack resources (β), it tries to break into a subset (β) of the disclosed nodes and starts the congestion phase. The attacker will congest all disclosed nodes and more; or only a subset of the disclosed nodes depending on its congestion capacity N_C. We assume X_j be the number of nodes whose identities are known to the attacker at the start of round j. Here we assume the attacker will not attempt to break into a node twice and a node broken into is not congested. Although there can be other variations of such successive attacks, we believe that our model is representative.

2) Analysis: We again use the average case approach to analyze the system and derive P_S. The problem typically is in discounting the overlaps among the bad (disclosed or broken-in) nodes. In the one-burst attack model we analyzed before, we had to take care of two possible overlap scenarios (1) a disclosed node could have been already broken-into, (2) the same node being disclosed by multiple lower layer nodes. The complexity in overlap is accentuated here due to the nature of successive attacks. This is

because there are multiple rounds of break-in attacks before congestion. We thus have to consider the above overlaps in the case of multiple rounds as well. In order to preserve the information about a node per round, we introduce the subscript j for round along with the subscript i that refers to layer information.

At the beginning of each round j, the attacker will base its attack on the set of nodes disclosed at the completion of round $j-1$. We denote the set of nodes which are disclosed at round $j-1$, on which break-in attempts is made in round j, as $h^D_{i,j}$. Depending on its spare capacity for that round, the attacker will also select more nodes to randomly break-into. We denote this set of nodes as $h^A_{i,j}$. We define $h_{i,j} = h^D_{i,j} + h^A_{i,j}$. It is the number of nodes on which break-in attempts (successfully/unsuccessfully) have been made at Layer i in round j. Once the attacker has launched its break-in attacks on these $h_{i,j}$ nodes, it will break into a set of nodes. We denote $b^D_{i,j}$ and $b^A_{i,j}$ as the set of nodes successfully broken into and denote $u^D_{i,j}$ and $u^A_{i,j}$ as the set of nodes unsuccessfully broken into after launching the break-in attacks on the $h^D_{i,j}$ and $h^A_{i,j}$ set of nodes respectively.

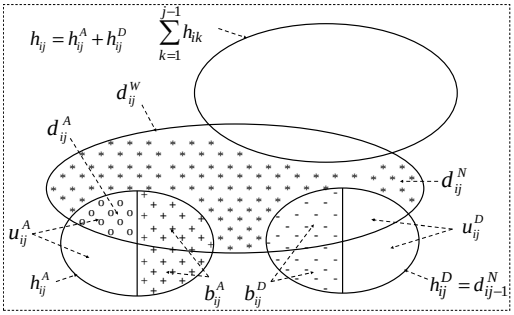


Fig. 4. Node demarcation in our successive attack at the end of Round j.

Breaking into nodes in sets $b^D_{i,j}$ and $b^A_{i,j}$ will disclose a set of nodes denoted by $d^W_{i,j}$. This set, $d^W_{i,j}$ will overlap with (1) the nodes attacked until all previous rounds denoted by $\sum_{k=1}^{j-1} h_{i,k}$, (2) the nodes in set $u^A_{i,j}$. We define such a set of the overlapped nodes as $d^A_{i,j}$, (3) the nodes in set $b^A_{i,j}$, (4) the nodes in set $b^D_{i,j}$ and $u^D_{i,j}$. Fig. 4 shows such overlaps at the end of round j. After discounting all the above overlaps from $d^W_{i,j}$, we can get the set of disclosed nodes which have never been attacked till the end of round j. We define this set as $d^N_{i,j}$. We define $X_{j+1} = \sum_i^L d^N_{i,j}$, on which the attacker will launch break-in attacks

at round $j+1$.

In the following, we proceed to describe the calculation of the above sets and then compute the number of congested nodes. Thus we typically compute s_i, and apply Formula (1) to obtain P_S. We would like to take case $X_j < \alpha < \beta_j$ in Algorithm 1 as an example. This is the most representative case among the ones possible. We also consider the other possible cases briefly after analyzing this case. In this case, the attacker at the beginning of round j of its break-in attack phase has resources to break into more nodes than those disclosed already prior to that round and has attack resources left, i.e. $\alpha - X_j$ to randomly attack other nodes.

a) The break-in attack phase: At the beginning of round j, the attacker will launch break-in attacks on the set of nodes disclosed in round $j-1$, i.e. $d^N_{i,j-1}$. The remaining break-in resources of that round will be randomly used. We then have,

$$h^D_{i,j} = d_{i,j-1}, \tag{11}$$

$$h^A_{i,j} = \frac{n_i - d_{i,j-1} - \sum_{k=1}^{j-1} h_{i,k}}{N - X_j - \sum_{q=1}^{L} \sum_{k=1}^{j-1} h_{q,k}} (\alpha - X_j), \tag{12}$$

$$h_{i,j} = h^A_{i,j} + h^D_{i,j}, \tag{13}$$

$$b^D_{i,j} = P_B * h^D_{i,j}, \tag{14}$$

$$b^A_{i,j} = P_B * h^A_{i,j}, \tag{15}$$

$$u^D_{i,j} = (1 - P_B) * h^D_{i,j}, \tag{16}$$

$$u^A_{i,j} = (1 - P_B) * h^A_{i,j}, \tag{17}$$

for $i = 1, 2, \ldots, L$.

In (11), note however that $d_{i,j-1}$ is 0 for $i = 1$. This is because the nodes at the first layer cannot be disclosed by means of a break-in attack in any round j. We define $b_{i,j}$ as the summation of $b^A_{i,j}$ and $b^D_{i,j}$. Hence we have,

$$\begin{aligned} b_{i,j} = {} & P_B * \frac{n_i - d_{i,j-1} - \sum_{k=1}^{j-1} h_{i,k}}{N - X_j - \sum_{q=1}^{L} \sum_{k=1}^{j-1} h_{q,k}} * (\alpha - X_j) \\ & + P_B * d_{i,j-1}, \qquad i = 1, 2, \ldots, L. \end{aligned} \tag{18}$$

The next part is to compute the set of nodes $d_{i,j}^N$ and $d_{i,j}^A$. As discussed above, we have to extract the set $d_{i,j}^N$ from $d_{i,j}^W$. Similar to the discussion in the one-burst attack case and from (5), (6) and (8), we calculate $d_{i,j}^N$ and $d_{i,j}^A$. We first calculate the set of nodes that have been either disclosed or attacked. This is given by,

$$z_{i,j} = n_i(1 - (1 - \frac{m_i}{n_i})^{b_{i-1,j}}(1 - \frac{\sum_{k=1}^{j} h_{i,k}}{n_i})), \tag{19}$$

for $b_{i-1,j} > 0$ and $i = 2, \ldots, L$. We then have,

$$d_{i,j}^N = z_{i,j} - \sum_{k=1}^{j} h_{i,k}, \tag{20}$$

for $b_{i-1,j} > 0$ and $i = 2, \ldots, L$. Note that in our attack model, the attacker will not try to break into a node twice. Hence, to calculate $d_{i,j}^N$, from $z_{i,j}$, we subtract the nodes on which a break-in attempt has been made. Similarly, we have,

$$d_{i,j}^A = (h_{i,j}^A - b_{i,j}^A)(1 - (1 - \frac{m_i}{n_i})^{b_{i-1,j}}), \tag{21}$$

for $b_{i-1,j} > 0$ and $i = 2, \ldots, L$.

We now wish to clarify the reader about the situations involving particular cases for the successive attack. Apart from the general case we have discussed, there are three other cases: (i) $X_j < \beta \leq \alpha$, (ii) $\alpha \leq X_j < \beta$, and (iii) $\beta \leq X_j$. For case (i), all the formulas we derived for the general case can be directly applied, except that α has to be replaced by β. For case (ii), all the formulas in the general case can be applied except that $h_{i,j}^A = 0$. For case (iii), we have $h_{i,j}^A$ = 0, and the formulas derived in the general case are not directly applicable. In this case, there are some disclosed nodes that the attacker does not try to break into due to consumption of all break-in resources. Such nodes will be attacked during the congestion phase. We denote this set of nodes in Layer i after round j as $f_{i,j}$. We wish to state here that $f_{i,j}$ has relevance (it could be non-zero) only when the attacker completes its break-in attack phase

at round j. Thus in this case we have,

$$f_{i,j} = d_{i,j-1} - (\frac{d_{i,j-1}}{X_j})\beta, \tag{22}$$

$$h^A_{i,j} = 0, \tag{23}$$

$$h^D_{i,j} = d_{i,j-1} - f_{i,j}, \tag{24}$$

for $i = 1, 2, \ldots, L$.

$$d^N_{i,j} = \begin{cases} 0, & i = 1, \\ n_i(1 - (1 - \frac{m_i}{n_i})^{b_{i-1,j}} \\ \quad (1 - \frac{\sum_{k=1}^{j} h_{i,k} + \sum_{k=1}^{j} f_{i,k}}{n_i})) \\ \quad - \sum_{k=1}^{j} h_{i,k} - \sum_{k=1}^{j} f_{i,k}, & i = 2, \ldots, L+1, \end{cases} \tag{25}$$

where $b_{i-1,j} > 0$, and $d^A_{i,j}$ is same as one in the general case.

b) The congestion attack phase: Let the final round of the break-in attack be $J(J \leq R)$. Defining N_D to be the number of disclosed nodes but not broken-into, we have,

$$\begin{aligned} N_D = & \sum_{i=1}^{L}\sum_{k=1}^{J} u^D_{i,k} + \sum_{k=1}^{J} d^N_{L+1,k} \\ & + \sum_{i=2}^{L} d^N_{i,J} + \sum_{i=1}^{L} f_{i,J} + \sum_{i=1}^{L}\sum_{k=1}^{J} d^A_{i,k}. \end{aligned} \tag{26}$$

We have the total number of broken-in nodes, $N_B = \sum_{i=1}^{L}\sum_{k=1}^{J} b_{i,k}$. If $N_C \geq N_D$, we have the number of congested nodes per layer, c_i as

$$c_i = \begin{cases} \sum_{k=1}^{J} u^D_{i,k} + d^N_{i,J} + \sum_{k=1}^{J} d^A_{i,k} \\ \quad + F_{i,J} + (N_C - N_D)(n_i \\ \quad - \sum_{k=1}^{J} b_{i,k} - \sum_{k=1}^{J} u^D_{i,k} - d^N_{i,J} \\ \quad - \sum_{k=1}^{J} d^A_{i,k} - F_{i,J})/(N \\ \quad - N_B - (N_D - \sum_{k=1}^{j} d^N_{L+1,k})), & i = 1, \ldots, L, \\ \sum_{k=1}^{j} d^N_{L+1,k}, & i = L+1. \end{cases} \tag{27}$$

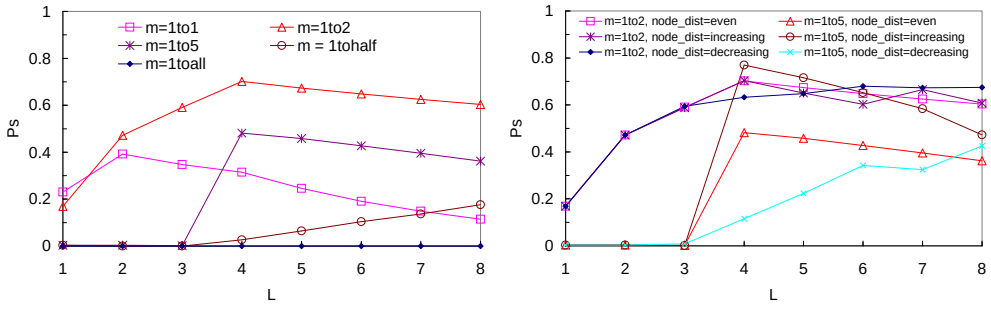
If $N_C < N_D$, we have

$$c_i = \begin{cases} \frac{N_C}{N_D} * (\sum_{k=1}^{J} u^D_{i,k} + d^N_{i,J} \\ \quad + F_{i,J} + \sum_{k=1}^{J} d^A_{i,k}), & i = 1, \ldots, L, \\ \frac{N_C}{N_D}(\sum_{k=1}^{J} d^N_{L+1,k}), & i = L+1. \end{cases} \tag{28}$$

Denoting $b_i = \sum_{k=1}^{J} b_{i,k}$, we have the set of bad nodes in Layer i, $s_i = b_i + c_i$. We then use Formula (1) to compute P_S.

Note that prior knowledge about identities of the first layer SOFS nodes, P_E, determines X_1, i.e. $X_1 = n_1 * P_E$. In fact, we can consider this information as that obtained from a break-in attack at *Round 0*. The number of nodes "disclosed" at *Round 0* is $n_1 * P_E$, all of which are distributed at the first layer. At round 1, the attacker will launch its break-in attack based on this information. Thus $b_{i,j}, d_{i,j}^N, c_i$ etc. can be calculated by application of Formulas (11) to (28). We wish to point out that if we set $P_E = 0$ and $R = 1$, the successive attack model degenerates into the one-burst attack model. Thus the formulas to compute $b_{i,j}, d_{i,j}^N, c_i$ etc. will be simplified to the corresponding ones derived in the previous sub-section.

3) Numerical Results and Discussions: In the following, we discuss the system performance (P_S) under the successive attack. Unless otherwise mentioned, the default system and attack parameters are $N = 10000$, $n = 100$, $N_C = 2000$, $N_T = 200$, $R = 3$, $P_B = 0.5$ and $P_E = 0.2$ and the SOFS nodes are evenly distributed among the layers. We introduce two new mapping degrees here, namely one to two mapping, meaning each SOFS node has 2 neighbors in the immediate higher layer; and the other is, one to five mapping, meaning each node has 5 neighbors in the next layer.



(a) (b)

Fig. 5. Sensitivity of P_S to L, m_i and node distribution.

Fig. 5 (a) shows the impact of L on P_S under different mapping degrees. Similar to Fig. 3 (a)(b), P_S is sensitive to L and the mapping degree, even when N_T is not zero and $R > 1$. Among the current configurations, the one with $L = 4$ and mapping degree one to two provides the best overall performance.

Fig. 5 (b) gives us an insight on the impact of node distribution on P_S when L and the mapping degree change. Other parameters remaining unchanged, here we show sensitivity of performance to three different node distributions per layer. The first is even node distribution wherein the nodes in each layer are the same (given by $\frac{n}{L}$). The second is increasing node distribution, wherein the number of nodes in the first layer are fixed ($\frac{n}{L}$). This is to maintain a degree of load balancing with the clients. The other layers have nodes in an increasing distribution of $1 : 2 : \ldots : L-1$. The third is decreasing node distribution where the number of nodes in the first layer is fixed ($\frac{n}{L}$) and those in the other layers are in decreasing order of $L-1 : L-2 : \ldots : 1$.

We make the following observations. The node distribution does impact performance. The sensitivity of P_S to the node distribution seems more pronounced for higher mapping degrees (more neighbors per node). A very interesting observation we make is that increasing node distributions performs best. This is because when the mapping degree is larger than one to one, breaking into one node will lead to multiple nodes being disclosed at the next layer, hence the layers closer to the target will have more nodes disclosed and are more vulnerable. More nodes at these layers can compensate the damage of disclosure . Also we observe that as the number of layers increases, the sensitivity to node distribution gradually reduces. This is because as L increases, the difference in the number of nodes per layer turns to be less for the different node distributions.

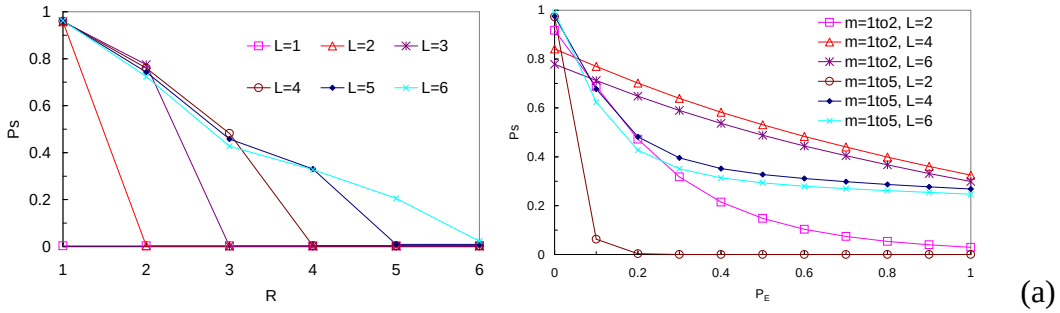
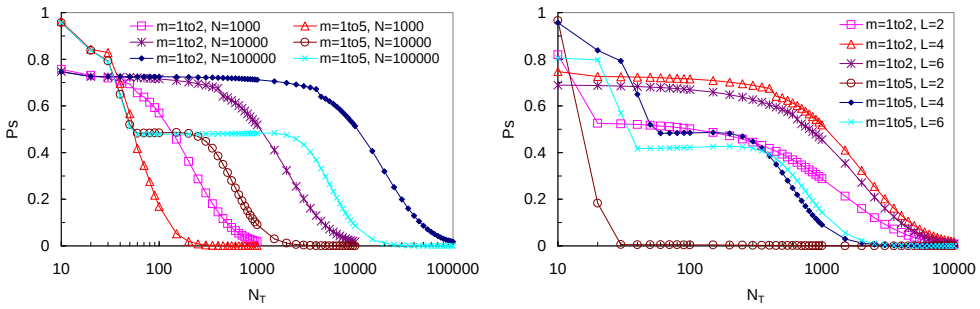


Fig. 6. Sensitivity of P_S to R (a) and P_E (b).

Fig. 6 (a) shows the impact of R on P_S under different L with mapping degree one to five. The nodes are

evenly distributed among the layers in this case. Overall, P_S is sensitive and decreases when R increases. For larger values of L, P_S is less sensitive to R because more layers can provide more protection from break-in attack even for higher round numbers. We also observe that P_S is sensitive to P_E in Fig Fig. 6 (b). For higher mapping degrees, P_S is more sensitive to changing P_E. The reason follows from previous discussions that a higher mapping degree discloses more nodes. For smaller L, P_S is more sensitive to changing P_E. The reason is, a smaller L increases the attacker's chance to penetrate the system, layer by layer.



(a) (b)

Fig. 7. Sensitivity of P_S to N_T under different L, m_i and N.

In Fig. 7 we show how P_S changes with N_T as the other system side parameters change. Fig. 7 (a) shows how the mapping and total number of overlay nodes influences the relation between N_T and P_S. In this configuration, we set $N_C = 2000$ and even SOFS node distribution. Fig. 7 (b) shows the sensitivity of P_S to changing L and mapping degrees under changing N_T. We make the following observations.

- P_S is sensitive to N_T. A larger N_T results in a smaller P_S. For higher mapping degrees, P_S is more sensitive to changing N_T. The reason follows from previous discussions that a higher mapping degree discloses more nodes under break-in attacks.
- From Fig. 7, there is portion of the curve, where P_S almost remains unchanged for increasing N_T. This stable part is due to advantages offered by means of the layering of SOFS architecture to disclosure-based break-in attack. The down slide in P_S beyond the stable part shows the effect of random break-in attack apart from disclosure-based attack.

- For a fixed N_T, an increase in the total number of overlay nodes N, decreases the chance that a random break-in attack is launched on an SOFS node, and P_S does increase.

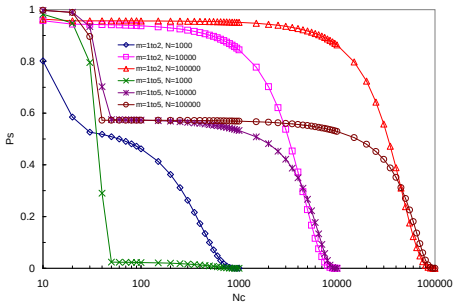


Fig. 8. Sensitivity of P_S to N_C under different m_i and N.

Fig. 8 shows the sensitivity of P_S to the changing of N_C under different mapping degree and N. Here we set $N_T = 200$.

- We observe that P_S is sensitive to N_C. As N_C increases, P_S is smaller. The sensitivity of P_S with small N_C shows that if N_C increases, more disclosed nodes can be congested, making P_S smaller. The stable part shows the protection of layering SOS architecture to disclosure-based congestion attack. The fall in P_S after the stable part of the curves shows the effect of randomly congestion attack beyond disclosure-based attack.
- If the mapping degree is larger, P_S is less sensitive to changes in N_C under the same N_T, especially when N_C is large. The reason is for higher mapping degree, more paths are available leading to more protection to congestion-based attack.
- For a fixed N_T, an increase in N increases P_S because more overlay nodes (N) makes the chance that a random attack targeting an SOS nodes small.

Fig. 9 illustrates the sensitivity of P_S to P_B with different L and mapping degrees. From this figure, we can find that P_S is sensitive to P_B. For higher mapping degrees and small , P_S is more sensitive to changing P_B. For smaller L, P_S is more sensitive to changing P_B. These observations are similar to the ones we obtained in Fig. 7. These findings can be explained as follows. As P_B increases, a SOFS node turns easier to be broken-in. Hence the consequence of increase in P_B is similar to increase in N_T. Hence

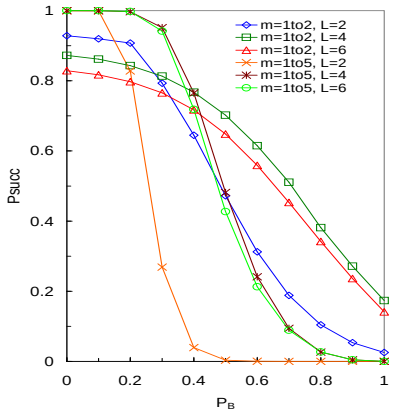


Fig. 9. Sensitivity of P_S to P_B different L and m_i.

we can have the similar observations.

We summarize our finding as follows. The attack strategies, intensities, prior knowledge about the system and robustness of SOFS nodes significantly impact system performance, however, the impacts are deeply influenced by the system design features. Larger values of L and smaller mapping degrees improve system resilience to break-in attacks, while the reverse is true for congestion based attacks. In order to compensate for the effects of break-in and congestion attacks, there is a clear trade-off in the layering as well as mapping degree. Thus, if the system is designed carefully keeping potential attack scenarios in mind, more resilient architectures can be designed.

IV. Analysis of the SOFS System under Continuous Attacks

In this section, we study the performance of the SOFS system in the presence of continuous attacks. Also, we study the impacts of system repair in this section. The performance metric P_S is still the same.

A. Attack Model and System Repair

The continuous attack model is different from the discrete round based model in the sense that the attacker continuously breaks into nodes as and when their identities are revealed to the attacker. We define N_T and N_C to be the maximum number of overlay nodes that can be simultaneously broken-into or congested. Furthermore, here the attacker can reuse its resources (N_T and N_C) in a more sophisticated way as follows. During system recovery (discussed next), the attacker will know the recovery of a compromised

node in time T_{dr}. If the attacker attacks a non-SOFS node [7], it will know that it is a non-SOFS node after time T_{ns}. Then it will redirect the attack after time T_{red}. An on-going congestion attack will keep sending traffic to a victim node as long as it is an SOFS node. Once a break-in attack is completed on a node, whether successfully or not, the attacker will redirect the break-in attack on another node. When the attacker redirects the attack, it will use the disclosed node list if there is any node in that list, otherwise it will randomly pick a node from all the overlay nodes except the nodes already under attack. Obviously, the disclosed nodes are all SOFS nodes, so they should be targeted by break-in attack if there are enough break-in attack resources. If the disclosed nodes number is larger than N_T, the congestion attack can also share the disclosed nodes list. That is, a part of the disclosed nodes will be congested, while the remaining will be broken into later.

We now discuss system recovery to defend against attacks. While there can be many potential recovery mechanisms, we generalize them into two categories, *proactive repair* and *reactive repair*. We consider a proactive repair mechanism that periodically resets every SOFS node. When proactive reset event happens, the SOFS system immediately replaces an old SOFS node with a new SOFS node which is pre-assigned. We assume that the interval between two successive proactive reset on one SOFS node be denoted as T_p. Besides *proactive repair*, the system can be *reactively* repaired. Reactive repair means that, once the SOFS node is attacked, the system will detect the attack on this node after a certain time, T_d. Then the system will immediately remove this SOFS node from the system and spend some time, T_{rp}, to find a substitute for this SOFS node.

B. Analysis

Our goal is to study the impacts of system design features on system performance under attacks. A mathematical analysis in the previous section gave us significant conclusions on system performance under discrete round based attacks without repair. A similar analysis here is too complicated to perform due to

[7]Recall that an SOFS node is one that currently is active in the overlay, while a non-SOFS node is one that is a part of the SOFS system, but is not a part of the overlay currently.

the complexities of the continues attack model coupled with our proposed approach for system repair. We thus use simulations in this section to study system performance.

In order to analyze the system, we implement a discrete event driven simulation tool to simulate the system repair and the attack model. The simulated system consists 5000 overlay nodes, among which there are 40 SOFS nodes, and 10 filters. Each client is connected to 5 first layer SOFS nodes. The simulation tool takes the delay values (e.g. T_p, T_d, T_{red} and etc.) and attack resources discussed in the previous section as part of the input parameters. We assume all the delays defined above follow exponential distribution. The *system recovery delay* over the *redirecting attack delay* is used to describe the speed competition between attacks and system repair, which is referred as r. With proactive repair, r is $\frac{T_p}{T_{red}}$; with reactive repaire, r is defined as $\frac{T_d}{T_{red}}$ In following simulations, the default system and attack parameters are $N_T = 200$, $N_C = 200$, $L = 4$, $P_B = 0.5$ and $P_E = 0.2$.

C. Numerical Results and Discussions

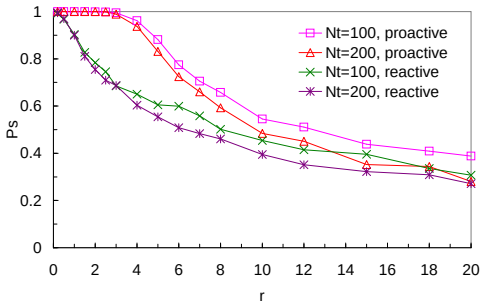
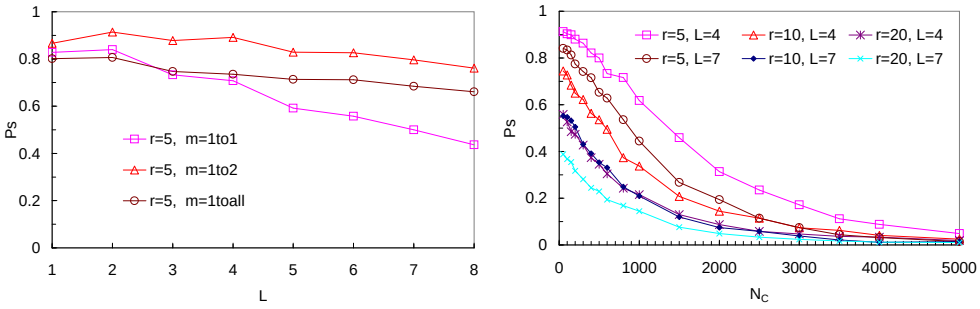


Fig. 10. Sensitivity of P_S to r under different N_T.

Fig. 10 shows the sensitivity of P_S to r, which represents speed competition between attacks and system repairObviously. Obviously, r is smaller, the system is faster in recovery, then the P_S should better. In Fig. 10, we change the r from 0 to 20 by changing T_p and T_d while fix T_{red}. $L = 4$ and the mapping used is one to half mapping in this figure. Another more important observation drawn from Fig. 10 is, the proactive repair's performance is better than reactive repair. This is easy to explain, (1) in proactive repair recovery scheme, a SOS node is replaced after every T_p, so the average delay after a attack till a proactive repair is half of T_p. In reactive repair scheme, the average delay between an attack event to

when system detect the attack is T_d. Thus, if T_p equals T_d, then proactive repair is actually faster than reactive repair. (2) in the reactive repair scheme, after the system detects the attack and shutdowns the compromised SOS node, it needs a period of time (T_rp) to find a substitute to replace the shutdown SOS node. Beyond the repair speed, the proactive repair scheme is cheaper, compared with the reactive repair which depends on the deployment of certain Intrusion Detection System. Thus, we believe the proactive repair scheme is better than the reactive scheme both in terms of performance and cost. Our following simulation will only use proactive reset scheme.

From our earlier observations in section III, we can see that of the three design features, the most sensitive ones are the number of layers and the mapping degree. In this section, we focus on these two important design features.



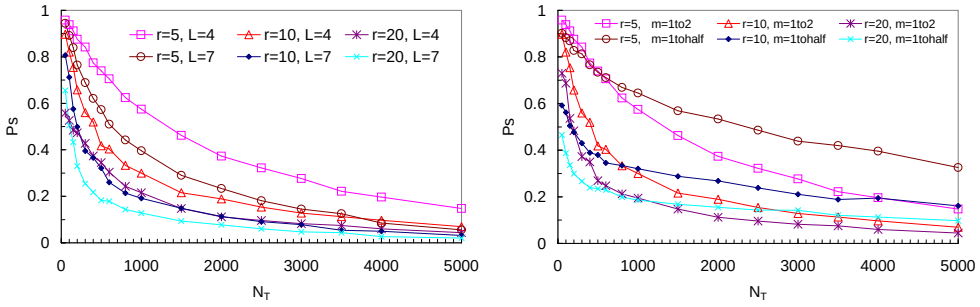
(a) (b)

Fig. 11. Sensitivity of P_S to L under different m (a), and to N_C under different L and r (b).

Fig. 11 (a) shows the impact of L on P_S under different mapping degrees when N_T and N_C are fixed as 200 respectively. Similar to Fig. 5 (a), P_S is sensitive to L and the mapping degree. The sensitivity of P_S to L and mapping degree is less than that in the previous model because here, while the attack breaks into nodes layer by layer, the system recovery replaces the compromised SOFS nodes and the disclosed un-attacked SOFS nodes. An increase in the number of layers can always slow the penetration of the break-in attack to the target. However, if the system deploys too many layers, it decreases the number of nodes on each layer and the number of paths between layers decreases correspondingly, which will cause a decrease in P_S (Recall that in our evaluation, the total number of SOFS nodes is fixed).

Note that in Fig. 11 (a), both N_T and N_C are set to a relatively small number, i.e. 200. As N_T and N_C increase, P_S will turn to be more sensitive to L and m, which is illustrated in Fig. 11 (b) and Fig. 12 (a), (b).

Fig. 11 (b) shows how L and r influence P_S when N_C changes, N_T is fixed as 200 and the mapping degree is fixed as one to two. Here $L = 4$ is always better than $L = 7$ because when N_T is fixed and N_C increases, $L = 7$ does not get so much benefit from delaying break-in attack when random congestion attacks dominate the overall attack. This also confirms that small number of layers provides better defense to congestion attack.



(a) (b)

Fig. 12. Sensitivity of P_S to N_T under different r and L (a), and different r and m (b).

Fig. 12 (a) shows how P_S changes with N_T as L and r change but the mapping degree here is fixed as one to two. When N_T is very small, a structure with more layers prevents the break-in attacks from penetrating the system, thus $L = 7$ is better than $L = 4$ especially when the system repair is very slow. But in all other cases, $L = 4$ is much better than $L = 7$ because deploying too many layers decreases the number of nodes on each layer, hence decreases the number of paths from clients to the target when there are more attacks.

Fig. 12 (b) shows how P_S changes with N_T under different mapping degrees and r when $L = 4$. We can make some interesting observations from this figure. P_S decreases rapidly when N_T increases from very small values, especially when the system repair is very slow, i.e. r is big. The system with the smaller mapping degree has better performance than the one with the bigger mapping degree. The observation

can be explained as follows. When r is big, the damage caused by the attack is recovered slowly and hence P_S decreases rapidly. Smaller mapping degree, less nodes disclosed and hence P_S is better. This finding matches the one we obtained in the round-based attack model.

However, when N_T is big enough, P_S decreases slowly with increase in N_T. More interestingly, P_S, with bigger mapping degrees is better than the one with smaller mapping degrees. This is opposite to the case when N_T is smaller. We can explain these observations as follows. When N_T is big enough, the attack totally overwhelms the system recovery and most of the SOFS nodes can be disclosed and compromised. Further increase of N_T will not make any bigger impact on P_S. Hence P_S decreases slowly as N_T increases. When N_T is very large, there are very few SOFS nodes alive in the system. It is the recovery process that maintains a certain number (possibly very small) of nodes alive, which guarantees P_S larger than 0. The number of alive nodes in this situation is mainly determined by r, which is not related to the mapping degree. Given a certain number of alive nodes, larger the mapping degree, more is the chance to find a path from a client to the target. Hence, large values of P_S can be achieved.

Based on the above simulations, we can make the following conclusions: the attack intensities and the system design features still have significant influence on system performance under continuous attacks with system repair. On the other hand, we find that the repair process does reduce the damage caused by the attack, particularly the break-in attack. Under very severe attacks, with the repair process, the system can maintain a certain level of system performance. The large mapping degree always helps to achieve better system performance in this circumstance.

V. Related Work

The main scope of this work is in the realm of overlay systems used for defending against Distributed DoS attacks. The surveys in [1], [2] on DDoS attacks and defense are exhaustive and interested readers can refer to those papers. In the following, we would like to focus on the work in overlay systems.

Overlay networks have been used for providing a wide variety of services like multicasting [8], [9], routing [10], [11] and file sharing [12], [13] etc. Recently, several works have been reported in the usage

of overlay solutions to enhance security of communication systems like [5], [6], [7], [14], [10]. An overlay solution to track DDoS floods has been proposed in [14]. [10] proposes a overlay routing infrastructure to enhance the resilience of the Internet. As mentioned above, [5], [6], [7] propose intermediate forwarding systems between the clients and the server to provide resilient communications under DDoS attacks. These systems are closely related to SOFS. We would like to discuss them further.

A Secure Overlay Services (SOS) architecture has been proposed in [5] in the framework of a set of clients communicating with a target during critical situations. The design rationale is to ensure that in the presence of DDoS attacks, the probability of all available paths between clients and the target being compromised is very small. In order to achieve this objective, the SOS architecture uses a set of overlay nodes arranged in 3 layers of hierarchy between the source and the target through which traffic is authenticated and then routed. Mayday [6] generalizes earlier work on SOS to provide denial of service resistance to servers. It improves upon the SOS work by separating the overlay routing and the lightweight packet filtering, and providing a more powerful set of choices for each layer. In [7], the authors propose a general, overlay-based Internet Indirection Infrastructure (I3) that offers a rendezvous-based communication architecture. Instead of explicitly sending a packet to a destination, each packet associated with an identifier will be forwarded to a rendezvous point. The receiver will then receive the message from that point. The SOFS system we study is a generalized architecture of the above systems.

Anonymity systems have several common features with our SOFS system. Anonymity systems usually use intermediate forwarding to achieve anonymity. However, there are some significant differences between these two types of systems. The goal of SOFS is to ensure that with high probability, any client can find a path to the target under DDoS attacks. Hiding the target (server) is an important approach to achieve its goal. Due to intermediate forwarding, the attacker has no idea about successive paths taken by messages, or the location of the target. However, besides this approach, SOFS also tries to ensure paths from client to the server by putting multiple connections between nodes in successive layers. Many anonymity systems, particularly the systems aiming to achieve receiver anonymity, depend on one or more third party nodes

to generate an anonymous path [15], [16], which is not good for SOFS. SOFS cannot rely on a centralized node to achieve receiver anonymity, since the centralized node can itself be the target of a DDoS attack. Our SOFS uses multiple layering technology to achieve receiver location anonymity in a distributed fashion.

Our work shares some similarity with the analysis in [5]. The performance metric, P_S is the same. However our work is significantly different from that work in the fact that the system and attacks we analyze are more general and sophisticated. We introduce intelligence to the DDoS attack by defining break-in attacks which can break into a node and disclose its neighbors. It is our belief that such attacks are practical and our data show that it is also very dangerous to the SOFS where there is a certain architecture present. The attacker can use the break-in attacks to infer the architecture of the system and launch congestion-based attacks more efficiently. Hence the approach used in [5] can not be applied in our work. Our analysis approach is much complicated, particularly in the case of multiple round based attacks, where we should consider a lot of overlaps among the attacked nodes. In the analysis of the original SOS architecture, it is assumed that each node can simultaneously provide the functionality of nodes at multiple layers. In the presence of break-in attacks, allowing this possibility is very dangerous in the sense that once such a node is broken-into, nodes in several other layers will be disclosed and we do not make this assumption.

VI. Final Remarks

In this paper, we have studied the impacts of architectural design features on SOFS, an generalized overlay intermediate forwarding system under intelligent DDoS attacks. We analyzed our SOFS system under a discrete round based attack using a general analytical approach, and analyzed the system under a continuous attack using simulations.

Our analysis results clearly demonstrate that design features are critical to ensure good system performance under intelligent DDoS attacks. Specifically, (1) larger number of layers and smaller mapping degrees improve system resilience to break-in attacks, while the reverse is true for congestion based

attacks, demonstrating the clear trade-off between these two features, (2) increasing node distributions perform better than other node distributions, (3) when the system tries to protect itself with a repair mechanism, the trends shown by our analysis without system repair still hold, although the attack impacts are reduced, and (4) Under extremely severe attacks, with system repair, the system can always maintain a certain level (possibly small) of system performance. The larger mapping degree is better in improving the system performance than the smaller mapping degree.

Our work thus has important impacts in the design of overlay systems to defend against DDoS attacks. The attack strategies, intensities, prior knowledge about the system and robustness of the overlay node significantly impact system performance. However with smarter handling of available resources in terms of designing good architectures, the impact of attacks can be significantly reduced. A part of future work along this direction is to study the impact of system design features on QoS, coupled with attack resilience. The work is challenging. An increase in the number of layers, while being more resilient to break-in attacks, increases the latency of communication. An increase in the mapping degree has the opposite effect of decreasing latency due to more choices for routing. We are in the process of designing an SOFS system that is highly resilient to attacks while still attempting to achieve a desired level of QoS.

The impacts of our work do not stop here. There are several other applications where a structure present, enables efficient delivery of services. These include Multicasting [8], [9], Real-time delivery [17], File Sharing systems [12], [13]. Similar to the discussion in this paper, attackers can cause significant damages to performance by exploiting the knowledge of structure already present in these systems. We believe that our work is a first step towards designing resilient architectures from this perspective of intelligent attacks. Analyzing the resilience of such systems under intelligent attacks will also be a part of our future work.

REFERENCES

[1] S. Savage, D. Wetherall, A. R. Karlin, and T. Anderson, "Practical network support for ip traceback," in *Proceedings of ACM SIGCOMM*, Stockholm, Sweden, August 2000.

[2] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," in *Proceedings of ACM SIGCOMM Computer Communication Review (CCR)*, Stockholm, Sweden, July 2002.

[3] K. Park and H. Lee, "On the eeffectiveness of route-based packet filtering for distributed dos attack prevention in power-law internets," in *Proceedings of ACM SIGCOMM*, San Diego, CA, August 2001.

[4] Aleksandar Kuzmanovic and Edward W. Knightly, "Low-rate tcp-targeted denial of service attacks (the shrew vs. the mice and elephants)," in *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[5] A. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure overlay services," in *Proceedings of ACM SIGCOMM*, Pittsburg, PA, August 2002.

[6] D. Andersen, "Mayday: Distributed filtering for internet services," in *Proceedings of the Usenix Symposium on Internet Technologies and Systems*, Seattle, WA, March 2003.

[7] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proceedings of ACM SIGCOMM Conference*, Pittsburge, PA, August 2002.

[8] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *Proceedings of ACM SIGMETRICS*, Santa Clara, CA, June 2000.

[9] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of ACM SIGCOMM Conference*, Pittsburgh, PA, August 2002.

[10] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of 18th ACM SOSP*, Banff, Canada, October 2001.

[11] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of internet path selection," in *Proceedings of ACM SIGCOMM Conference*, Cambridge, MA, August 1999.

[12] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An architecture for global-scale persistent storage," in *Proceedings of ASPLOS*, Cambridge, MA, November 2000.

[13] F. Dabek, M. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with cfs," in *Proceedings of ACM SOSP*, Chateau Lake Louise, Canada, October 2001.

[14] R. Stone, "Centertrack: An ip overlay network for tracking dos floods," in *9 th USENIX Security Symposium*, San Francisco, CA, August 2000.

[15] M. Reiter and A. Rubin, "Crowds: Anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, November 1998.

[16] L. Xiao, Z. Xu, and X. Zhang, "Mutual anonymity protocols for hybrid peer-to-peer systems," in *Proceedings of IEEE ICDCS*, Providence, RI, May 2003.

[17] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *Proceedings of IEEE INFOCOM*, San Francisco, CA, April 2003.

Procedure 1 Pseudocode of the successive attack strategy

System parameters: N, n, L, P_B; Attack parameters: N_T, N_C, R, X_1, β, α

Phase 1 :

1: $\beta = N_T$, $\alpha = \frac{N_T}{R}$;

2: **for** $j = 1$ to R **do**

3: **if** $X_j < \alpha < \beta$ **then**

4: launch break-in attack on all X_j nodes and randomly launch break-in attack on $\alpha - X_j$ nodes and calculate the set X_{j+1} disclosed nodes; update $\beta = \beta - \alpha$;

5: **end if**

6: **if** $X_j < \beta \leq \alpha$ **then**

7: launch break-in attack on all X_j nodes and randomly launch break-in attack on $\beta - X_j$ nodes and calculate the set X_{j+1} disclosed nodes; break;

8: **end if**

9: **if** $\alpha \leq X_j < \beta$ **then**

10: launch break-in attack on all X_j nodes and calculate the set X_{j+1} disclosed nodes; update $\beta = \beta - X_j$;

11: **end if**

12: **if** $X_j \geq \beta$ **then**

13: launch break-in attack on β nodes among X_j nodes and calculate the set X_{j+1} disclosed nodes; break;

14: **end if**

15: **end for**

16: calculate N_D;

Phase 2:

1: **if** $N_C > N_D$ **then**

2: congest the N_D nodes and randomly congest ($N_C - N_D$) nodes;

3: **else**

4: congestion N_C nodes among N_D nodes randomly;

5: **end if**