Analyzing the Secure Overlay Services Architecture under Intelligent DDoS Attacks

Dong Xuan, Sriram Chellappan, Xun Wang and Shengquan Wang*

Abstract

Distributed Denial of Service (DDoS) attacks are currently major threats to communication in the Internet. A secure overlay services (SOS) architecture has been proposed to provide reliable communication between clients and a target under DDoS attacks. The SOS architecture employs a set of overlay nodes arranged in three hierarchical layers that controls access to the target. Although the architecture is novel and works well under simple congestion based attacks, we observe that it is vulnerable under more intelligent attacks. We generalize the SOS architecture by introducing more flexibility in layering to the original architecture. We define two intelligent DDoS attack models and develop an analytical approach to study the impacts of the number of layers, number of neighbors per node and the node distribution per layer on the system performance under these two attack models. Our data clearly demonstrate that performance is indeed sensitive to the design features and the different design features interact with each other to impact overall system performance.

1. Introduction

Current level of sophistication in system resilience to Distributed Denial of Services (DDoS) or other forms of attacks is far from definite. Tremendous amount of research is being done in order to improve the system security under DDoS attacks. Communication reliability over the Internet is critical in emergency, medical, and other related services. Apart from providing a high degree of path availability for communication, such systems need to be resilient to attacks from malicious users within and outside of the system that aim to disrupt communication. Also, attacks on special nodes or hot spots in such systems can have catastrophic effects.

A Secure Overlay Services (SOS) architecture has been proposed in [1] in the framework of a set of clients communicating with a target during critical situations. The SOS architecture provides a high degree of path availability in the presence of random DDoS attacks. The design rationale is to ensure that in the presence of DDoS attacks, the target is not overloaded; the probability of all available paths between clients and the target being compromised is very small; and the attack traffic is dropped. In order to achieve these objectives, 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. The proposed SOS architecture has a variety of very nice and novel features. It is simple and easily deployable. In fact with only a very few set of nodes across the 3 layers, the SOS architecture provides good performance in terms of providing path availability between clients and the target even without system recovery under on-going attacks. However on a critical note, the following questions are naturally raised while analyzing the system:

- The system can be targeted by *intelligent* attackers. An intelligent attacker can have the ability to break into nodes in order to disclose their neighbors and may also be aware of identities of some nodes in the overlay prior to an attack. By *intelligent DDoS* attacks, we mean the attacker can launch a large amount of congestion-based DDoS attacks with a certain level of intelligence, such as obtaining system knowledge prior to launching DDoS attacks. An interesting question is, how is the SOS system performance (*in terms of path availability between clients and the target*) impacted under such intelligent DDoS attacks?
- The SOS architecture comprises of three key design features; number of layers, number of neighbors per node and the node distribution per layer. 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? What is the impact of node distribution per layer on the sys-

* Dong Xuan, Sriram Chellappan, and Xun Wang are with The Department of Computer Information and Science, The Ohio-state University, Columbus, OH 43210. E-mail: {xuan,chellapp,wangxu}@cis.ohio-state.edu. Shengquan Wang is with the Department of Computer Science, Texas A&M University, College Station, TX 77843. E-mail: swang@cs.tamu.edu.

tem performance? More interestingly, how do these design features combine with each other to impact system performance under different intensities of intelligent DDoS attacks?

In this paper, we aim to address the above issues. Specifically, (1) We generalize the SOS architecture such that the design features are flexible and contingent on expected attacks. (2) We define two intelligent DDoS attack models and develop an analytical approach to analyze the generalized SOS architecture under these attack models. The approach is general and can be applied to analyze other systems. (3) We analyze the generalized SOS architecture in detail under intelligent DDoS attacks towards understanding the sensitivity of system performance to each design feature. We observe that the number of layers and the number of neighbors per node have opposite effects on the resilience to break-in and congestion attacks. More layers and less neighbors per node improve 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 the number of neighbors per node. We also observe that the system performance is sensitive to the node distribution per layer, particularly when the number of neighbors per node is large.

2. The SOS architectures

In this section, we provide a brief description to the overall SOS system [1] from the point of view of the basic architecture and the attack scenarios analyzed there.

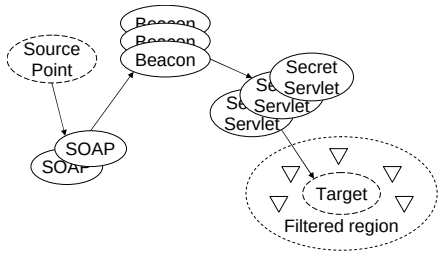


Figure 1. The original SOS architecture.

In the SOS architecture shown in Fig. 1, communication between clients and a target is through 3 intermediate layers. These layers are SOAP (Secure Overlay Access Point), Beacons and Secret Servlets. A client that wishes to communicate with a target first contacts a node in the SOAP layer. The node in the SOAP layer contacts a beacon, which then contacts a secret servlet, which routes the data through a filter towards the target. A set of filters acts as a firewall surrounding the target. In this architecture each source point is aware of nodes in the SOAP layer, which are aware of the Beacons, which know the Secret Servlets, which in turn know the identities of the filters. Nodes in each layer (and the filters) ensure that they route packets to the next layer after verification that the packet indeed arrived from a legitimate node in a lower layer. The underlying routing protocol used, is Chord [2] for more anonymity. The performance metric is the probability that a client can communicate successfully with the target by finding a path to it. For analysis purposes the attack model is *random congestion-based* DDoS attacks. Although a congested node does not allow attack traffic to pass through because of validation, it nevertheless becomes non functional due to DDoS attacks compromising path availability.

We wish to refer back to the questions about the SOS architecture raised in Section 1. The architecture although performs well for random congestion based attacks, will be fragile in the presence of intelligent attacks like break-in attacks as we show later. Under break-in attacks, the attacker can easily find the location of nodes towards the target. We believe that fixing the number of layers as 3 is not always the optimal choice. We aver that for given system resources, an increase in the number of layers will enable pure congestion based attacks to be more successful. In fact fixing the number of layers as 1 is the best choice to defend against such attacks. We believe 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.

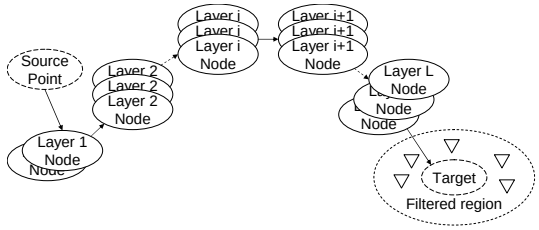


Figure 2. The generalized SOS architecture.

Following the above discussions, we generalize the original SOS architecture. In simple terms, our generalized architecture extends from the original SOS architecture and consists of multiple layers of nodes as shown in Fig. 2. The number of layers is denoted by L. The layering features are given below.

- The first layer and the last layer provide similar functionality as the SOAP layer and the Secret Servlet layer respectively in the original SOS architecture.
- The intermediate layers perform the functionality of the beacons in the SOS architecture. The difference is

that in our generalized architecture, there can be multiple layers. Similar to the SOS architecture, nodes in Layer $i+1$ will forward traffic that arrive only from a node at Layer i.

Having described our generalized architecture from the layering perspective, we formally introduce two other design features; the number of nodes in Layer i denoted by n_i and the number of neighbors a node in Layer $i-1$ has in Layer i (referred to as the mapping degree), denoted as m_i.

The novelty of our generalized architecture is its flexibility. Here, L, n_i and m_i are designed depending on the system resources and attacks. Our architecture being flexible can be designed easily considering other factors such as delay performance, guaranteed delivery for special clients etc.

3. Analysis of the Generalized SOS architecture

In the following we conduct an extensive analysis to our generalized SOS architecture on two attacks models: one-burst attack model and the successive attack model. In both the attack models, the attacker conducts the attack in two phases, (1) break-in attack phase and (2) congestion attack phase. The break-in attack phase discloses some nodes while the congestion attack phase congests the nodes based on the information about the disclosed nodes by the break-in attacks. The only difference between the one-burst and the successive attack model is that in the former, the break-in attack phase is conducted in one round, while in the latter it is conducted in successive rounds.

The system we study consists of a total of N overlay nodes, of which n nodes are in the SOS system (denoted as SOS nodes). In our attack model, the attacker has resources to launch break-in attacks on N_T nodes and congest N_C nodes. The attacker may have some prior knowledge about the identities of the SOS nodes before the attack. With a probability P_B, the attacker can successfully break into a node.

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. In this paper, we do not consider the dynamics of system repair to attacks, which is our future work.

3.1. Under one-burst attack without any prior knowledge about the SOS nodes

3.1.1. Attack model 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, which the attacker knows and still wishes to proceed with the attack. Here we assume the attacker has no prior knowledge about the SOS nodes.

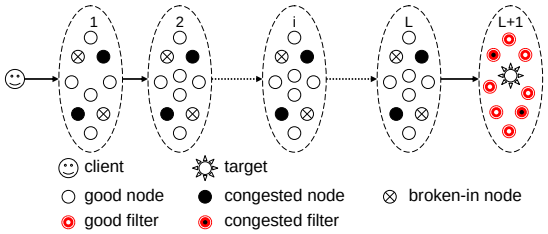


Figure 3. A Snapshot of the generalized SOS architecture under the intelligent DDoS attacks.

Before proceeding with the analysis, we emphasize here that the system/attack model we are analyzing is different from the one in [1] although the performance metric (P_S) is the same; (1) In our case, the break-in attacks will disclose some nodes, and the congestion attacks will focus on such nodes to attack and are not just performed in a totally random fashion. (2) Even before the start of an attack, the attacker has some prior knowledge about SOS nodes. Although in the analysis of the one-burst attack we discount this situation, it is not the case in the successive attack model we analyze which introduces additional complexity. (3) 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 once such a node is broken-into, nodes in several other layers will be disclosed and so we do not make this assumption. We observe that the factors mentioned above make our analysis harder, but is more realistic and the resulting architecture from this analysis is naturally more robust.

The key defining feature of our analysis is in determining the set [1] 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 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 SOS nodes

1 We use the term *set* and *number of nodes in a set* interchangeably.

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.

Recall that P_S is the probability that a user can successfully communicate with the target. 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* overlay 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. 3. 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:

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

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 SOS nodes. In our model, once a node is broken into, it is compromised and the attacker will not 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} 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,

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

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.

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

and then z_i is given by,

$$z_i = \sum_{j=1}^{n_i} (1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}(1 - \frac{h_i}{n_i})) \quad (4)$$

$$= n_i(1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}(1 - \frac{h_i}{n_i})). \quad (5)$$

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

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

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 unsuccessfully. This set is denoted by d_i^A and is given by,

$$d_i^A = \sum_{j=1}^{h_i - b_i} (1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}) = (h_i - b_i)(1 - (1 - \frac{m_i}{n_i})^{b_{i-1}}). \quad (7)$$

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

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 [2]. Therefore, the total number of congested overlay nodes in Layer i is,

$$c_i = \begin{cases} d_i^N + d_i^A + (N_C - N_D)* \\ \quad (\frac{n_i - b_i^A - d_i^N - d_i^A}{N - N_B - (N_D - d_{L+1}^N - d_{L+1}^A)}), & i = 1, \ldots, L, \\ d_i^N, & i = L+1. \end{cases} \tag{8}$$

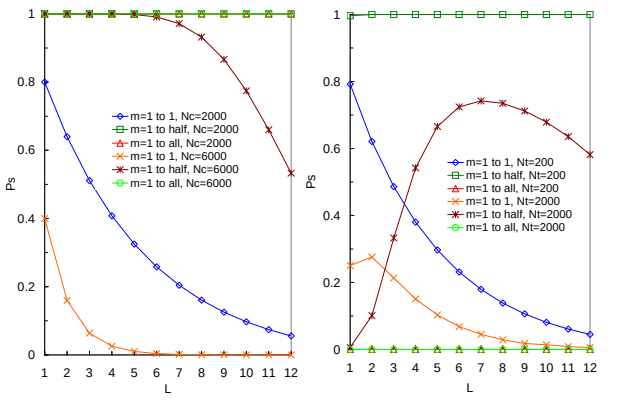
- $N_C < N_D$: The attacker can randomly congest the subset of N_C candidates among N_D disclosed nodes. In this case,

$$c_i = \frac{N_C}{N_D}(d_i^N + d_i^A), \tag{9}$$

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

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

3.1.2. Numerical Results and Discussion Fig. 4 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 attack model. The mapping degrees used here are: one to one mapping which means each SOS node has only one neighbor in the next layer; one to half mapping which means each node has half of all the nodes in the next layer as its neighbors; and one to all mapping which means each node has all the nodes in next layer as its neighbors. Other system and attack configuration parameters are: $N = 10000$, $n = 100$, $P_B = 0.5$, the SOS nodes are evenly distributed among layers. The number of filters is set as 10. In Fig. 4 (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. 4 (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;



(a) (b)

Figure 4. Sensitivity of P_S to L and m_i under different attack intensities.

- Fig. 4 (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 original 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 Layer L, a higher 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. 4 (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. 4 (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.

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

3.2. Under a successive attack with the prior knowledge about the SOS nodes

3.2.1. Attack model Our successive attack model extends from the one-burst attack model in two ways; (1) the attacker has some prior knowledge about the first layer SOS nodes. Let P_E represent the percentage of nodes at the 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 SOS nodes are disclosed in a round by round fashion thus accentuating the effect of attack. However in reality, R cannot be too large as that would allow the system enough time to detect and recover from an on-going attack before the attack is completed.

The strategy of the successive attack is shown in Algorithm 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 all of them. 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 SOS 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 enough.

3.2.2. 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 three possible overlap scenarios; (1) a disclosed node could have been already broken-into, (2) the same node being disclosed by multiple lower layer nodes and (3) a disclosed node could have been unsuccessfully broken-into. The complexity in overlap is accentuated here due to the nature of the successive attack model. 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.

Algorithm 1 Pseudocode of the successive attack strategy.

Parameters: System parameters : N, n, L, P_B and attack parameters : N_T, N_C, R, X_1, β, α.

Phase 1 of the attack strategy:

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;
5: update $\beta = \beta - \alpha$;
6: **end if**
7: **if** $X_j < \beta \leq \alpha$ **then**
8: 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;
9: break;
10: **end if**
11: **if** $\alpha \leq X_j < \beta$ **then**
12: launch break-in attack on all X_j nodes and calculate the set X_{j+1} disclosed nodes;
13: update $\beta = \beta - X_j$;
14: **end if**
15: **if** $X_j \geq \beta$ **then**
16: launch break-in attack on β nodes among X_j nodes and calculate the set X_{j+1} disclosed nodes;
17: break;
18: **end if**
19: **end for**
20: Calculate N_D;

Phase 2 of the attack strategy:

1: **if** $N_C \geq N_D$ **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**

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 a break-in attempt 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 successfully break into some of them. 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.

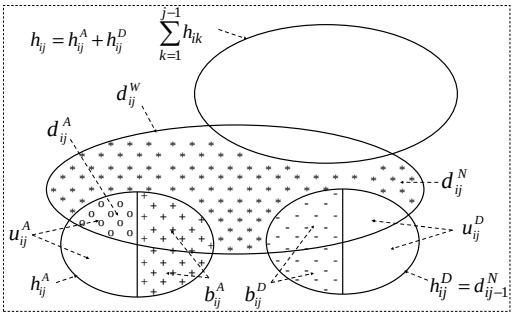


Figure 5. Node demarcation in our successive attack at the end of Round j.

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. 5 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 (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 $(\alpha - X_j)$ to break into more nodes than those disclosed already prior to that round. The attacker will expend these resources randomly.

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,

$$h^D_{i,j} = d_{i,j-1}, \tag{10}$$

$$h^A_{i,j} = \frac{n_i - d_{i,j-1} - \sum_{k=1}^{j-1} h_{i,k}}{N - X_j - \sum_{q=1}^{L} \sum_{k=1}^{j-1} h_{q,k}} (\alpha - X_j), \tag{11}$$

$$h_{i,j} = h^A_{i,j} + h^D_{i,j}, \tag{12}$$

$$b^D_{i,j} = P_B * h^D_{i,j}, \tag{13}$$

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

$$u^D_{i,j} = (1 - P_B) * h^D_{i,j}, \tag{15}$$

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

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

In (10), 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 $b^D_{i,j}$ and,

$$b_{i,j} = P_B * d_{i,j-1} + P_B\left(\frac{n_i - d_{i,j-1} - \sum_{k=1}^{j-1} h_{i,k}}{N - X_j - \sum_{q=1} L \sum_{k=1}^{j-1} h_{q,k}}\right) (\alpha - X_j), \tag{17}$$

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

The next part is to compute the set of nodes $d^N_{i,j}$ and $d^A_{i,j}$. As discussed above, we have to extract the set $d^N_{i,j}$ from $d^W_{i,j}$. Similar to the discussion in the one-burst attack case and from (5), (6) and (7), we calculate $d^N_{i,j}$ and $d^A_{i,j}$. We first calculate the set of nodes that have been either disclosed or attacked. For $b_{i-1,j} > 0$ and $i = 2, 3, \ldots, L$,

$$z_{i,j} = n_i\left(1 - \left(1 - \frac{m_i}{n_i}\right)^{b_{i-1,j}} \left(1 - \frac{\sum_{k=1}^{j} h_{i,k}}{n_i}\right)\right) \tag{18}$$

$$d^N_{i,j} = z_{i,j} - \sum_{k=1}^{j} h_{i,k}. \tag{19}$$

Note that in our attack model, the attacker will not try to break into a node twice. Hence, to calculate $d^N_{i,j}$, from $z_{i,j}$, we subtract the nodes on which a break-in attempt has been made. We then have,

$$d^A_{i,j} = (h^A_{i,j} - b^A_{i,j})\left(1 - \left(1 - \frac{m_i}{n_i}\right)^{b_{i-1,j}}\right), \tag{20}$$

for $b_{i-1,j} > 0$ and $i = 2, 3, \ldots, L$.

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: (1) $X_j < \beta \leq \alpha$, (2) $\alpha \leq X_j < \beta$ and (3) $\beta \leq X_j$. For case (1), all the formulas we derived for the general case can be directly applied, except that α has to be replaced by β. For case (2), all the formulas in the general case can be applied except that $h^A_{i,j} = 0$. For case (3), we have $h^A_{i,j}$ = 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 exhaustion 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,

$$f_{i,j} = d_{i,j-1} - (\frac{d_{i,j-1}}{X_j})\beta, \quad (21)$$

$$h^A_{i,j} = 0, \quad (22)$$

$$h^D_{i,j} = d_{i,j-1} - f_{i,j}, \quad (23)$$

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

$$d^N_{i,j} = \begin{cases} 0, & i = 1, \\ n_i(1-(1-\frac{m_i}{n_i})^{b_{i-1,j}} \\ \quad (1-\frac{\sum_{k=1}^{j} h_{i,k}+\sum_{k=1}^{j} f_{i,k}}{n_i})) \\ \quad -\sum_{k=1}^{j} h_{i,k} - \sum_{k=1}^{j} f_{i,k}, & i = 2, 3, \ldots, L+1, \end{cases} \quad (24)$$

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

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,

$$N_D = \sum_{i=1}^{L}\sum_{k=1}^{J} u^D_{i,k} + \sum_{k=1}^{J} d^N_{L+1,k} + \sum_{i=2}^{L} d^N_{i,J} + \sum_{i=1}^{L} f_{i,J} + \sum_{i=1}^{L}\sum_{k=1}^{J} d^A_{i,k}. \quad (25)$$

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,

$$c_i = \begin{cases} \sum_{k=1}^{J} u^D_{i,k} + d^N_{i,J} + \sum_{k=1}^{J} d^A_{i,k} \\ \quad + F_{i,J} + (N_C - N_D)(n_i \\ \quad - \sum_{k=1}^{J} b_{i,k} - \sum_{k=1}^{J} u^D_{i,k} - d^N_{i,J} \\ \quad - \sum_{k=1}^{J} d^A_{i,k} - F_{i,J})/(N \\ \quad - N_B - (N_D - \sum_{k=1}^{j} d^N_{L+1,k})), & i = 1, 2, \ldots, L, \\ \sum_{k=1}^{j} d^N_{L+1,k}, & i = L+1. \end{cases} \quad (26)$$

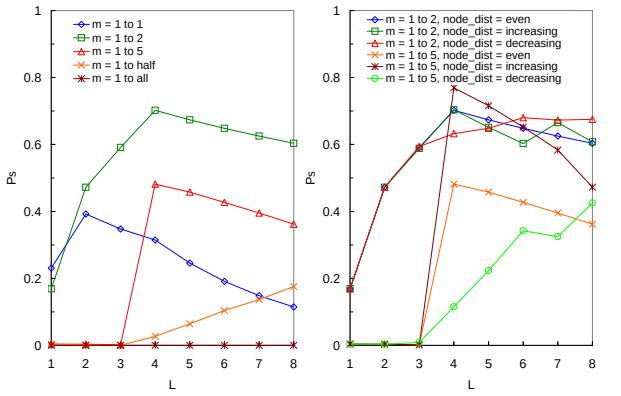
If $N_C < N_D$, we have,

$$c_i = \begin{cases} \frac{N_C}{N_D} * (\sum_{k=1}^{J} u^D_{i,k} + d^N_{i,J} \\ \quad + F_{i,J} + \sum_{k=1}^{J} d^A_{i,k}), & i = 1, 2, \ldots, L, \\ \frac{N_C}{N_D}(\sum_{k=1}^{J} d^N_{L+1,k}), & i = L+1. \end{cases} \quad (27)$$

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 (1) to compute P_S.

Note that prior knowledge about identities of the first layer SOS 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 $b_{i,j}, d^N_{i,j}, c_i$ etc. can be calculated by application of equations (10) to (27). 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 $b_{i,j}, d^N_{i,j}, c_i$ etc. will be simplified to the corresponding ones derived in the previous sub-section.

3.2.3. Numerical Results and Discussion 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 SOS nodes are evenly distributed among the layers. We introduce two new mapping degrees here, namely one to two mapping, meaning each SOS 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)

Figure 6. Sensitivity of P_S to L, m_i and node distribution.

Fig. 6 (a) shows the impact of L on P_S under different mapping degrees. Similar to Fig. 4 (a)(b), P_S is sensitive to L and the mapping degree even when $N_T > 0$ and $R > 1$. Among the current configurations, the one with $L = 4$ and mapping degree one to two provides the best overall performance.

Fig. 6 (b) gives us an insight on the impact of node distribution on P_S when L and the mapping degree changes. Other parameters remaining unchanged, here we show sensitivity of performance to three different node distributions per layer. The first is even node distribution where the number of nodes in each layer is the same ($\frac{N_T}{L}$). The second is increasing node distribution, where the number of nodes in the first layer are fixed ($\frac{N_T}{L}$). 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 ($\frac{N_T}{L}$) 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.

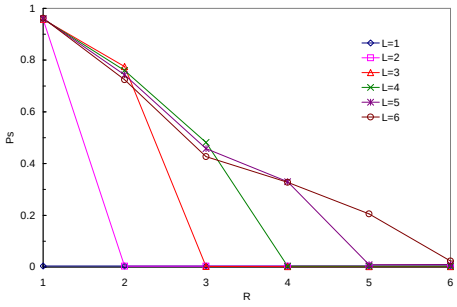


Figure 7. Sensitivity of P_S to R under different L.

Fig. 7 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.

In Fig. 8 we show how P_S changes with N_T as the other system side parameters change. Fig. 8 (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 SOS node distribution. Fig. 8 (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.

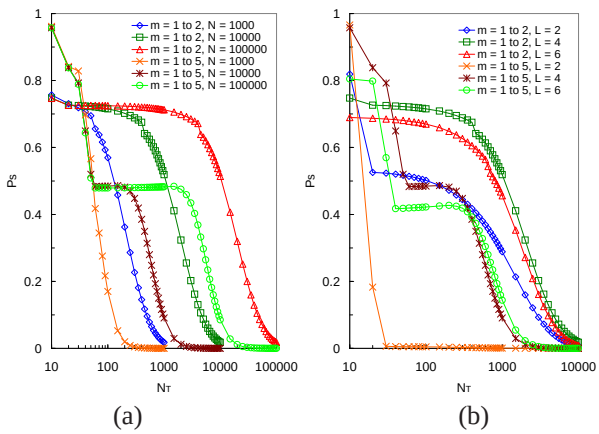


Figure 8. Sensitivity of P_S to N_T under different L, m_i and N.

- From Fig. 8, there is a 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 SOS 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 SOS node, and P_S does increase.

Due to the space limitations, we do not report our analysis on the sensitivity of P_S to N_C. Interested readers can refer [3]. However, we summarize all our findings as follows. The attack strategies and intensities 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. We also demonstrated why increasing node distributions perform better than other node distributions. Thus, if the system is designed carefully keeping potential attack scenarios in mind, more resilient architectures can be designed.

4. Related Work

The main purpose of this work is for analyzing system resilience against Distributed DoS attacks. The survey in [1] is exhaustive and interested readers can refer to that paper.

In the following, we would like to focus on work in overlay and anonymity systems.

Overlay networks have been widely used for multicasting[4], routing [5] and file sharing [6] etc. However, less work has been reported on the use of overlay solutions to enhance security of communication systems. Three of them are [1], [7] and [8]. Mayday [7] is a generalized SOS architecture that separates the overlay routing and lightweight packet filtering and provides a more powerful set of choices for each layer. However, it does not address the problems of layering and mapping degree issues, which our paper focuses on. An overlay solution to track DDoS floods is proposed in [8].

The goal of SOS and our generalized SOS is to ensure that, with high probability any client can find a path to the target under DDoS attacks. The attacker needs to find out the location of target to congest it or disrupt all possible intermediate paths. Hence the target protection is very important. The key technology used by SOS is providing receiver (target) location anonymity by allowing sources to contact SOAP layer nodes only. The attacker has no idea about successive paths taken by messages or the location of the target. Besides using the anonymity approach, SOS also tries to ensure a path from clients to the target by putting multiple connections between nodes in successive layers. A lot of anonymity systems, particularly ones aiming to achieve receiver anonymity, depend on one or more third party nodes to generate an anonymous path [9, 10], which is not good for SOS. SOS cannot rely on a centralized node to achieve receiver anonymity, since the centralized node can itself be the target of a DDoS attack. SOS uses multiple layering technology to achieve receiver location anonymity in a distributed fashion. Our generalized architecture further extends these technologies.

5. Final Remarks

Our contributions in this paper are (1) systematically studying the existing SOS architecture from the perspective of its basic design features, (2) proposing a generalized SOS architecture by introducing flexibility to the design features, (3) defining two intelligent DDoS attack models and developing an analytical approach towards analyzing the generalized SOS architecture under these two attack models. We make interesting observations on the sensitivity of system performance to the design features. There are some open issues related to this study, mentioned below:

More sophisticated attack models and Dynamic repair: We can further refine our attack model by introducing more intelligence. For instance, during the break-in phase of the attack, the attacker can also find previous layer nodes of an attacked node by monitoring the on-going traffic and can also build up a layering model of the architecture causing an increased damage to the system. Also, we do not consider system repairs here. It is very hard, if not impossible, to mathematically analyze such sophisticated attacks with dynamic repair mechanisms. Also, attacks on the underlying network are possible, although hard to analyze especially when the attacker is intelligent. We are planning to study the system behavior under such sophisticated attacks and system dynamics using extensive simulations.

Timely delivery: Timely delivery is an open issue of SOS [1]. In our generalized architecture, an increase in the number of layers increases resilience to break-in attacks and also the latency of communication. An increase in the mapping degree decreases resilience to break-in attacks. However the latency here may be minimized due to more routing choices. Thus from the perspective of timely delivery, there are interesting trade-offs in layering and the mapping degree.

References

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

[2] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of ACM SIGCOMM*, San Diego, CA, August 2001.

[3] D. Xuan, S. Chellappan, X. Wang, and S. Wang, "Analysis of the generalized secure overlay services architecture," Technical Report, The Department of Computer and Information Science, The Ohio State University, August 2003.

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

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

[6] 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.

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

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

[9] 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.

[10] 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.