Protecting Resources Against Volumetric and Non-volumetric Network Attacks

Rajorshi Biswas[1] and Jie Wu[2]
Information Sciences and Technology, Penn State Berks, Reading, PA, USA[1]
Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA[2]

***Abstract*—Cyber attacks are growing with the increase in internet usage. In a volumetric attack, the target resource is taken down with a huge amount of traffic. Distributed denial-of-service and link flooding attacks are examples of these types of attacks. In a non-volumetric attack, the attackers try to steal or get illegal authorization of some resources in a network. This type of attack can be severe even with a small amount of traffic. Non-volumetric attacks can be stopped by applying a moving target defense approach at the nodes on the attack path. An attack path is a series of steps and the attacker needs to succeed in all of those steps to gain access to the resources. In this paper, we propose an architecture to defend against both types of attacks. We formulate a problem to minimize the damage caused by the volumetric attack by using a limited number of blockage at some routers. This problem is NP-hard and we provide a greedy solution and provide an approximation ratio of it. We formulate another optimization problem to minimize the damage while securing the resources by deploying the minimum number of moving target defense methods. We provide a dynamic programming based solution to this problem. We conduct an extensive simulation to support our proposed models.**

Index Terms*—*Traffic engineering, link flooding attack, software defined networking, minimize rules

I. Introduction

The growth of internet usage and services produces a lot of opportunities for network attackers. Based on the effectiveness of the volume of attack packets we divide the attacks into two categories: volumetric and non-volumetric attacks. In a volumetric attack, the amount of damage depends on the amount of attack volume. The packets are not harmful or contain malware but the amount creates congestion in the network and cause it to stop serving regular users. The distributed denial-of-service (DDoS) and link flooding attacks (LFA) are this kind of attack.

To defend against this attack, we use filters that consist of some rules to block certain traffic based on its source and destination address. A filter can be applied to the routers by the defender to reduce the unnecessary traffic reaching the resources. There is limited storage for filters and the owner ISP may charge money for assigning filters. Therefore, we consider a limited budget for the number of filters. A good filter assignment can block more traffic and wast least resources. For example, in Fig.1, if we are allowed to place only one filter, we may place it on router B. A filter on B will allow a maximum of 1 attack traffic to the resource R. If we place it on E then 3 attack traffic can reach R at most. Therefore, a good filter assignment is needed to minimize the amount of attack traffic reaching the resources to minimize damage.

This research was supported by NSF grants CNS 2128378, CNS 2107014, CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, and CNS 1651947.

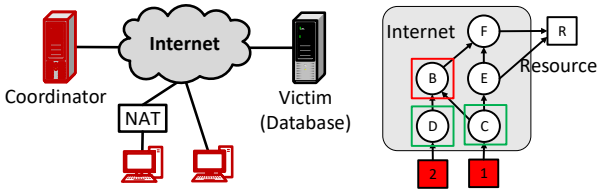


Fig. 1: An example of attack and defense mechanism.

In a non-volumetric attack, the amount of traffic is not proportional to the damage to the resources. The attacker intends to steal the content of the resource or gain special access permission to the resources. To achieve the final goal, the attacker needs to pass multiple steps. There can be one or multiple ways (series of steps) to reach the goal. For these types of attacks, the defender needs to cut all possible ways to reach the resources. Let us consider the figure in Fig. 1 and assume the nodes are the steps. The first step of the attackers is either passing through D or C. In reality, the D step can represent the guessing of the password of a server. Step B can be gaining root privilege of that server. When an attacker passes one more step, it gets closer to the goal and causing damage to the network or datacenter. To prevent the attacker, the defender needs to apply a moving target defend (MTD) approach to stop the attacker at some particular steps. Simultaneously, the attacker needs to be stopped as early as possible to minimize the damage. In Fig. 1, we need at least two MTDs to deploy to stop the attack. The best locations for the MTDs are at step D and C because they yield no damage to the network/datacenter. If we assign the MTDs at B and E it would stop the attacker from reaching the goal but we would lose D and C. Therefore, a good MTD assignment is necessary to minimize the damage while ensuring the protection of the resources.

In this paper, we study the defending mechanism of volumetric and non-volumetric attacks and model the problem in an optimization framework. We formulate two problems for these two types of attacks with different objectives and constraints. We consider the amount of damage to be proportional to the amount of traffic in volumetric attacks. The amount of damage in a non-volumetric attack is proportional to the number of steps. For simplicity, we are considering the amount of damage for each step to be the same. Therefore, the main contributions are the following:

1) We study an optimization problem for minimizing damage caused by the volumetric attack and provide an

approximation solution with a performance guarantee.
2) We formulate another problem for minimizing damage for non-volumetric attacks by ensuring the protection of the resources. A dynamic programming solution is provided for this problem.
3) An extensive simulation is conducted to evaluate the solutions.

The remainder of this paper is arranged as follows: Section II presents some related works. In Section III, we present the system, attacker, and cost models. Section IV and Section V present the formal definitions of the problems and our proposed solutions, respectively. Section VI presents some simulation results. Finally, Section VII concludes the paper.

II. Related Works

For volumetric attacks there exist many statistical methods, including correlation, co-variance, entropy, cross-correlation, and information gain to detect anomalous attack requests [1]. A rank correlation-based method and information theoretical approach are proposed in [2] and [3], respectively. An Artificial Neural Network based approach is proposed in [4]. In proposed in [5] authors propose a dynamic neural network that learns to activate the neuron based on input data using unsupervised learning. Other type of DDoS attack defense mechanism includes statistical methods to classify packets and block them [2].

There exists many research on another type of volumetric attack called link flooding attack. In [6], the authors propose SPIFFY that logically increases link capacity when it detects congestion. A router functionality based mechanism is proposed in [7], in which each router detects and preferentially drop packets that likely belong to an attacker. In ColDef [8] mechanism, the domains which are uncontaminated by attackers help to route the legitimate traffic. It also enables routers to detect low-rate attacks flows. In [9], authors proposed a link flooding attack mitigation system by using BGP rules in BGP routers. If a link congestion is detected, then the BGP router advertises its neighbors to avoid the congested link.

There also exists several works for mitigating non-volumetric attacks. In [10], authors present a path discovery method of cyber-attacks. The method uses DFS search to effectively generate attack graphs. Authors first generate the gaph from capability and location data. Then they reduce the graph by removing resources that are out of reach of the attacker. In [11] authors propose a model only with the path of the network nodes involved in the attack to be analyzed in detail. A network attack path detection model based on attack graph is also proposed. First, the formulate an attack graph and use it to describe the transfer relationship between nodes. They map the process of the attack from one host to the next host and discover the path to identify the attack intention. An MTD based defense mechanism is proposed in [12] for non-patchable vulnerabilities. They propose to change the attack surface of the IoT network to increase the attack effort. They develop two proactive defense mechanisms that reconfigure the SDN-based IoT network topology. In [13], authors propose a strategy to use a diverse set of security mechanisms, such that the impact from a vulnerability in any security mechanism is minimized. They introduce a game-theoretic graph coloring technique to get the optimal allocation of security mechanisms that minimizes the impact of security vulnerabilities to the power grid.

We discussed three types of existing systems: (1) statistical approaches that analyze packets or traffic properties to detect and block volumetric attack traffic (2) usage of machine learning to detect both volumetric and non-volumetric attack traffic and block, and (3) game theory based system that incorporate attack paths to defense non-volumetric attacks. None of these system consider FR or MTD as defense mechanism and utilize them perfectly. Earlier works on volumetric attacks based on FR use tree based topologies which is rate nowadays. Therefore it is important to develop a system that can work on any kind o topology.

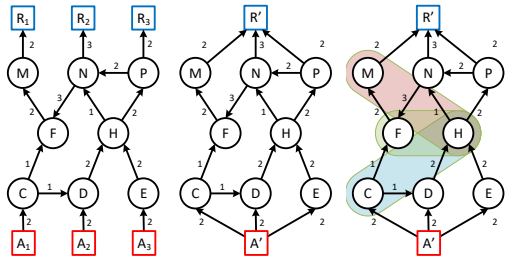
III. System Model

A. Network Model

Our network is composed of filter routers (FR), attackers, a defender, resources, and legacy routers (LR). The resources can be databases, files, and credential servers which are the most valuable resources that need high protection. The servers can be connected to any location of the network. A filter router is a special type of router which is capable of accepting filters and applying them to block some traffic [14]. A filter is a packet blocking rule based on source and destination IP address. The filter sent by the ISP is only applicable to the packets which are destined for the resources owned by that ISP. It is possible that an attacker spoofs the IP addresses of the ISP and sends the wrong filters to FRs so that legitimate traffic is blocked. This spoofed filter request can be detected using a simple handshaking protocol. The spoofing attacker is not capable of exchanging handshake messages with the spoofed IP address. The defender is responsible for deploying filters and the moving target defense (MTD) module to protect the resources owned by that ISP. An MTD module changes the configuration of some node or server periodically so that the attacker cannot succeed. The configuration changes while the attacker is in the progress of action so that it needs to start over. Therefore, when the MTD module is applied the attacker cannot succeed in that step of the attack. Each Filter or MTD module deployment incurs some cost to the ISP. Therefore, the ISP wants to deploy a limited number of filters and MTD modules while getting maximum protection.

B. Attack Model

We consider two types of attacks in our attack model: volumetric and non-volumetric attacks. In a volumetric attack, the attack traffic is not harmful but the amount of traffic harms the service. For example, DDoS attacks are volumetric because the packets fired by attackers are not harmful but the amount of traffic exhaust the capacity of the servers. The attackers are usually user devices with malicious programs that can generate traffic as commanded by the master. The



(a) Topolgoy G. (b) G'. (c) Min-cuts (G').

Fig. 2: An example for volumetric attack.

master can send different types of attack commands to the programs. This type of malicious program is called a bot, and a network of bots is called a botnet. In a non-volumetric attack, the amount of traffic is not important but the packets are harmful. The attackers are either program that may reside on compromised machines that can generate attack traffic destined for a target. The programs are usually controlled by a human attacker. Sometimes the human attacker can devise an attack based on vulnerabilities of a network. For example, a password guessing attack may produce a scanty amount of traffic but have the outcome of exposing the password database to the attackers. There are several types of non-volumetric attacks such as malware installation, administrator access acquiring and accessing private networks and data. These attacks are conducted in series for more optimistic goal. For example, an attacker trying to get data from a server that is connected to a private network. One simple way to do this is to gain access to the machine which is connected to both networks. This may include gaining root privilege of the intermediate machine which increases one more step to the goal. Then using that machine an attacker can gain access to the target machine.

C. Cost, Budget and Damage Model

The cost of defense and damages incurred by the attack are different for these two types of attack. In a volumetric attack, the cost is incurred by the filter assignment. The hosting internet service provider of FRs may charge money for applying filters. In reality, the cost for filters may vary for different ISPs but for simplicity, we assume a uniform cost of filters. We assume the network provider has a limited budget for the number of filters. We denote the budget of the service provider as K. Therefore, the service provider wants to minimize the damage incurred by the attack to the minimum.

As the traffic is benign (not intent to steal), the service provider can allow some of it pass through to the resource servers. The amount of damage is proportional to the amount of traffic received by the resources. This is because the wasted bandwidth could be used to serve its users. So, the cost can be defined as $C = \sum_{r \in R} B(r)$ Here, the set of resources is R. $B(n)$ denotes the amount of attack traffic incoming to node n.

IV. Volumetric Attack Problem Formulation

In this section, we formulate the problem of filter assignment so that traffic reaching the resources is minimal.

Problem I: *Find K number of nodes to apply filters so that the traffic reaching the resources is minimum.*

Let the topology be $G = (V, E)$ where V is the set of nodes and E is the set of links. Let the set of attacker and resources are A and R ($R \subset V$ and $A \subset V$). $B(n)$ denotes the amount of attack traffic of attack success through node n. Therefore, the problem can be expressed as the following:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{r \in R} B(r) \\ \text{subject to} \quad & \textstyle\sum_{n \in V} M(n) \leq K \end{aligned} \tag{1}$$

Here, $M(n)$ is 1 if a MTD is applied on n, otherwise 0.

A. Solution

This problem is NP-Hard and we provide a greedy solution based on the min-cut problem. We first create $G' = (V', E')$ from the original topology G. In G', we combine all of the attackers and create a super attacker A'. We also combine all the resources and create a super resource R'. Therefore, $V' = V \cup A' \cup R' - (A \cup R))$. Now, the problem can be viewed as a flow problem where we want to minimize the maximum flow by removing some nodes.

Next, we need to find all possible minimum cuts in the flow network. To find all possible min-cuts we adopt the Kanevsky [15] method. If the size of the cut set is equal or less than K, then any of the cut set is the best solution. If the size of the cut set is higher than K, then we use a greedy procedure to find the best K nodes. For each cut set, we calculate the maximum blockage for K nodes.

To find the maximum blockage for a cut set $S_c = n_1, n_2, ...$ for K nodes, we calculate the contribution to max flows of each nodes in S_c. We calculate the maximum incoming and outgoing flows MAX-FLOW(A, n_i) and MAX-FLOW(n_i, R) to and from node n_i. The contribution of node n_i in the max-flow is the minimum of the incoming and outgoing flows. We choose the node with the maximum contribution to max-flow first. Then, the capacities of each link carrying the flows are reduced. Similarly, we calculate the contributions of the rest of the nodes and pick the one with the highest contribution. This process continues until we pick up K nodes. The complete algorithm is shown in Alg. 1.

B. An Example

Let us consider the example in Fig. 2. Fig. 2(a) shows the original topology G of the network. R_1, R_2, and R_3 are the resources and they need to be protected as much as possible. A_1, A_2, and A_3 are the attackers and they launch a DoS attack. The links are directed since we only consider the incoming traffics of the resources. The number beside each link shows the capacity of the links. Other nodes in the topology are considered as FRs. We transform the original topology G to G' by combining the resources and the attacker. Fig. 2(b) shows the G'. In G', A_1, A_2, and A_3 are combined to A' and the

links are added with the corresponding capacities. Similarly, the resources R_1, R_2, and R_3 are combined to R'. Next, we find all of the minimum cut sets in G' using the Kanevsky [15] method. We do not show the details of this process to save space. There are three minimum cut sets $\{C, H\}$, $\{C, AD\}$, and $\{H, M\}$. The cut sets are shown in Fig. 2(c).

If the maximum allowable number of filter (K) is 2, then any of the cut sets is the optimal filter assignment. In this case, no attack traffic will reach the resources. Let us assume that $K = 1$. Now we need to find the maximum blockage for each cut sets for one blockage. For $\{C, H\}$, if we remove H from G', then the maximum flow going through C is 1. Therefore, the contribution of node C is 1. Similarly, if we remove C from G', the maximum flow going through H is 3. Therefore, the contribution of node H is 3. We select the best 1 node ($K = 1$), which is H. Therefore, the maximum blockage for $\{C, H\}$ is 3 for $K = 1$. Similarly, the maximum blockage of $\{C, D\}$ and $\{H, M\}$ cut sets is 3. Therefore, we pick $\{C, H\}$, and the filter needs to be installed at node H. The amount of traffic reaching the resources after installing the filter is 1.

Theorem 1. *The complexity of Alg. 1 is* $O(|S_c||V|(|V| + |E|f))$.

Proof. To calculate the complexity of Alg. 1, we need to calculate complexity of MAXBLOCKAGE(S_c, K). According to [15], Step 9 takes $O(\kappa(|V| + |E|))$, where κ is the connectevity of graph G. Step 12 and 13 take $O(|E|f)$. Here, f is the maximum attack traffic flow in the network. The loop at Step 10, takes $|V||E|f)$. Here, $|V|$ is the maximum number of nodes in a cut sets. Therefore, MAXBLOCKAGE(S_c, K) takes $O(\kappa(|V|+|E|)+|V||E|f)$. In the worst case, the connectivity can be $|V|$. The complexity of MAXBLOCKAGE(S_c, K) is $O(|V|^2 + |V||E| + |V||E|f)$ which is $O(|V|(|V| + |E|f))$. Therefore, the Alg. 1 takes $O(|S_c||V|(|V| + |E|f))$. □

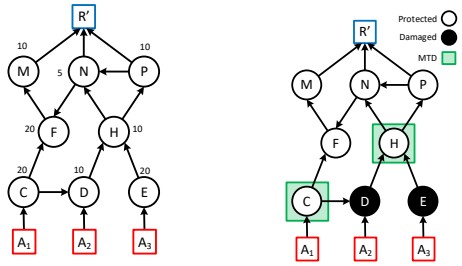
Theorem 2. *The approximation raio of Alg. 1 is* $(1 - 1/e)$.

Proof. To find the approximation ratio of Alg. 1, we need to find the approximation ratio of procedure MAXBLOCKAGE(S_c, K). The procedure picks the best node having maximum max flow by deleting the other cut nodes. When we remove the other cut nodes, all the traffic passes through that node. Then we reduce the capacity of the link passing traffic. This process is similar to the solution of the greedy maximum coverage problem. If we consider each node in the cut set as a set and each link on each distinct path with a unit amount of flow as an element of the set represented by the node, then the problem is to find K sets with maximum elements. The approximation ratio of the maximum coverage problem is $(1 - 1/e)$. Therefore, the approximation ratio of the MAXBLOCKAGE(S_c, K) is $(1 - 1/e)$. The rest of the parts of Alg. 1 search in all possible options. Therefore, the approximation ratio of Alg. 1 is $(1 - 1/e)$. □

Algorithm 1 Greedy Blocking Strategy

Input: The number of filters K and topology graph G.
Output: A set of nodes in G.

Procedure: BLOCK(K, G)
  $N \leftarrow$ number of nodes in G
  $S \leftarrow$ All minimum cut sets.
  for $S_c \in S$ **do**
    $M[S_c] \leftarrow$ MAXBLOCKAGE(S_c, K).
  return ARGMAX(M)
Procedure: MAXBLOCKAGE(S_c, K)
  $N \leftarrow$ number of nodes in G
  $S \leftarrow$ All minimum cut sets.
  for $n \in S_c$ **do**
    $G' \leftarrow$ Remove $S_c - n$ nodes from G.
    $f_i \leftarrow$ MAX-FLOW(A, n, G').
    $f_o \leftarrow$ MAX-FLOW(n, R, G').
    $C[n] \leftarrow$ MIN(f_i, f_o)
  $MaxC \leftarrow$ ARGMAX(C)
  $MaxBlockage \leftarrow MaxBlockage +$ MAX(C)
  Reduce capacity of link carrying flows through $MaxC$
  return $MaxBlockage$.



(a) G''. (b) MTD assignment in G''.

Fig. 3: An example for non-volumetric attack.

V. NON-VOLUMETRIC ATTACK PROBLEM FORMULATION

In this section, we formulate the problem of assigning the MTD methods to the machines so that no attack traffic can reach the resource servers.

Problem II: *Find K number of nodes to apply MTD so that the system is secured and the damage is the minimum.*

Let, the topology is $G = (V, E)$ where V is the set of nodes and E is the set of links. Let the set of attacker and resources are A and R. The set of nodes that can be chased by an attacker is D. Therefore, the problem can be expressed as the following optimization problem:

$$\begin{aligned} &\text{minimize} && |D| \\ &\text{subject to} && \textstyle\sum_{n \in V} M(n) \leq K, \\ & && \textstyle\sum_{r \in R} B(r) = 0 \end{aligned} \tag{2}$$

Here, $M(n)$ is 1 if a MTD is applied on n, otherwise 0.

A. Solution

We solve the problem using dynamic programming. To solve the problem, we define the following problem:

$P(n, k, t)$: Find and return the minimum damage in the subgraph (DAG) rooted by n for k number of MTD by yielding $(k = 1)$ or blocking all $(k = 0)$ attack traffic. The optimal damage, MTD assignments, covered nodes, and damaged nodes are stored in $D[n, k, t]$, $A[n, k, t]$, $C[n, k, t]$, and $L[n, k, t]$ to reuse in dynamic programming.

D[R']

k \ T	0	1
0	-	105
1	-	80
2	30	

D[M]

k \ T	0	1
0	-	105
1	75	80
2	30	50

D[F]

k \ T	0	1
0	-	95
1	75	70
2	30	40

D[N]

k \ T	0	1
0	-	75
1	50	55
2	20	35

D[P]

k \ T	0	1
0	-	70
1	50	50
2	20	30

D[H]

k \ T	0	1
0	-	60
1	50	30
2	20	20

D[D]

k \ T	0	1
0	-	30
1	20	10
2	0	20

D[E]

k \ T	0	1
0	-	20
1	0	-
2	0	-

D[C]

k \ T	0	1
0	-	20
1	0	-
2	0	-

A[R']

k \ T	0	1
0	-	∅
1	-	H
2	CH	

A[M]

k \ T	0	1
0	-	∅
1	F	H
2	CH	DE

A[F]

k \ T	0	1
0	-	∅
1	F	H
2	CH	DE

A[N]

k \ T	0	1
0	-	∅
1	H	C
2	DE	CE

A[P]

k \ T	0	1
0	-	∅
1	H	C
2	DE	CE

A[H]

k \ T	0	1
0	-	∅
1	H	C
2	D,E	CE

A[D]

k \ T	0	1
0	-	∅
1	D	C
2	C,D	C

A[E]

k \ T	0	1
0	-	∅
1	E	-
2	E	-

A[C]

k \ T	0	1
0	-	∅
1	C	-
2	C	-

Fig. 4: Values of A and D for all nodes and k.

There are two options to assign MTD. $P(n,k,t)$ is the minimum of the following two options:

Option I: The minimum total damage, if we assign 1 MTD to node n, divide the rest of the $k-1$ MTDs into $k_1, k_2, ..., k_\Delta$ parts, assign the parts to the subgraphs $c_1(n), c_2(n), ..., c_\Delta(n)$, and either block or partially block attack traffic, respectively. Therefore, the number of damaged nodes will be the union of the minimum damages for blocked/unblocked attack traffics in $c_1(n), c_2(n), ..., c_\Delta(n)$. In this case, the MTD assigned to n blocks all of the attack traffic. Therefore, this option is applicable to $k = 0$ only. As there are overlap among the subgraphs, and if a node is covered by MTD assignment by a subgraph and not coverved by other subgraphs, then the node will be covered in the current DAG.

$$\begin{aligned} L[n,k,0] &= \min_{\forall_{k_\delta},t} |\bigcup_{\delta=1}^{\Delta} P(c_\delta(n),k_\delta,t) - C[c_\delta(n),k,t]| \\ P(n,k,0) &= \sum_{n' \in L[n,k,t0]} V(n') \end{aligned} \tag{3}$$

Here, $\forall_\delta \sum_{\delta=1}^{\Delta} k_\delta = K-1$.

Option II: The minimum total damage, if we divide the number of MTDs into $k_1, k_2, ..., k_\Delta$ parts, assign them to the subtrees $c_1(n), c_2(n), ..., c_\Delta$(n), and either block or partially block attack traffic, respectively. Therefore, the damage for this option will be:

$$\begin{aligned} L[n,k,t] &= \min_{\forall_{k_\delta},t} |\bigcup_{\delta=1}^{\Delta} P(c_\delta(n),k_\delta,t) - C[c_\delta(n),k,t]| \\ P(n,k,t) &= \sum_{n' \in L[n,k,t]} V(n') + tV(n) \end{aligned} \tag{4}$$

Here, $\sum_{\delta=1}^{\Delta} k_\delta = K$. We take the minimum quantity from the above two options. Let us consider an N node DAG with the maximum node degree Δ. We define D as an $N \times K \times 2$ array that contains optimal damage for every node, budget, and blocked or unblocked attack traffic. For example, $D[n,k,0]$ contains optimal damage in the sub DAG rooted by n of budget k by blocking all attack traffic.

We define A as an $N \times K \times 2$ array which contains the MTD assignments in subgraph rooted by every node. We also define C and L as $N \times K \times 2$ arrays that contain the protected and damaged nodes in subgraph rooted by every node, respectively.

The complete algorithm is shown in Alg. 2.

Algorithm 2 DP MTD assignment strategy for Problem 2

Input: Budget on MTD K, and topology graph G''.
Output: A set of nodes in G''.

Procedure: ASSIGN-MTD-DP(K, G)
  $N \leftarrow$ number of nodes in G''
  $S \leftarrow$ topological order of nodes in G''
  for every entry node n **do**
    for $k = 0$ to K **do**
      Initialize $D[n,k]$, $T[n,k]$, and $A[n,k]$
  for every $n \in S$ **do**
    for $k = 0$ to K **do**
      $OP_1 \leftarrow D[n,k]$ using equation 3
      $OP_2 \leftarrow D[n,k]$ using equation 4
      $D[n,k] \leftarrow$ MIN(OP_1, OP_2)
      $A[n,k] \leftarrow$ ARGMIN($A_O P_1 \cup A_O P_2$)
  return $A[R', K]$

B. An Example

Let us consider the attack path graph in Fig. 3(a). We compute the values of $D[n,k,t]$, and $A[n,k,t]$ for every node, $k = 0, 1,$ and 2 and $t = 0$ and 1. We are not showing details calculation of $C[n,k,t]$ and $L[n,k,t]$ because of the limited space. The values of $V(n)$ are given in the DAG in Fig. 3(a). We first find a topological order for the calculation. One of the topological orders of the DAG in Fig. 3(a) is $\{C, E, D, H, P, N, F, M, R'\}$.

Calculation for Nodes C and E: The nodes C and E do not have any children. Therefore, the calculations of D and A are straightforward. For example, $D[C,0,0] = -$. Here "$-$" indicates an invalid option. This is because without any MTD, it is not possible to block all attacks C. If we assign a MTD to node C, then we are saving one node, thus $D[C,1,0] = 0$. Similarly, $D[C,2,0]$ is 0. If we want to yield attack traffic without any MTDs, then the node C gets damaged. Therefore, $D[C,0,1] = 20$. If we assign any MTD to C, then we cannot yield any attack to its ancestors. Therefore, $D[C,1,1]$ and $D[C,2,1]$ is $-$.

Calculation for Node H Using 2 MTDs: For $k = 2$, we have two options for assigning the MTDs.

Option I: 1 MTD for node H and one MTDs for its subgraphs. We can assign 1 MTD to the subgraphs in two ways: $(k_1 = 1, k_2 = 0)$ or $(k_1 = 0, k_2 = 1)$. For the first way, $(k_1 = 1, k_2 = 0)$, we assign the subgraph rooted by D and E to one MTD and zero MTDs. We can also consider the choices for $t = 0$ and $t = 1$. Therefore, we have eight choices for two ways (each way can be assigned in four ways).

Choice(1): ($k_1 = 1, k_2 = 0, t_1 = 0, t_2 = 0$) The total lost nodes for this choice is $(L[D,1,0] \cup L[E,0,0]) \setminus (C[D,1,0] \cup L[E,0,0]) = (\{C\} \cup -) \setminus (\{D\} \cup -)$. Therefore, this choice is invalid. **Choice(2):** ($k_1 = 1, k_2 = 0, t_1 = 0, t_2 = 1$) The total lost nodes for this choice is $(L[D,1,0] \cup L[E,0,1]) \setminus (C[D,1,0] \cup L[E,0,1]) = (\{C\} \cup \{E\}) \setminus (\{D\} \cup \emptyset) = \{C, E\}$. Therefore, the damage for this choice is 40 and because of $t_2 = 1$, this choice will yield attack traffic. **Choice(3):** ($k_1 = 1, k_2 = 0, t_1 = 1, t_2 = 0$) The total lost nodes for this choice is $(L[D,1,1] \cup L[E,0,0]) \setminus (C[D,1,1] \cup L[E,0,0]) = (\{D\} \cup -) \setminus (\{C\} \cup -)$. Therefore, this choice is also invalid. **Choice(4):** ($k_1 = 1, k_2 = 0, t_1 = 1, t_2 = 1$) The total lost nodes for this choice is $(L[D,1,1] \cup L[E,0,1]) \setminus (C[D,1,1] \cup L[E,0,1]) = (\{D\} \cup \{E\}) \setminus (\{C\} \cup \emptyset) = \{D, E\}$. Therefore, the damage for this choice is 30 and because of $t_1 = 1$ and $t_2 = 1$, this choice will yield attack traffic.

Similarly, we can calculate the damages other choices: **Choice(5):** ($k_1 = 0, k_2 = 1, t_1 = 0, t_2 = 0$), **Choice(6):** ($k_1 = 0, k_2 = 1, t_1 = 0, t_2 = 1$), **Choice(7):** ($k_1 = 0, k_2 = 1, t_1 = 1, t_2 = 0$), and **Choice(8):** ($k_1 = 0, k_2 = 1, t_1 = 1, t_2 = 1$).

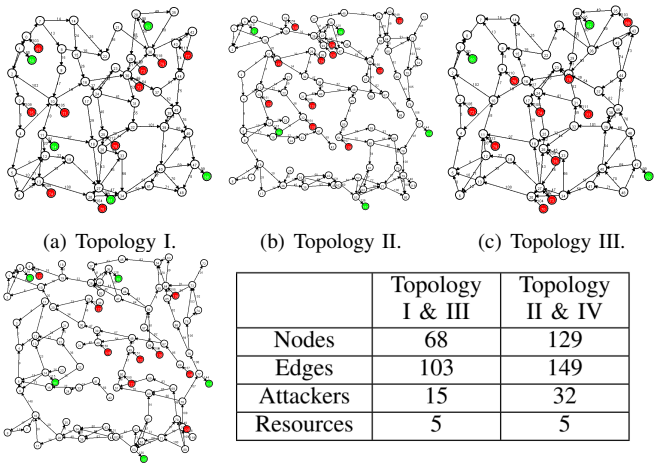
Option II: We assign 0 MTDs to node H and rest of the MTDs to its subgraphs. We have three ways to assign MTDs to its subgraphs: ($k_1 = 2, k_2 = 0$), ($k_1 = 1, k_2 = 1$), or ($k_1 = 0, k_2 = 2$). For each way we need to consider four choices. Because of the limited space we are only showing the choice that produce minimum damage. The choice ($k_1 = 1, k_2 = 1, t_1 = 0, t_2 = 0$) will produce the minimum damage by yielding no attack traffic. The total lost nodes for this choice is $(L[D,1,0] \cup L[E,1,0]) \setminus (C[D,1,0] \cup L[E,1,0]) = (\{C\} \cup \emptyset) \setminus (\{D\} \cup \{E\}) = \{C\}$. Therefore, the damage for this choice is 20 and because of $t_1 = 0$ and $t_2 = 0$, this choice will yield no attack traffic. Therefore, $A[H,2,0] = 20$ and $A[H,2,0] = (A[D,1,0] \cup A[E,1,0]) = \{D, E\}$.

The choice ($k_1 = 1, k_2 = 1, t_1 = 1, t_2 = 0$) will produce the minimum damage by yielding attack traffic. The total lost nodes for this choice is $(L[D,1,1] \cup L[E,1,0]) \setminus (C[D,1,1] \cup L[E,1,0]) = (\{D\} \cup \emptyset) \setminus (\{C\} \cup \{E\}) = \{D\}$. Therefore, the damage for this choice is $10 + 10 = 20$ as the attack traffic damages node H ($V(H) = 10$) . Therefore, $A[H,2,1] = 20$ and $A[H,2,1] = (A[D,1,1] \cup A[E,1,0]) = \{C, E\}$. Similarly, we calculate the rest of the values in the tables. The complete values of A and D are shown in Fig. 4. Due to limited space we did not show C and L here. According to the table, if we want to assign two MTDs, then the best location for applying the MTDs are on nodes C and H.

Theorem 3. *The complexity of Alg. 2 is* $O(|V|^2 K^2 \Delta)$.

Proof. In Alg. Step 4 to 6 take $O(K|V|)$ to initialize the values of D, T, and A. Steps 9 and 10 take $O(\Delta|V|K)$ in the worst case because they needs to compute the union of sets and the maximum size of the can be $|V|$. Therefore, Steps 7 to 12 dominates the complexity of the algorithm. Steps 7 to 12 take $O(|V|^2 K^2 \Delta)$. Therefore, the complexity of the algorithm is $O(|V|^2 K^2 \Delta + K|V|)$ which is $O(|V|^2 K^2 \Delta)$. □

Theorem 4. *Alg. 2 is produces an optimal MTD assignment.*



(a) Topology I. (b) Topology II. (c) Topology III.

(d) Topology IV.

	Topology I & III	Topology II & IV
Nodes	68	129
Edges	103	149
Attackers	15	32
Resources	5	5

(e) Topology Parameters.

Fig. 5: Randomly generated topologies.

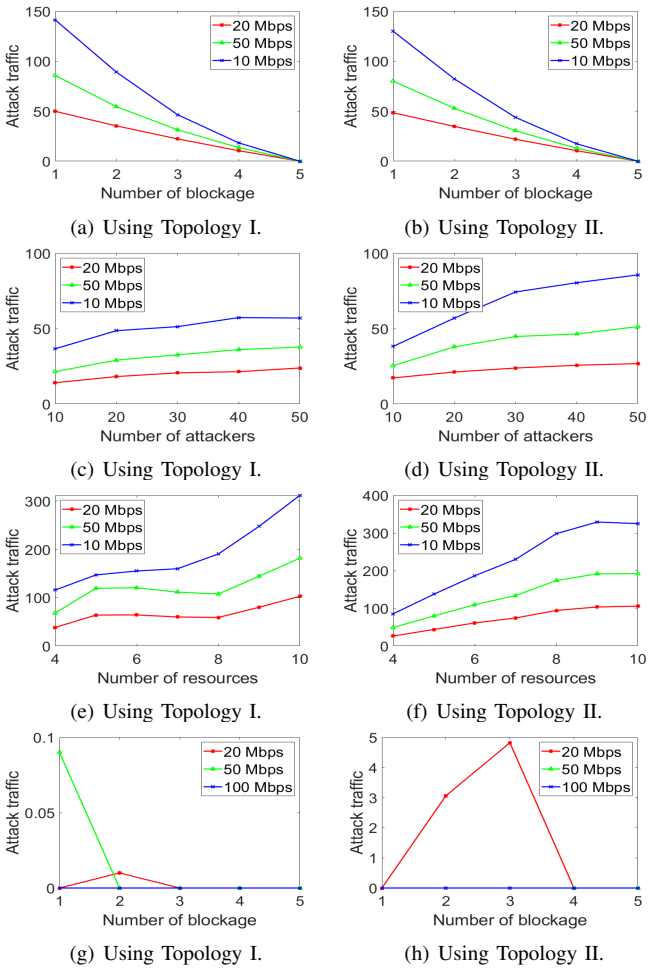
Proof. Alg. 2 uses a dynamic programming bottom-up strategy to search the optimal assignment. For a one-node DAG, if the node color is attached to the attacker, then there is no solution for $K = 0$ and $T = 1$. This is because, without any MTD, the attack will be succeeded in the next step. For $K \geq 1$, there is only one choice for selecting MTDs, which is that node. If that node is selected, the attack is stopped and the number of damaged nodes is 0. In each step, Alg. 2 chooses the best allocation of MTDs to itself or the sub-DAGs. Therefore, Alg. 2 provides an optimal MTD assignment to the nodes through an exhaustive search. □

VI. Simulation

A. Simulation Settings

We built a java simulator to conduct all of the simulations. We want to count the amount of damage and attack traffic reaching the victim for the different settings. We do not need to analyze the real transmission time, link bandwidth, congestion, or packet drop scenarios. Besides the topologies, in this simulation we consider contain hundreds of nodes, links, resources, and attackers. The NS3 or other similar simulators would take a long time to produce results compared to our java simulator.

We use randomly generated topologies for the simulations. We first divide an area of 500×500 square units into 50×50 blocks. A certain number of nodes are placed at random locations in each block. We limit the minimum distance between a couple of nodes. Then, the edges are generated based on the distance between the nodes. When a node is within a certain distance of another node, we add an edge between them. Then a few edges are added by picking up a pair of nodes randomly. Finally, we connect a certain number of resources and attackers to some of the randomly selected nodes. We set the remaining capacities of each edge randomly from a range. The minimum remaining capacity of a link is set to 10 Mbps and different maximum capacity is set for different simulations. In a real-world network, the link maximum capacity varies (100Mbps/1Gbps/10Gbps), and



(a) Using Topology I. (b) Using Topology II.

(c) Using Topology I. (d) Using Topology II.

(e) Using Topology I. (f) Using Topology II.

(g) Using Topology I. (h) Using Topology II.

Fig. 6: Simulation results of Problem 1.

other flows use up some of the link capacities. We do not simulate the other flows so that we set the remaining link capacities randomly. If the link capacity is higher, then the amount of traffic arriving at resources and damages are higher. Topology II relatively large and contains 149 edges and 32 attackers. Simulations related to problem 1 is conducted using Topologies I and II. Topologies III and IV are generated from topologies I and II, respectively. The directions of the links are changed to ensure no cycle in the topologies. Simulations related to problem 2 are conducted using Topologies III and IV. The details are shown in Table 5(e) and Fig. 5.

We measure the performances of our proposed solutions in terms of received attack traffic (RAT) and the amount of damage for different numbers of filters, MTDs, attackers, and resources. We compare the result of the first problem with the optimal solution. The optimal solution is obtained using the brute-force method. As the solution to the second problem is optimal we do not need to compare it with other works. All of the results presented in plots are average of 1000 runs.

B. Simulation Results of Volumetric Attack

We first conduct a simulation to measure the amount of attack traffic for different numbers of filters. Fig. 6(a) shows the amount of attack traffic received by the resources in Topology I. We vary the number of filters from 1 to 5 and keep the number of attackers and resources as in the original topology. For 20 Mbps maximum remaining link capacity, if the number of filters is 1, the amount of RAT is 49.90 Mbps. If the number of filters is 4, the amount of RAT is 10.54 Mbps. For the increase of 3 filters, the attack traffic reduces about 78%. For a higher maximum remaining link capacity, we observe a higher amount of attack traffic received by the resources. For 100 Mbps maximum remaining link capacity, if the number of filters is 1, the amount of RAT is 141.33 Mbps. If the number of filters is 4, the amount of RAT is 18.345 Mbps. For the increase of 3 filters, the attack traffic reduces about 87%. When there are 5 filters, all of the attack traffic is blocked. The amount of attack traffic decreased almost linearly with the increase in the number of filters.

Figs. 6(b) shows the amount of attack traffic received by the resources in Topology II. We keep the same settings as before for this simulation. For 20 Mbps maximum remaining link capacity, if the number of filters is 1, the amount of RAT is 49.90 Mbps. If the number of filters is 4, the amount of RAT is 10.54 Mbps. For the increase of 3 filters, the attack traffic reduces about 78%. For higher maximum remaining link capacity we observe a higher amount of attack traffic received by the resources. For 100 Mbps maximum remaining link capacity, if the number of filters is 1, the amount of RAT is 141.33 Mbps. If the number of filters is 4, the amount of RAT is 18.345 Mbps. For the increase of 3 filters, the attack traffic reduces about 87%. When there are 5 filters, all of the attack traffic is blocked. The amount of attack traffic decreased almost linearly with the increase in the number of filters.

Next, we vary the number of attackers to measure the amount of attack traffic. Fig. 6(c) shows the amount of attack traffic received by the resources in Topology I for different numbers of attackers. We vary the number of attackers from 10 to 50 and keep the number of filters at 3. For 20 Mbps maximum remaining link capacity, if the number of attackers is 10, the amount of RAT is 14.11 Mbps. If the number of attackers is 50, the amount of RAT is 23.9 Mbps. For the increase of 40 attackers, the attack traffic increases about 69%. For higher maximum remaining link capacity, we also observe a higher amount of attack traffic received by the resources. For 100 Mbps maximum remaining link capacity, if the number of attackers is 10, the amount of RAT is 36.65 Mbps. If the number of attackers is 50, the amount of RAT is 57.06 Mbps. For the increase of 40 attackers, the attack traffic increases about 35%. The amount of attack traffic increases almost linearly with the increase in the number of attackers.

Fig. 6(d) shows the amount of attack traffic received by the resources in Topology II for a different number of attackers. We keep the same settings as the previous simulation. For 20 Mbps maximum remaining link capacity, if the number of attackers is 10, the amount of RAT is 17.32 Mbps. If the number of attackers is 50, the amount of RAT is 26.82 Mbps. For the increase of 40 attackers, the attack traffic increases about 54%. For higher maximum remaining link capacity, we

also observe a higher amount of attack traffic received by the resources. For 100 Mbps maximum remaining link capacity, if the number of attackers is 10, the amount of RAT is 38.27 Mbps. If the number of attackers is 50, the amount of RAT is 85.68 Mbps. For the increase of 40 attackers, the attack traffic increases about 12.3%.

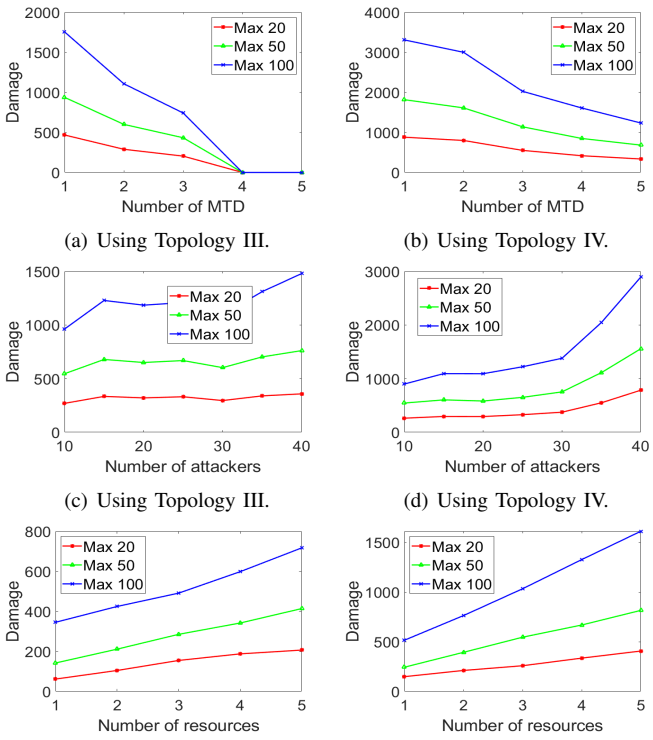
After that, we vary the number of resources and measure the amount of attack traffic. Fig. 6(e) shows the amount of attack traffic received by the resources in Topology I for different numbers of resources. We vary the number of resources from 4 to 10 and keep the number of filters at 3. For 20 Mbps maximum remaining link capacity, if the number of resources is 4, the amount of RAT is 38.44 Mbps. If the number of resources is 10, the amount of RAT is 102.85 Mbps. For the increase of 6 resources, the attack traffic increases about 16%. For 100 Mbps maximum remaining link capacity, if the number of resources is 4, the amount of RAT is 115.86 Mbps. If the number of resources is 10, the amount of RAT is 311.79 Mbps. For the increase of 6 resources, the attack traffic increases about 17%. The amount of attack traffic increases with the increase in the number of resources.

Fig. 6(f) shows the amount of attack traffic received by the resources in Topology II for different numbers of resources. For 20 Mbps maximum remaining link capacity, if the number of resources is 4, the amount of RAT is 26.55 Mbps. If the number of resources is 10, the amount of RAT is 105.74 Mbps. For the increase of 6 resources, the attack traffic increases about 30%. For 100 Mbps maximum remaining link capacity, if the number of resources is 10, the amount of RAT is 85.48 Mbps. If the number of resources is 50, the amount of RAT is 324.99 Mbps. For the increase of 6 resources, the attack traffic increases about 28%. The amount of attack traffic increases with the increase in the number of resources.

Finally, Figs. 6(g) and 6(h) show the difference in attack traffic between the optimal and our approaches. We can observe that most of the cases our proposed approach produce optimal result. Few cases shows little higher than optimal for 20 and 50 Mbps max remaining link capacity in topology I. In topology II, only 20 Mbps remaining link capacity shows higher than optimal. Compared to the total attack traffic the amount of difference is negligible.

C. Simulation Results of Non-volumetric Attack

We conduct a simulation to measure the damage corresponding to different numbers of MTDs. Figs. 7(a) shows the damage caused by the attackers in Topology III. We vary the number of MTDs from 1 to 5 and keep the number of attackers and resources the same as in the original topology. For 20 Mbps maximum remaining link capacity, if the number of MTDs is 1, the amount of damage is 467.66. If the number of MTDs is 4, the amount of damage becomes 0. For higher maximum remaining link capacity we observe higher damage caused by the attackers. For 100 Mbps maximum remaining link capacity, if the number of MTDs is 1, the amount of damage is 1755.3. If the number of MTDs is 4, the amount of damage also becomes 0. When the number of MTDs is more than 4, all of the attack traffic is blocked and no damage caused.



(a) Using Topology III. (b) Using Topology IV.

(c) Using Topology III. (d) Using Topology IV.

(e) Using Topology III. (f) Using Topology IV.

Fig. 7: Simulation results of Problem 2.

Figs. 7(b) shows the damage caused by the attackers in Topology IV. We keep the same settings as before for this simulation. For 20 Mbps maximum remaining link capacity, if the number of MTDs is 1, the amount of damage is 879.96. If the number of MTDs is 5, the amount of damage is 336.33. For the increase of 4 MTDs, the damage reduces about 61%. For higher maximum remaining link capacity we observe higher damage caused by the attackers. For 100 Mbps maximum remaining link capacity, if the number of MTDs is 1, the amount of damage is 3309.9. If the number of MTDs is 5, the amount of damage is 1235.26. For the increase of 4 MTDs, the damage reduces about 63%.

Next, we vary the number of attackers to measure the damage. Fig. 7(c) shows the damage caused by the attackers in Topology III for different numbers of attackers. We vary the number of attackers from 10 to 40 and keep the number of MTDs as 3. For 20 Mbps maximum remaining link capacity, if the number of attackers is 10, the amount of damage is 269.3. If the number of attackers is 40, the amount of damage is 357.83. For the increase of 30 attackers, the attack traffic increases about 32%. For higher maximum remaining link capacity, we also observe higher damage caused by the attackers. For 100 Mbps maximum remaining link capacity, if the number of attackers is 10, the amount of damage is 962.2. If the number of attackers is 40, the amount of damage is 1479.46. For the increase of 30 attackers, the attack traffic increases about 53%. The damage increases about linearly with the increase in the number of attackers.

Fig. 7(d) shows the damage caused by the attackers in Topology IV for different numbers of attackers. We keep the same settings as the previous simulation. For 20 Mbps maximum remaining link capacity, if the number of attackers is 10, the amount of damage is 262.16. If the number of attackers is 40, the amount of damage is 784.06. For the increase of 30 attackers, the damage increases about 200%. For higher maximum remaining link capacity, we also observe higher damage caused by the attackers. For 100 Mbps maximum remaining link capacity, if the number of attackers is 10, the amount of damage is 899.13. If the number of attackers is 40, the damage is 2893.26. For the increase of 30 attackers, the damage increases about 220%. Because of being a larger topology, topology IV gets higher damage than topology III for higher number of attackers.

After that, we vary the number of resources and measure the damage in the network. Fig. 7(e) shows the damage caused by the attackers in Topology III for different numbers of resources. We vary the number of resources from 1 to 5 and keep the number of MTDs as 3. For 20 Mbps maximum remaining link capacity, if the number of resources is 1, the amount of damage is 61.76. If the number of resources is 5, the amount of damage is 206.6. For the increase of 4 resources, the damage increases about 237%. For 100 Mbps maximum remaining link capacity, if the number of resources is 1, the amount of damage is 345.4. If the number of resources is 5, the amount of damage is 717.0. For the increase of 4 resources, the attack traffic increases about 107%. The damage increases with the increase in the number of resources.

Fig. 7(f) shows the damage caused by the attackers in Topology IV for different numbers of resources. For 20 Mbps maximum remaining link capacity, if the number of resources is 1, the amount of damage is 148.56. If the number of resources is 5, the amount of damage is 406.53. For the increase of 4 resources, the attack traffic increases about 175%. For 100 Mbps maximum remaining link capacity, if the number of resources is 1, the amount of damage is 515.06. If the number of resources is 5, the amount of damage is 1611.33. For the increase of 4 resources, the attack traffic increases about 212%. The damage increases with the increase in the number of resources.

Therefore, from the above simulation results we can conclude that the proposed filter assignment approach for volumetric attack performs nearly optimal. The MTD assignment approach for non-volumetric attack depends on the parameters which need to be adjusted based on network properties.

VII. Conclusion

Due to increased threats on the internet, it is important to protect the network-connected resources such as web, database, and file servers. We have considered two types of attacks with different objectives for attackers and formulate the defense mechanism as optimization problems. We considered that the amount of attack traffic is proportional to the damage from the volumetric attacks. We provide an approximation filter assignment solution to block the maximum amount of attack traffic with a limited number of filters. The amount of attack traffic is not related to the damage caused by the non-volumetric attack. We considered that the steps passed by the attacker are equal to the damage. We propose a dynamic programming-based optimal solution to assign a moving target defense approach to some nodes. We conducted extensive simulations to observe behaviors of the defense mechanisms for different settings. We also observe that the approximation solution is very close to the optimal solution.

References

[1] J. Wang and I. C. Paschalidis, "Statistical Traffic Anomaly Detection in Time-Varying Communication Networks," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, Jun 2015.

[2] W. Wei, F. Chen, Y. Xia, and G. Jin, "A Rank Correlation Based Detection against Distributed Reflection DoS Attacks," *IEEE Communications Letters*, vol. 17, no. 1, Jan 2013.

[3] A. Kulkarni and S. Bush, "Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics," *J. Netw. Syst. Manage.*, vol. 14, no. 1, Mar. 2006.

[4] T. A. Ahanger, "An effective approach of detecting DDoS using Artificial Neural Networks," in *2017 International Conference on Wireless Communications, Signal Processing and Networking*, Mar 2017.

[5] D. Almomani, M. Alauthman, F. Albalas, O. Dorgham, and A. Obeidat, "An Online Intrusion Detection System to Cloud Computing Based on Neucube Algorithms," *International Journal of Cloud Applications and Computing*, vol. 8, 04 2018.

[6] M. Suk Kang, V. D. Gligor, and V. Sekar, "SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks," in *Network and Distributed System Security Symposium*, Jan 2016.

[7] J. Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," in *Network and Distributed System Security Symposium*, Mar 2002.

[8] S. B. Lee, M. S. Kang, and V. D. Gligor, "CoDef: Collaborative Defense Against Large-scale Link-flooding Attacks," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, Dec 2013.

[9] J. M. Smith and M. Schuchard, "Routing Around Congestion: Defeating DDoS Attacks and Adverse Network Conditions via Reactive BGP Routing," in *IEEE Symposium on Security and Privacy*, May 2018.

[10] N. Polatidis, M. Pavlidis, and H. Mouratidis, "Cyber-attack path discovery in a dynamic supply chain maritime risk management system," *Computer Standards Interfaces*, vol. 56, pp. 74–82, 2018.

[11] X. Liu, "A network attack path prediction method using attack graph," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–8, 2020.

[12] M. Ge, J. B. Hong, S. E. Yusuf, and D. S. Kim, "Proactive defense mechanisms for the software-defined internet of things with non-patchable vulnerabilities," *Future Generation Computer Systems*, vol. 78, pp. 568–582, 2018.

[13] M. Touhiduzzaman, A. Hahn, and A. K. Srivastava, "A diversity-based substation cyber defense strategy utilizing coloring games," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5405–5415, 2019.

[14] R. Biswas and J. Wu, "Optimal filter assignment policy against distributed denial-of-service attack," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020.

[15] A. Kanevsky, "Finding all minimum-size separating vertex sets in a graph," *Networks*, vol. 23, no. 6, pp. 533–541, 1993.