

# Internet Infrastructure Security: A Taxonomy

Anirban Chakrabarti and G. Manimaran  
 Dependable Computing & Networking Laboratory  
 Dept. of Electrical and Computer Engineering  
 Iowa State University, Ames, IA 50011  
 {anirban,gmani}@iastate.edu

**Abstract**—The pervasive and ubiquitous nature of the Internet coupled with growing concerns for cyber terrorism demand for immediate solutions for securing the Internet infrastructure. So far, the research in Internet security primarily focused on securing the information rather than securing the infrastructure itself. Given the prevailing threat situation, there is a compelling need to develop architectures, algorithms, and protocols to realize a dependable Internet infrastructure. In order to achieve this goal, the first and foremost step is to develop a comprehensive understanding of the security threats and existing solutions. This paper attempts to fulfill this important step by providing a taxonomy of security attacks which are classified into four main categories: DNS “hacking”, routing table “poisoning”, packet “mis-treating”, and denial-of-service attacks. The paper also discusses the existing solutions for each of these categories, and also outlines a methodology for developing secure protocols.

## I. INTRODUCTION

The Internet has been witnessing enormous growth over the last several years. Until now, the main research focus has been on improving the performance and scalability of the Internet. Although, the performance and scalability have their place in Internet research, the enormity of the Internet has forced the research community to look at the dependability aspects of the Internet. The Internet, like any other product, is prone to failures and researchers have started to realize the importance of dependable communication to tolerate *device failures* (e.g., link and node failures) and to overcome the presence of *malicious users* or “hackers”. The importance of securing the Internet has grown rapidly due to a series of attacks that shut down some of the world’s most high profile Web sites, including Amazon and Yahoo. Several such attacks have also been reported in CERT advisories [1]. These attacks, coupled with the growing fear of *cyber-terrorism*, have made researchers think of possible means and methods to protect users from the adversaries.

Securing the Internet, like any other fields of computers, is based on the principle of *confidentiality* and *integrity*. Confidentiality indicates that all data sent by users should be accessible to only “legitimate” receivers, and integrity indicates that all data received should only be sent/modified by “legitimate” senders. These principles exist in every field, but the presence of *packet sniffers*, *malicious routers*, *covert channels*, and *eavesdroppers* in the Internet makes this extremely important problem quite challenging ([2] and references therein).

The past several years have seen a surge of Internet security research in the field of *information assurance*, which primarily

focused on protecting the data using techniques such as *authentication* and *encryption*. However, information assurance assumes that the devices responsible for encrypting, forwarding, and sending of packets are trustworthy. Scientists are now questioning these assumptions, as instances have taken place where the network infrastructure (e.g., routers, servers) are compromised to the advantage of malicious adversaries. Thus, network infrastructure security is clearly a pressing need, especially in light of recent national attacks, as the attacks have the potential for affecting the entire Internet infrastructure, which may have serious consequences on the security and economic vitality of the society. As Richard Clarke, Homeland security adviser for combating cyber terrorism, puts it (CNN news, Oct. 9, 2001): “Our very way of life depends on the secure and safe operations of critical systems that depend on cyberspace”. Therefore, infrastructure security is a pressing issue which needs immediate research attention ([3], [4], [5]).

## II. MOTIVATION

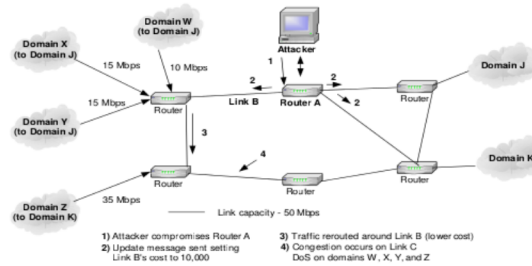


Fig. 1. An example of a router attack and its consequences

Attack on the Internet infrastructure can lead to enormous destruction, as different infrastructure components of the Internet have implicit trust relationship with each other. Consider a scenario listed in Figure 1. In this scenario, an intruder wishes to attack Domain Z, which contains a high-profile server. Most of the links are fairly heavily loaded but are under capacity (70-80% usage). The attacker compromises Router A, so that the router increases the cost of link B to an artificially high value, say 10,000. Traffic, in the Internet, is generally routed along the shortest path. Since link B has a high cost, packets will be routed around B. Thus, packets will be routed through the border router of domain Z. This causes enormous congestion at domain Z. Artificial congestion, thus created, will slow down

the services to the clients of domain  $Z$  (also  $W$ ,  $X$ , and  $Y$ ), and many clients will be denied access to the server.

As shown with this fairly simple example, it is possible for an attacker to create a large amount of service disruption. Such service disruption has already been noticed with untargeted breaches in the infrastructure such as fiber cuts as well as BGP routing flaps due to Nimbda/Code Red. Also, these types of attacks are very difficult to detect as the attacker is hidden during the actual transmission of packets. The attacker can achieve the same results as above, by sending more packets than the border router of domain  $Z$  can handle. In addition, the Router A can also misroute packets causing congestion at the border router of domain  $Z$ . Thus, compromising the infrastructure can lead to potentially dangerous attacks in the Internet.

The effect of the types of attacks mentioned above are dangerous because the attacker knows the network topology and intelligently takes advantage of the basic flaws of the networking protocols. Though security research in this area is considered absolutely necessary, there has been a dearth of a framework which would encompass all the possible attack scenarios in the Internet. Because of the lack of a guiding framework, research efforts in Internet security have lacked direction. Research efforts have been undertaken by need of the hour rather than to achieve long term goals of achieving secure communication over the Internet. In this paper, we present an Internet security attack taxonomy to fill the void in this area and identify the areas which require immediate research attention. The objectives of our paper are:

- Categorize the possible Internet infrastructure attacks.
- Identify the attacks within each categories.
- Identify existing solutions which deal with the attacks.
- Present guidelines for important and less researched areas.

### III. TAXONOMY OF INTERNET INFRASTRUCTURE ATTACKS

In order to achieve the goals mentioned in the previous section, the attacks need to be categorized. The Internet infrastructure attacks can be broadly classified into the following four categories as shown in Figure 2.

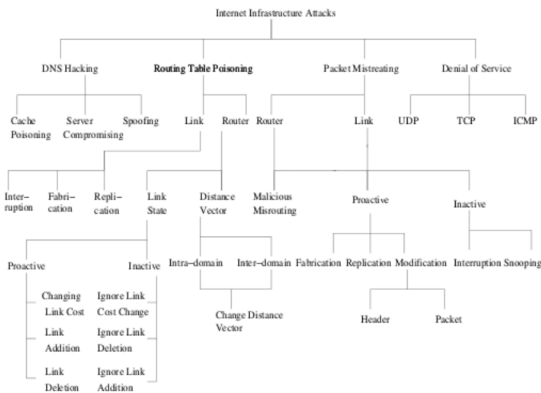


Fig. 2. Types of security attacks on the Internet

**DNS “hacking” attacks:** Domain Name System (DNS) is a distributed, hierarchical global directory that translates machine/domain names to numeric IP addresses. The DNS infrastructure consists of 13 root servers at the top layer, top-level domain (TLD) servers (.com and .net), as well as country code top-level domains (.us, .uk and so on) as the lower layers. Due to its ability to map human memorable names to numerical addresses, its distributed nature, and its robustness, DNS has evolved into a critical component of the Internet. Therefore, an attack on the DNS infrastructure has the potential to affect a large portion of the Internet [6].

**Routing table “poisoning” attacks:** Routing tables are used to route packets over the Internet. They are created by exchange of routing information or updates between routers. Poisoning attacks refer to the malicious modification or “poisoning” of routing tables. This can be achieved by maliciously modifying the routing information update packets required by the routing protocols. This attack can result in wrong entries in the routing table and could lead to a breakdown of one or more domains of the Internet [7], [8].

**Packet “mistreating” attacks:** In this type of attacks, the malicious router mishandles packets, thus resulting in congestion, denial-of-service, and so on. The problem becomes intractable if the router selectively interrupts or misroutes packets resulting in *triangle routing* [9], that is loop formation. An example of triangle routing is shown in Figure 3. The shortest path from 1 to 4 is 1–2–4, and the shortest path from 3 to 4, is 3–1–2–4. Let 2 be the malicious router. Whenever 2 gets a packet from 1 destined for 4, it routes it to 3. Since the shortest path from 3 to 4 is through 1, a loop is created. This type of attacks are very difficult to detect.

**Denial of Service (DoS) attacks:** These attacks are directed at specific hosts with an intention of breaking into the system or causing denial of service [10]. These attacks may be carried out by individuals or groups who may use such attacks for personal gain or notoriety. These attacks become extremely dangerous and hard to prevent if a group of attackers coordinate in DoS. This type of attacks are called Distributed Denial-of-Service (DDoS) attacks. It is to be noted that, DoS can also result from routing table “poisoning” and packet “mistreating”. We categorize DoS attacks as those attacks, which are directed towards the end-system rather than towards the transmission infrastructure like the routers/links.



Fig. 3. Example of “triangle routing”

Among these attacks, the last type (DoS attacks) is related to end systems while the other three are related to the network infrastructure (DNS, backbone routers and communication links). In fact, most of the traditional security research has not focused on transmission system related attacks, rather focused on the security of the end-system. Although DoS attacks against spe-

cific machines are an important threat, the potential for attacks against the transmission infrastructure can result in a massive DoS attack against entire groups or whole portions of the Internet. Thus, the routers and the other networking infrastructure components represent ideal targets for disrupting the national infrastructure.

We describe each of the three infrastructure attacks with existing solutions in Sections IV, V, VI, and VII. In Section VIII, we summarize all the state-of-the-art research in this area along with future directions.

#### IV. DNS “HACKING” ATTACKS

DNS attacks are being focused in this section. Attacks of this type have illustrated the lack of authenticity and integrity of the data held within DNS as well as in the protocols that use host names as an access control mechanism.

##### A. Impact of “hacking”

DNS, being a critical infrastructure, is contacted by all hosts during accessing servers and starting connections. The impact of DNS attacks is quite widespread that include:

**Denial-of-Service:** DoS is one of the most dangerous impacts of DNS “hacking”. DoS can be achieved in several ways: One way is to send back negative responses indicating that the DNS name does not exist. Another way is to redirect the client’s request to a server which does not contain the service the client is requesting. DoS attacks on DNS servers can also achieve the same objective with greater effect.

**Masquerading:** The adversary can use DNS attacks to redirect communication to masquerade as a trusted entity. If this is accomplished, an attacker can intercept, analyze, and/or intentionally corrupt the communications [6].

**Information Leakage:** DNS threats also include leakage of information concerning internal networks to an attacker. Frequently, host names can represent project names that may be of interest revealing the operating system of the machine.

**Domain Hijacking:** By compromising insecure mechanisms used by customers to update their domain registration information, attackers can take over the domain registration process to hijack legitimate domains.

##### B. Types of “hacking”

DNS consists of a distributed database which lends to its robustness and also leads to various types of vulnerabilities, which can be categorized into three main types:

1) **Cache poisoning:** Generally, to hasten the process of query response, DNS servers store the common information in a cache. If the DNS server is made to cache bogus information, the attacker can redirect traffic intended for legitimate site to a site under the attacker’s control.

2) **Server compromising:** Attackers can compromise a DNS server, thus giving them the ability to modify the data served to the users. These compromised servers can be used for cache “poisoning” or DoS attacks on some other server.

3) **Spoofing:** In this type of attack, the attacker masquerades as a DNS server and feeds the client wrong and/or potentially malicious information. This type of attack can also redirect the traffic to site under attacker’s control and also launch a DoS attack on the unsuspecting client.

In order to address DNS attacks, the IETF added security extensions to DNS, collectively known as DNSSEC [11].

**Outline of DNSSEC:** DNSSEC provides authentication and integrity to the DNS updates. All of the previously mentioned DNS attacks are mitigated with the addition of data origin authentication, and transaction and request authentications. The authentications are provided through the use of digital signature technology. The digital signature contains the encrypted hash of the RRset (Resource Record Set). The recipient can then check the digital signature against the received data. To make the DNSSEC proposals valid, secure servers and secure client environment must be created. Also, DNSSEC is unable to provide security against information leakage as it is mainly concerned with authentication.

#### V. ROUTING TABLE “POISONING” ATTACKS

In this section, we focus our attention to the routing table “poisoning” threat. It is a challenging problem, because security was not introduced into the routing protocols from the start. It is important because the routing table forms the basis of the Internet and any corruption of routing table may lead to significant consequences.

##### A. Impact of “poisoning”

Routing table poisoning could have the following impacts:

**Sub-optimal routing:** With the emergence of Internet as a means for supporting soft real-time applications, optimality in routing assumes significant importance. Routing table “poisoning” attacks can result in sub-optimal routing that can affect the real-time applications over the Internet.

**Congestion:** Routing table “poisoning” can lead to artificial congestion if packets are forwarded to only certain portions of the network. Artificial congestion, thus created, is not solved using the traditional congestion control mechanism.

**Partition:** Wrong entries in the routing table may result in the creation of artificial partitions in the network. This becomes a significant problem as hosts residing in one partition will be unable to communicate with hosts residing in the other partition.

**Overwhelmed host:** Routing table “poisoning” may be used as a weapon for DoS attacks. If a router sends updates which results in concentration of packets to one or more selected servers, the servers can be taken out of service because of huge amounts of traffic (such an attack is illustrated in Figure 1). This type of DoS attack is more potent as the attacker is not “spoofing” identity, and is thus impossible to detect by the detection techniques mentioned in [12], [13].

**Looping:** The creation of triangle routing caused due to packet “mistreating” attacks (see Section III) could also be simulated through improper updation of the routing table.

**Access to data:** Adversaries may gain illegal access to data through the “poisoning” of routing table attack. This may lead

to adversaries snooping packets, which were not supposed to pass through that part of the network.

With all of these possible attacks, routing table “poisoning” has the potential to be a killer DoS attack for those wishing to wage cyber-warfare with devastating effect. Unfortunately, this particular field of security research has not received as much attention as it needs. In the next few sections, we identify the different types of routing table attacks and discuss the known solutions to the problem.

### B. Types of “Poisoning”

The majority of work on routing protocols for the Internet has proceeded in two main directions: distance vector protocols (e.g. RIP [14]) and link state protocols (e.g. OSPF [15]). Since both link state and distance vector protocols exhibit different characteristics in state information and their exchange, and route computation, they are exposed to different types of vulnerabilities, which provide unique sets of challenges for securing them. In a link state protocol, each node periodically floods the state of its links to all the nodes in the network. After receiving the link state updates (called a Link State Advertisement (LSA) in OSPF), each router computes the shortest path tree (SPT) with itself as the root of the tree. In a distance vector protocol, each node sends its routing distances (in the form of distance vector packet) to its neighbors. A neighbor upon receiving the distance vector packet, updates its routing table, if necessary. Thus the distance vector routing protocols, unlike link state routing protocols, suffer from lack of the full topology information at each node. This lack of knowledge lends to a variety of attacks that are not possible in the case of link state protocols. Also, typically link state protocols consist of two phases: hello phase and link state advertisement. In hello phase hello packets are exchanged for neighbor establishment. Link states are then advertised infrequently (every 30 minutes), or if a link fails. In case of distance vector protocols, hello phase is not present. Therefore, distance vector updates are exchanged even for neighbor establishment. Hence, distance vector protocols consume more bandwidth than link state protocols.

As mentioned in the Figure 2, routing table poisoning can be broadly categorized into (a) link and (b) router attacks. Link attacks, unlike the router attacks, are similar in case of both link state and distance vector protocols.

1) **Link Attacks:** Link attacks occur when the adversary gets access to a link. Thus, the adversary can intercept, interrupt, and/or modify the routing messages. Current routing protocols employ techniques to prevent these types of attacks. The various forms of link attacks and their known solutions are discussed below:

**Interruption:** Routing information can be intercepted by an adversary, and the information can be stopped from propagating further. However, interruption is not effective in practice. The reason for this is that, in the current Internet scenario there is generally more than one path between any two nodes, since the average degree of each node is quite high (around 3.7). Therefore, even if an adversary stops a routing update from propagating, the victim may still be able to obtain the information from other sources.

**Solutions:** Most routing protocols employ robust updates between neighbors [14], [15], by using acknowledgments. Link attacks are detected in those cases. However, if links interrupt selectively, it is possible to have unsynchronized routing tables throughout the network. The after-effects of such routing tables is looping and denial-of-service. Unsynchronized routing tables can also be created if a router drops the updates, but sends an acknowledgment. The problem of router dropping routing updates selectively has not been studied in the literature.

**Modification/Fabrication:** Routing information packets can be modified/fabricated by an adversary who has access to a link in the network.

**Solutions:** Digital signatures [16] are used for the integrity and authenticity of messages. In the case of digital signatures, the sender signs the packets with its private key, and all nodes can verify the signature based on the sender’s public key. In this case, the routing updates increase by the size of the signature (typically between 128 to 1024 bits). This is a viable solution in link state routing protocols, since the LSAs are transmitted infrequently. This is also proposed as a solution for distance vector protocols.

**Remarks:** Distance vector protocols suffer from excessive bandwidth consumption as the distance vectors are exchanged quite frequently. Therefore, the addition of extra overhead in the form of digital signature has been looked upon by the research community with concern. Efforts have been undertaken to reduce the overhead through the use of efficient digital signatures [17]. Another problem with this approach is that it relies on the existence a public key infrastructure (PKI) for its functioning [16]. In absence of a PKI, the proposed solutions are not viable.

**Replication:** Routing table “poisoning” can also be in the form of replication of old messages, where a malicious adversary gets hold of routing updates and replays them later. This type of attacks cannot be solved using digital signature schemes, because the updates are valid, only they are time shifted.

**Solutions:** Sequence information are used to prevent this attack [8]. Sequence information can be in the form of sequence numbers or time stamps. An update is accepted as a valid update if the sequence number in the packet is greater than or equal to the sequence number of the previously received update from the same router [14].

**Remarks:** This solves the problem of replication, however the packets within the same clock period can be replayed if the time stamp is used as sequence information. No remedy has been found for this problem. However, this problem has limited effect as it can be employed only if a router sends multiple updates within the same time period.

2) **Router Attacks:** A router can be compromised, making it *malicious* in nature. Router attacks differ in their execution depending on the nature of the routing protocol. In case of link state routing protocol, a router sends information about its neighbors. Hence, a malicious router can send incorrect updates about its neighbors, or remain silent if the link state of the neighbor has actually changed. However, in the case of distance vector protocols, routers can send wrong and potentially dangerous updates regarding any nodes in the network, since the nodes do not have the full network topology. Router attacks,



in the case of both link state and distance vector protocols are very difficult to prevent if the routers exhibit *Byzantine faults* [18]. Router attacks require significant research attention, as very little research efforts have been undertaken in this direction. The different types of router attacks and known solutions are described below:

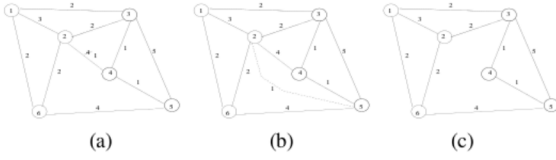


Fig. 4. Link State attack scenarios

**Link State Router Attacks:** A router attack can be *proactive* or *inactive* in nature. In case of proactive router attack, the malicious router can add a fictitious link, delete an already existing link, or change the cost of a link proactively. In case of inactive router attacks, a router ignores a change in link state of its neighbors. Examples of proactive link state attacks are shown in the Figure 4. *Node2* is the malicious node in the network. In Figure 4(a), *node2* advertises the cost of link (2 – 4) as 1, instead of 4. The link state protocol is unable to detect the attack. As a result, all of the nodes in the network assume that link (2 – 4) has a cost of 1. Therefore, *node1* computes the shortest path to *node5* through *node2*, instead of *node6*. Hence, not only the connection is sub-optimal, but also the attacker (*node2* in this case) gets access to data which it was not supposed to get. The same results can be achieved by the malicious node 2 by advertising that there exists a fictitious link between 2 and 5 having a cost of 1. This attack has been shown in Figure 4(b). In Figure 4(c), the node advertises that there is no link between *node2* and *node4*.

**Solutions:** The solutions proposed for router attacks in link state protocols can be categorized into two types: *intrusion detection* and *protocol-driven*. Use of intrusion detection techniques have been suggested as a mechanism to detect router attacks [7]. In these techniques, a centralized attack analyzer module detects attacks based on some possible alarm events sequences. Using an attack analyzer module in Internet scenario is not a scalable solution. In a protocol-driven solution, the detection capability is embedded in the link state protocol itself. In [19], Secure Link State Protocol (SLIP) has been proposed, where attack detection capability has been incorporated in the routing protocol itself. A router does not believe an update, unless it receives “confirmation” link state update from the node supporting the questionable link. However, the solution is not complete as it works only in symmetric network where both nodes supporting a link can identify the change in the link state. It also makes an assumption that no malicious collusion exists in the network.

**Distance Vector Router Attacks:** In distance vector protocols, if a malicious router creates a wrong distance vector and sends it all its neighbors, the neighbors accept the update since there is no way to validate it. As the router itself is malicious, standard techniques like digital signatures does not work.

**Inconsistency Detection** In [8], the authors have proposed a validation scheme through the addition of predecessor informa-

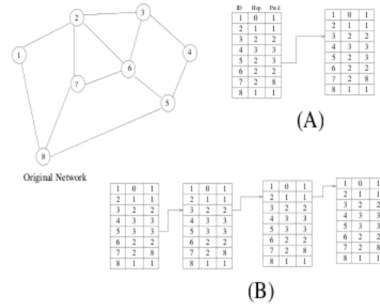


Fig. 5. An example illustrating the working of Consistency Check algorithm

tion in the distance vector update. Figure 5 shows the working of the algorithm proposed in [8] (referred to as Consistency Check algorithm or CC). Let *node1* send its distance vector to all its neighbors. In the figure the distance vector updates are represented as entries having three columns: destination node id, shortest distance, and the predecessor for each shortest path. Whenever a node receives the distance vector from *node1*, it carries out consistency check by tracing the path from each destination to *node1*. Two instances are illustrated in Figure 5A and 5B. In Figure 5A, the distance vector update from *node1* claims that the predecessor of *node5* is not *node8*, but *node3*. This type of inconsistency in the update can be identified by the CC algorithm. Figure 5(B) shows an example when the CC algorithm fails. Let the routing update sent by *node1* has distance entry to *node5* as 3 instead of 2, and the predecessor as *node3* instead of *node8*. Tracing to source *node1* indicates that the update is consistent. Therefore, in this case, the CC algorithm is unable to detect an incorrect update. The above example shows that the CC algorithm is unable to detect router attacks when the malicious router changes the update intelligently, keeping the network topology in mind. Though this is an important issue, there is not much work done to solve the various problems associated with this issue, and hence it requires significant research attention.

## VI. PACKET “MISTREATING” ATTACKS

Adversaries may get hold of actual data packet and mistreat them. This is an attack during data transmission phase, unlike the “poisoning” attack. Packet “mistreating” attacks have limited effectiveness compared to the routing table “poisoning” and DoS attacks. This is because, the attacks are limited to a part of the network rather than the whole network as in the case of “poisoning” attacks. However, this type of attacks are possible and are very difficult to detect.

### A. Impact of “mistreating”

Referring back to Figure 2, the third broad category of attacks is the packet “mistreating” attack. Though limited in their effectiveness, packet “mistreating” attacks can result in:

**Congestion:** Similar to “poisoning”, packet “mistreating” attacks can also result in congestion in the network. Congestion is caused by misrouting the packets to heavily loaded links.

**Lowering Throughput:** “Mistreating” attacks can result in lowering of connection throughput. Malicious adversaries can prevent TCP packets from propagating further. The source, sensing congestion, lowers the sending window resulting in drop in connection throughput.

**Denial-of-service:** Packet “mistreating” attacks can be used to indirectly cause denial-of-service attacks by directing an uncontrollable number of packets towards a victim.

### B. Types of “mistreating”

Similar to the “poisoning” attacks, an adversary can gain access to a link resulting in link attacks, or get access to a router resulting in router attacks.

1) **Link Attacks:** An adversary, on gaining access to a link, can interrupt, modify/fabricate or replicate data packets.

**Interruption:** As mentioned earlier, interruption of TCP packets may reduce the overall throughput of the network. One of the earliest work dealing with this subject is WATCHERS project [9]. The project is based on the “principle of conservation” i.e. the number of packets going into any node is equal to the number of packets going out, excluding the number of packets destined for that node. This solution is not practical in Internet settings, as packets may be dropped because of legitimate reasons (e.g., congestion). Another interesting work in this area is reported in [20], where the authors first showed that selectively dropping even a small number of packets can degrade the performance of TCP to a large extent. The authors used packet dropping profiles and intrusion detection to identify the attacks. These are the only solutions attempted to detect these types of attacks. However, questions remain regarding the scalability of intrusion detection techniques over the Internet.

**Modification/Fabrication:** Similar to routing updates, data packets can be modified/fabricated by adversaries. IPSec, the standard protocol suite for adding security features to the IP layer of the Internet [21], provides authentication and encryption for the data packets over the Internet.

**Replication:** In order to counter replay attacks, IPSec incorporates a small protocol, called anti-replay window protocol. This protocol can provide anti-replay service by including a sequence number in each IPSec message and using a sliding window. The description of IPSec given below describes the anti-replay protocol and its deficiencies.

**Description of IPSec:** IPSec is used as a standard authentication and encryption protocol over the Internet. These objectives are met through the use of two traffic security protocols, the Authentication Header (AH) and the Encapsulating Security Payload (ESP), and through the use of cryptographic key management procedures and protocols. AH provides authentication for as much of the IP header as possible, as well as for upper layer protocol data. However, some IP header fields may change in transit and the value of these fields may not be predictable by the sender, when the packet arrives at the receiver. The values of such fields cannot be protected by AH. The ESP header is inserted after the IP header and before the upper layer protocol header. Other than complexity, IPSec is immune from basic flaws and is thus quite widely used.

IPSec also incorporates anti-replay mechanism. According to IPSec, a unidirectional security association can be established between any two computers in their network (source and destination). The source keeps a counter for the sequence numbers used for sending messages and includes the current value of the sequence number with any messages sent. The destination uses a sliding window to determine whether a received message is a normal message or a replayed message. If the sequence number of the received message is less than the number represented by the left edge of the window, then the message is regarded as a replayed message and is discarded by the destination. If the sequence number of the replayed message falls inside the window, the destination can determine whether the message is a replayed message or not by checking the information kept in the window. If the sequence number of the received message is larger than the number represented by the right edge of the window, the message is accepted as a fresh message and the right edge is made equal to the received sequence number. This method, though effective, can result in discarding of good messages. A solution to this problem was suggested in [22], where the authors presented a controlled shift mechanism, which results in discarding of fewer number of good messages.

2) **Router Attacks:** Malicious routers can cause all the link attacks. In addition to such attacks, malicious routers can misroute packets. Malicious misrouting of packets may result in congestion, or can even be used as a DoS attack. These types of attacks have not been studied in detail in the literature. The attack is mentioned in Cisco white papers [10], where packets sent and received by the same interface of a router are discarded. This simple filtering scheme can prevent a naive misrouting attack. However, a malicious router can create triangle routing or looping which is an open problem (refer to Figure 3).

## VII. DENIAL-OF-SERVICE ATTACKS (DoS)

In these sort of attacks, the packets are routed correctly but the destination becomes the target of the attackers [1]. DoS attacks are very easy to generate and are very difficult to detect, and hence are attractive weapons for the hackers. In a typical DoS attack, the attacker node spoofs its IP address and uses multiple intermediate nodes to overwhelm other nodes with traffic. DoS attacks are typically used to take important servers out of action for a few hours, resulting in DoS for all the users served by the server. It can also be used to disrupt the services of the intermediate routers.

Generally, DoS attacks can be categorized into two main types: (a) ordinary and (b) distributed. In an ordinary network-based denial of service attack, an attacker uses a tool to send packets to the target system. These packets are designed to disable or overwhelm the target system, often forcing a reboot. Often, the source address of these packets is spoofed, making it difficult to locate the real source of the attack. In the Distributed DoS (DDoS) attack, there might still be a single attacker, but the effect of the attack is greatly multiplied by the use of attack servers known as “agents”. To get an idea of the scope of this attack, over 1,000 systems were used at different times in a concerted attack on a single server at the University of Minnesota. The attack not only disabled that server but denied access to a very large university network [1].

As mentioned in the previous sections, routing table poisoning and packet mistreating attacks are capable of causing denial-of-service. Also, new techniques are being invented every day to create denial-of-service attacks, following are the common types of attacks:

**UDP Flood:** UDP flood technology is used by the hackers to launch a DoS attack. For example, by sending UDP packets with spoofed return addresses, a hacker links one system's UDP character-generating (chargen) service to another system's UDP echo service.

**TCP/SYN Flood:** In this type of attacks, the hacker sends a large volume of SYN packets to a victim. The return addresses of the packets are spoofed. Thus, the victim queues up SYN-ACKs but cannot continue sending them because it never receives ACKs from the spoofed addresses.

**ICMP/Smurf:** In this type of attacks, the hacker broadcasts ICMP ping requests with the return address spoofed to show the ultimate victim's address, to a large group of hosts on a network. The hosts send their responses to the ultimate victim, whose system is overwhelmed and cannot provide service.

#### A. Types of Solutions

Solutions proposed in literature for DoS attacks, can be broadly categorized as (i) Preventive and (ii) Reactive. Preventive DoS solutions take precautionary steps in preventing DoS attacks. A wide range of solutions have been proposed, however, this problem still remains an open one. The reactive solutions aim at identifying the source of the attacks. This is very important because attackers spoof their addresses, thus techniques are needed to trace back to the source of the attack. We discuss in this section some of the interesting solutions.

1) **Preventive:** All preventive DoS detection techniques are based on some prior information, on the basis of which the filtering is carried out. A few filtering techniques are described in Cisco white papers [10]. These strategies are currently employed in Cisco routers to combat DDos attacks. Strategies like unicast reverse path verification, SYN packet rate control, checking of outgoing and incoming interfaces are some of the techniques that help to weed out majority of DoS attacks. In [23], the authors have presented two techniques for preventing DoS attacks through filtering techniques. They presented a technique called Distributed Packet Filtering (DPF), where decision to drop or accept the packet is made based on the incoming packet interface. The main problem hindering these approaches is that they need constant upgrade, as the adversaries become more and more sophisticated. Also, these techniques can filter only those packets which are considered malicious based on some filtering algorithm. This reduces the effectiveness of these techniques.

2) **Reactive:** Reactive techniques aim at identifying the attacker after the attack has been completed. This is an active area of research because the current identification techniques are totally manual, and may span over months. The current solutions can be broadly categorized into: (i) link testing, (ii) logging, (iii) ICMP traceback, and (iv) IP traceback.

**Link Testing:** This technique involves iteratively checking the upstream link until the source is reached. This type of identification technique assumes that the attack remains active after

the completion of the trace. One type of link testing approach is called *input debugging*, where routers develop an attack signature based on some attack pattern. The victim informs the operator about the signature which then checks the packets, and iteratively carries out this process. This is employed in some routers now, though the process is time-consuming. Another suggested link testing is through *controlled flooding* [12]. In this type of technique, the victim floods all the links based on the assumption that packet drop taking place from an attacked link is much more than from any other link. This technique suffers from being a mode of DoS attack by itself.

**Logging:** A simple technique has been suggested in [13], where logging of data packets are done at key routers. Traceback is carried out by using data mining techniques. This technique suffers from scalability problem, as enormous resources are required to carry out logging based identification.

**ICMP Traceback:** In the Internet draft [24], the author has proposed a scalable technique where each router stores packet with a low probability (1/20000). Whenever a packet is stored the router sends ICMP traceback message towards the destination. When attacked, the destination can traceback to the source based on the router ICMP messages. This scheme has a problem as the ICMP messages can be used by an adversary to cause DoS attacks.

**IP Traceback:** One of the earliest efforts to identify the source of the packet through IP traceback was done in [25]. In this technique, a router marks any packet flowing through it with a very small probability. Getting sufficient number of packets (in case of DoS attacks), the destination can retrace the packets back to the source, based on the information in the marked packet. This scheme was further extended in [26], where the authors showed that using partial network information, the number of packets required to traceback can be substantially reduced. Another interesting work in this area is reported in [27], where the authors have presented a hash-based technique for IP traceback that generates audit trails for traffic within the network. The origin of packets can be traced back to the source based on the audit trails.

## VIII. CONCLUSIONS AND FUTURE WORK

The pervasive nature of the Internet coupled with recent threats for cyber terrorism makes Internet Infrastructure security an area of significant importance. In this paper, we have presented a taxonomy of Internet infrastructure security attacks and discussed known solutions for some of the attacks. Our survey of the attacks and solutions (summarized in Table I) reveal that there are several important security issues which need immediate research attention. These include the following:

- Scalability and deployment issues in DNSSEC.
- Robust routing protocols to prevent routing table "poisoning" attacks. These include secure versions of link state protocols (e.g. OSPF) and distance vector protocols (e.g. RIP). The problem of securing the protocols from router attacks require significant research attention.
- Secure router design to prevent packet "mistreating" attacks. No work has been carried out to solve the packet "mistreating" attacks when a router is responsible for "triangle routing".

TABLE I  
SUMMARY OF INTERNET THREATS, ATTACKS, SOLUTIONS

Category	Type	Attacks	Solutions	Remarks
DNS "hacking"	All	All	DNSSEC [11]	Assumes secure client/server, no security against leakage
Routing table "Poisoning"	Link	Interruption	Acknowledgments [14], [15]	Attack has limited significance
		Modification/ Fabrication	Digital Signatures [14]	Excessive overhead in distance vector protocols, assumption that PKI exists
		Replication	Sequence Numbers [14], [15]	Updates within the same time period can be replayed (limited effect)
	Router	Link State	SLIP [19]	Assumes symmetric network and no collusion
			JiNaO [7]	Not scalable
		Distance Vector	Consistency Checks [8]	Unable to detect consistency attacks
Packet "mistreatment"	Link	Interruption	WATCHERS [9], packet profile [20]	Not scalable
		Fabrication/ Modification	IPSec [21]	High complexity
		Replication	IPSec	Unnecessary dropping of good packets
DoS	All	All	Filtering [10], [23]	Can prevent limited attacks
			Link Testing [12]	Not scalable, may be a tool for DoS attacks
			Logging [13]	Not scalable
			ICMP Traceback [24]	May be used as a DoS attack
			IP Traceback [25], [26], [27]	Not complete, still evolving

- Detection, location and recovery from DDoS attacks. This is an active area of research, and IP traceback based identification approach is an evolving area.

The ultimate goal of Internet infrastructure security is to protect the Internet protocol suites against both known and unknown security attacks. This ambitious goal cannot be achieved in a single stroke as there are several intricacies associated with each attack, and also the vulnerability caused by an attack is protocol dependent. A pragmatic approach to solve this problem, is to develop secure versions of the protocols in an evolutionary manner as given below.

#### Repeat

- 1) Identify specific vulnerabilities and threats in the current implementation of the protocols.
- 2) Develop realistic threat models based on the threats analyzed.
- 3) Develop counter-measures based on the threat models developed. Counter-measures should aim at combating both known and unknown threats.
- 4) Carry out quantitative and qualitative evaluation of the counter-measures developed.

#### Until robust solution is achieved.

This approach will not only enable developing robust protocols, but also will provide significant insight into the nature of the security attacks leading to sustained development of better protocols.

#### REFERENCES

- [1] Kevin. J. Houle and George. M. Weaver, "Trends in Denial of Service Attack Technology," *CERT Advisory*, v1.0, Oct. 2001.
- [2] Charles. P. Pfleeger, "Security in Computing," *Prentice Hall*, 1996.
- [3] Report to Senator Robert F Bennett, "Information Sharing - Practices That Can Benefit Critical Infrastructure Protection," *GAO-02-24*, Oct. 2001.
- [4] Richard Clarke, "Looking at Vulnerability Issues in Cyber-Security," *Business Session of the President's National Security Telecommunications Advisory Committee (NSTAC)*, Mar. 2002.
- [5] Joint Economic Committee, "Security in the Information Age: New Challenges, New Strategies," *Report to United States Congress*, May 2002.
- [6] Computer Emergency Response Team, "Multiple Vulnerabilities in BIND," in *CERT Advisory* Nov. 1998.
- [7] F. Wang, F. Gong, F.S. Wu, and R. Narayan, "Intrusion Detection for Link State Routing Protocol Through Integrated Network Management," in *Proc. ICCCN*, 1999, pp. 694-699.
- [8] Bradley R. Smith, Shree Murthy, and J.J. Garcia-Luna-Aceves, "Securing Distance-Vector Routing Protocols," in *Proc. SNDSS*, Feb. 1997, pp. 85-92.
- [9] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee and R. A. Olsson, "Detecting Disruptive Routers: A Distributed Network Monitoring Approach," *Symp. on Security and Privacy*, May 1998, pp. 115-124.
- [10] Cisco White Papers, "Strategies to Protect against Distributed Denial of Service Attacks (DDoS)," Feb. 2000.
- [11] D. Eastlake, "Domain Name System Security Extensions," *RFC 2535*, Mar. 1999.
- [12] H. Burch and B. Cheswick, "Tracing anonymous packets to their approximate sources," *Proc. 2000 USENIX LISA Conf.*, Dec. 2000, 319-327.
- [13] G. Sager, "Security Fun with OCxmon and elflowd," *Internet2 Working Group Meeting*, Nov. 1998.
- [14] G. Malkin, "RIP Version 2," *RFC 2453*, Nov. 1998. *RFC 1058*, June 1988.
- [15] J. Moy, "OSPF Version 2," *RFC 1583*, March 1994.
- [16] Stephen Kent, Charles Lynn, and Karen Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE JSAC*, vol. 18, no. 4, Apr. 2000i, pp. 582-592.
- [17] K. Zhang, "Efficient Protocols for Signing Routing Messages," in *Proc. of SNDSS*, 1998.
- [18] L. Lamport, R. Shostak and M. Pease, "The Byzantine General's Problem," *ACM Trans. Prog. Languages and System*, vol. 4, no. 3, Apr. 1982, pp. 382-401.
- [19] Anirban Chakrabarti and G. Manimaran, "Secure Link State Routing Protocol," *Technical Report*, Dept. ECpE, Iowa State University, 2002.
- [20] Xiaobing Zhang, S. Felix Wu, Zhi Fu, and Tsung-Li Wu, "Malicious Packet Dropping: How It Might Impact the TCP Performance and How We can Detect It," *Symp. on Security and Privacy*, May 1998, pp. 263-272.
- [21] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," *RFC 2401*, Nov. 1998.
- [22] Chin-Tser Huang, and Mohammed G. Gouda, "An Anti-Replay Window Protocol with Controlled Shift," *Proc. ICCCN*, 2001, pp. 242-247.
- [23] Kihong Park and Heejo Lee, "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internet," *Proc. SIGCOMM*, Aug. 2001, pp. 15-26.
- [24] S. M. Bellovin, "ICMP Traceback Messages," *Internet Draft*, Mar. 2000.
- [25] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, "Network Support for IP Traceback," *IEEE Trans. on Networking*, vol. 1, no. 3, June 2001, pp. 226-237.
- [26] Dawn Xiaodong Song, and Adrian Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," *Proc. INFOCOM*, Apr. 2001, pp. 878-886.
- [27] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, T. Kent, and Timothy Strayer, "Hash-Based IP Traceback," *Proc. SIGCOMM*, Aug. 2001, pp. 3-14.