Securing User-Controlled Routing Infrastructures

Mr. Rabinarayan Panda
Asst. Professor, Dept. of MCA
The Oxford College of Engineering
Bommanhalli , Hosur Road , Bangalore-68
Email : rabi.mtech2011@gmail.com

***Abstract*—Design a WiFi network infrastructure untrusted third parties (intruders and malicious internal nodes) control over routing is a challenging task to research directions for achieving security, scalability, flexibility and efficient communication over WiFi network communication. However, the flexibility control plane can be exploited to launch many types of powerful attacks with little effort.**

In this paper, we make several contributions to overcome security issues on forwarding routing infrastructures. We propose a architectural model for a forwarding infrastructures, prevent potential security vulnerabilities, and address these vulnerabilities for future reference. The main technique that we introduce in this paper is use is simple, light-weight, cryptographic constraints on forwarding entries. We show that it is possible to prevent a large class of attacks on end-hosts, and bound the flooding attacks that can be launched on the infrastructure nodes to a small constant value. Our mechanisms are general and apply to a variety of earlier proposals such as NIRA, i3, Data Router and Network Pointers.

Keywords : WiFi Network, System Security,NIRA

I INTRODUCTION

Several recent proposals have argued for giving third-parties and end-users control over routing in the network infrastructure. Some examples of such routing architectures include TRIAD [2], i3 [15], NIRA [21], Data Router [16], and Network Pointers 34]. While exposing control over routing to third parties departs from conventional network architecture, these proposals have shown that such control significantly increases the flexibility and extensibility of these networks. Using such control, hosts can achieve many functions that are difficult to achieve in the Internet today, such as support for mobility, multicast, content routing, and service composition. Another somewhat surprising application is that such control can be used by hosts to protect themselves from packet-level denial of-service (DoS) attacks [18], since, at the extreme, these hosts can remove the forwarding state that malicious hosts use to forward packets to them. While each of these specific functions can be achieved using a specific mechanism—for example, mobile IP allows host mobility—we believe that these FIs provide architectural simplicity and uniformity in providing several functions that makes them worth exploring. Forwarding infrastructures typically provide user control by either allowing source-routing (such as [2], [15] or allowing users to insert forwarding state in the infrastructure (such as [15], [13]). Allowing forwarding entries enables functions like mobility and multicast that are hard to achieve Using source-routing alone. While there seems to be a general agreement over the potential benefits of user-controlled routing Architectures, the security vulnerabilities that they introduce has been one of the important concerns that has been not addressed fully. The flexibility that the FIs provide allows malicious entities to attack both the FI as well as hosts connected to the FI.For instance, consider i3 [15]an indirection-based FI which allows hosts to insert forwarding entries of the form (id,R), so that all packets addressed to id are forwarded to R. An attacker A can eavesdrop or subvert the traffic directed to a victim V by inserting a forwarding entry (idV ,A); the attacker can eavesdrop even when it does not have access to the physical links carrying the victim's traffic. Alternatively, consider an FI that provides multicast; an attacker can use such an FI to amplify a flooding attack by replicating a packet several times and directing all the replicas to a victim. These vulnerabilities should come as no surprise; in general, the greater the flexibility of the infrastructure, the harder it is to make it secure [1], [18]. In this paper, we aim to push the envelope of the security that truly flexible communication infrastructures, that provide adverse set of operations including packet replication, allow. Our main goal in this paper is to show that FIs are no more vulnerable than traditional

communication networks such as IP, which do not export control on forwarding. To this end, we present several mechanisms that make these FIs achieve certain specific security properties, yet retain the essential features and efficiency of the original design. Our main defense technique, which is based on light-weight cryptographic constraints on forwarding entries, prevents several attacks including eavesdropping, loops, and traffic amplification attacks. From earlier work, we leverage some techniques, such as challenge-responses and erasure coding techniques, to thwart other attacks.

The organization of the rest of the paper is as follows:

- To abstract away the details of the several forwarding infrastructures, we propose a simple model for FIs in Section II.
- We present the desirable security properties of a FI that can be roughly summarized as follows (Section IV): (a) an attacker should not be able to eavesdrop on the traffic to an arbitrary host, (b) an attacker should not be able to amplify its attack on end-hosts using the FI, (c) an attacker can only cause a small bounded attack on the FI, and (d) an attacker that has compromised an FI node can only affect traffic that it forwards. For each of these properties, we also present examples of attacks that show why a naive FI design violates these properties.
- We describe a set of security mechanisms that achieve these properties (Section V). The most important contribution, *light-weight cryptographic constraints on forwarding entries*, allows the construction of only acyclic topologies, thus preventing malicious hosts from using packet replication of the infrastructure to multiply flooding attacks. For example, to prevent loops, we leverage the difficulty in finding short loops in the mapping defined by cryptographic hash functions [22]. To the best of our knowledge, this is the first system that exploits the difficulty in finding short loops in cryptographic hash functions for designing a secure routing system

II. FORWARDING INFRASTRUCTURE MODEL

Since the designs of various FIs proposals vary greatly, we present a simplified model that abstracts the forwarding operations of these proposals. The following FI model we present is similar to MPLS [13]; in summary, the model tries to abstract the forwarding operation performed at an FI node to an update of the identifier that is contained in the packet header.

A. Identifiers and Forwarding Entries

Each packet header contains an identifier *id* that contains both the next-hop that the packet is addressed to (*id.node*), and a flat label used to match the routing table at the next-hop (*id.key*). The structure of *id.node* depends on the underlying routing used by the particular FI; for example, it could represent the IP address of the node (e.g. Data Router [16]), or the DHT identifier of the node (e.g. i3 [15]). When a host A wishes to communicate with host B using the FI, the host A sends a packet containing an identifier *id* that would eventually be routed to host B.

Each FI node maintains a table of *forwarding entries*. A forwarding entry is a pair $(id, E_k(k,data_i))$, where *id* has the same structure and semantics as the packet identifier and $E_k(k,data_i)$ (shorthand for encrypt the *forwarding information*) is additional information that is used to modify the header before forwarding the packet.

In the simplest case, the $E_k(k,data_i)$ is just the identifier to which the packet is next forwarded to, but it could also represent other types of forwarding information such as a source route or a stack of identifiers. The notion of $E_k(k,data_i))$ is introduced here just to show how we can accommodate several FIs;

The scope of the *key* of an identifier is local an FI node, and there may be several entries with the same key at a node to allow multicast. While precluding replication would eliminate several of the attacks that we discuss in this paper, we believe that multicast is a key functionality that future FIs will provide. These forwarding entries are maintained in the FI as soft-state that must be refreshed periodically.

B. Packet Routing Functions

The three steps in routing a packet are: (1) matching the packet header with forwarding entries at a node, (2) modifying the packet header based on the forwarding entry it matches, and (3) forwarding the packet to the next hop. Figure 1 illustrates the packet processing at an FI node.

Packet Matching. When a packet arrives at node, the packet identifier is matched against the the forwarding table by a matching function:

match (id, F) → {k1, k2, . . . , kn} (1)

Which takes as input a packet's *id* and a forwarding table F (stored at node *id.node*), and outputs a set of entries. For achieving our security properties, we'll later require that the matching operation matches a certain number of bits in the identifier exactly.

Packet Header Update. The header and destination of a packet are based only on the incoming packet's header and the matching entry. If multiple entries are matched, the packet is replicated. The update function

Update (p1(id,kn1,data1) → k1 (2)

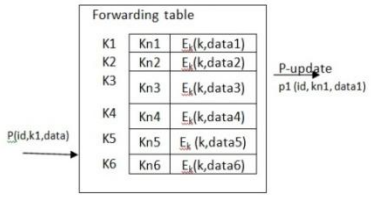

Figure 1. Forwording Routing Table

Where packet header p and entry e and produce a modified packet header kn1.

III. THREAT MODEL

We describe our assumptions and the attacker threat model, and then derive the attacks that can be launched.

A. Security Assumptions

Our main goal in this paper is to show that the FIs are no more vulnerable than traditional communication networks such as IP, which do not export control on forwarding. To achieve this goal, we rely on several assumptions about the underlying routing layer. We assume that the virtual links between FI nodes as well as the link between the end-hosts and the FI node it is connected to1 provide secrecy, authenticity, and replay protection—i.e., we do not consider link-level adversaries that can eavesdrop on arbitrary network links. These virtual links represent ISP-ISP relationships, which can be readily secured through standard security protocols (*e.g.*, IPsec [16]), and do not need a public-key infrastructure. FI proposals rely on an underlying routing protocol that routes packets between FI nodes. For example, Data Router uses IP routing, and i3 uses the Chord lookup protocol addressing security issues of these underlying protocols is outside the scope of this paper. We note that there are several ongoing research efforts to address security issues both in the context of IP routing [5], [6], [7] and DHT-routing [1], [15]. Finally, we do not consider processing or state-based attacks (such as insertion of many forwarding entries at an FI node) since these attacks are well-studied in the literature and can be solved using cryptographic puzzles [3], [4], [11].

B. Attacker Threat Model

We consider two attacker types: internal and external attackers. An *external attacker* does not control any compromised FI node but misuses the flexibility given by the FI. An external attacker can perform only the operations that a legitimate host can: insert a forwarding entry and send a packet. An *internal attacker* is an adversary who controls some compromised FI nodes. Ideally, we want to ensure that an external attacker cannot misuse an FI network to amplify the magnitude of a 1We assume that in real deployments, end-hosts are connected to one or a few FI nodes that act as the entry point of all packets of the hosts; hence, assuming that a host shares a key with a couple of FI nodes is reasonable. Flooding attack2. In the case of an internal attack, we want to ensure that an attacker who compromises an FI node cannot affect other traffic that is not forwarded through that compromised FI node

IV. PROPERTIES OF A SECURE FI

In this section, we precisely state the properties of a secure FI that we seek to achieve, and resent some simple examples of how these properties are violated in the naive FI designs.

A. Preventing Eavesdropping and Impersonation

Property 1: Let [id → X] be a public forwarding entry inserted by a host. Then, an external attacker cannot insert a forwarding entry with the same identifier id. This property prevents eavesdropping and impersonation by preventing an external attacker from inserting a forwarding entry with the same ID as that of the victim. The property also covers the case in which the victim has no entry in the FI at the time the attacker inserts its entry. Hence, even if the attacker causes the removal of the victim's entry (*e.g.*, by flooding the victim), it cannot impersonate the victim. To demonstrate that the basic FI design does not guarantee this property, we list an example each of an eavesdropping attack and an impersonation attack

Eavesdropping. Consider an end-host R that inserts a public forwarding entry3 [id→R] (see Figure 3(d)). An attacker X can eavesdrop on packets sent to R by inserting forwarding entry [id→X]. All packets that are forwarded via [id→R] will be replicated and forwarded via [id→X] to X as well

Impersonation. A variant of eavesdropping involves an attacker X making an end-host R drop its public entry by flooding it.4 then, if attacker X inserts [id→X], X can not only eavesdrop on R's traffic but also actively respond to it, thus impersonating R.

B. Preventing Flooding Attacks on End-Hosts

The following property prevents an external attacker from using the FI to: (a) amplify the traffic it sends to a victim host, and (b) redirect traffic meant for other hosts to the victim host.

Property 2: An external attacker cannot make a single victim end-host receive more packets than the attacker itself sends or receives.

In essense, the property bounds the worst-case flooding attack that an external attacker can perform to what the attacker can do in today's Internet: send packets directly to the victim. However, the basic design of FIs does not guarantee the property; we illustrate this using some intuitive examples.

Malicious linking. Consider a forwarding entry [id1→X] that receives a large number of packets. An attacker can sign up an end-host R, with an existing public forwarding entry [id→R], to the high bandwidth traffic stream of the popular entry by inserting the entry [id1→id].

Cycles involving end-hosts. Consider two benign hosts R1 and R2 inserting entries [id1→R1] and [id2→R2] respectively. An attacker can create a cycle by inserting entries [id1→id2] and [id2→id1]. Packets sent to id1 and id2 would be indefinitely replicated, thus overwhelming R1 and R2.

End-host confluence this is a variant of the confluence attack where the target is an end-host rather than an FI node. By making the leaves of the tree point to the public entry of an end-host (see Figure 3(c)), an attacker can overwhelm the host.

CONCLUSIONS

Giving hosts control over forwarding in the infrastructure has become one of the promising approaches in designing flexible network architectures. In this paper, we addressed the security concerns of these forwarding infrastructures. We presented a general FI model, analyzed potential security vulnerabilities and presented several mechanisms to alleviate attacks. Our key defense mechanism, based on lightweight cryptographic constraints, provably prevents a large set of attacks. In contrast to previous efforts that detect and mitigate malicious activity, the cryptographic mechanism prevents attacks altogether. Our mechanisms are applicable to many earlier proposals such as i3 [15] and Data Router [16] while requiring only modest changes. In providing secure forwarding, we make the deployment of these promising architectures much more viable.

REFERENCES

[1]. M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure Routing for Structured Peer-to-peer Overlay Networks," in *Proc. OSDI*, Dec. 2002.

[2]. D. R. Cheriton and M. Gritter, "TRIAD: A New Next Generation Internet Architecture," Mar. 2000, http://www-dsg.stanford.edu/triad/triad.ps.gz.

[3]. D. Dean and A. Stubblefield, "Using Client Puzzles to Protect TLS," in *Proc. of the 10th USENIX Security Symposium*, 2001.

[4]. C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," in *Advances in Cryptology — CRYPTO '92*, ser. LNCS, E. Brickell, Ed., vol. 740, International Association for Cryptologic Research. Springer-Verlag, 1993, pp. 139–147.

[5]. G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin, "Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing," in *Proc. of NDSS*, Feb. 2003.

[6]. S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE JSAC*, vol. 18, no. 4, pp. 582–592, Apr. 2000.

[7]. K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica, "Taming IP Packet Flooding Attacks," in *Proc. ACM HotNets-II*, Cambridge, MA, Nov. 2003.

[8]. K. Lenstra and E. R. Verheul, "Selecting Cryptographic Key Sizes," *Journal of Cryptology*, vol. 14, no. 4, pp. 255–293, 2001

[9]. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, ser. CRC Press series on discrete mathematics and its applications. CRC Press, 1997, iSBN 0-8493-8523-7.

[10]. R. Merkle, "Secure Communication Over Insecure Channels," vol. 21, no. 4, pp. 294–299, Apr. 1978.

[11]. G. O'Shea and M. Roe, "Child-proof authentication for MIPv6 (CAM)," *Computer Communication Review*, vol. Apr., no. 31, p. 2, 2001.

[12]. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001.

[13]. Secure Origin BGP (soBGP), ftp://ftp-eng.cisco.com/sobgp.

[14]. E. Sit and R. Morris, "Security Considerations for Peer-to-peer Distributed Hash Tables," in *Proc. of IPTPS*, 2002.

[15]. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet Indirection Infrastructure," in *Proc. SIGCOMM*, 2002.

[16]. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001.

[17]. Secure Origin BGP (soBGP), ftp://ftp-eng.cisco.com/sobgp.

[18]. E. Sit and R. Morris, "Security Considerations for Peer-to-peer Distributed Hash Tables," in *Proc. of IPTPS*, 2002.

[19]. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet Indirection Infrastructure," in *Proc. SIGCOMM*, 2002.