Brief Announcement: Towards a Secure Indirection Infrastructure

Karthik Lakshminarayanan UC Berkeley
Daniel Adkins UC Berkeley
Adrian Perrig CMU
Ion Stoica UC Berkeley

ABSTRACT

Designing a flexible, yet secure communication infrastructure has long been an elusive goal. Most of the proposals that seek to address the problem of flexibility have opened up the system for new forms of attacks. In this paper, we consider one particular proposal, *i3* [2], a flexible indirection infrastructure that provides natural support for a multitude of communication primitives such as multicast, anycast and mobility. We systematically identify the attacks on *i3*, and propose techniques that address the security problems without sacrificing the flexibility that *i3* offers. Our techniques, ranging from cryptographically constraining the forwarding entries to challenge-based mechanisms for inserting forwarding entries, while being simple, both conceptually and to implement, make most of the attacks provably hard. We believe that this paper represents an important step towards designing communication infrastructures that are both secure and flexible.

Categories and Subject Descriptors: C.2.0 [Computer-Communication Networks]: General - *Security and Protection*

General Terms: algorithms, security

Keywords: routing, security, cryptographic functions

1. ATTACKS

The Internet Indirection Infrastructure (*i3*) [2], is a flexible overlay that supports many communication primitives such as multicast, anycast, mobility and service composition. *i3* achieves this level of generality by allowing the end-hosts to insert *triggers*, or routing entries, in the *i3* infrastructure[1].

The generality of *i3*, however, comes at the cost of security. Figure 1 show some simple attacks that an attacker can launch on *i3*.

Eavesdropping and Impersonation. If the attacker inserts a trigger (id, X) when there is a trigger (id, X') in the infrastructure, the attacker can eavesdrop on all the traffic to id by simply setting X to its own address A (see Figure 1(a)).

Malicious Linking. An attacker can sign up end-host R to a high bandwidth traffic stream sent to id by inserting a trigger (id, R).

Cycles. An attacker may form a loop by inserting triggers $(id_1, id_2), (id_2, id_3), \ldots, (id_{n-1}, id_n), (id_n, id_1)$ (see Figure 1(b)). Packets sent to any of the IDs of the loop would indefinitely cycle around and consume infrastructure resources.

Confluence. An attacker can construct a confluence as shown in Figure 1(c). In the event of a confluence formation, packets are first replicated as they would be in a multicast tree. Then, instead of being delivered to separate end-hosts, all the replicated packets converge to eventually overwhelm an end-host via its public trigger.

Dead-ends. An attacker can construct a chain of triggers or a portion of a tree which does not ultimately point to a valid end-host (see Figure 1(d)). A data packet sent on such a topology would be routed through the chain of triggers only to be dropped when it reaches the dead-end. Such a packet will consume infrastructure resources without doing any useful work.

[1]The reader is assumed to be acquainted with the details of *i3* for the rest of this brief announcement.

PODC'04, July 25–28, 2004, St. John's, Newfoundland, Canada.
ACM 1-58113-802-4/04/0007.

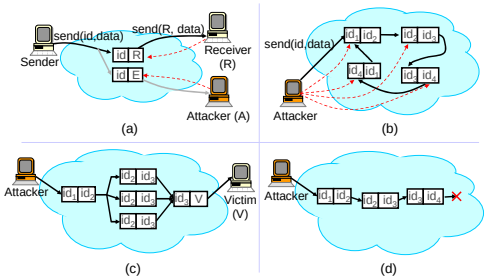

Figure 1: Types of attacks on *i3*.

2. DEFENSE MECHANISMS

The table in Figure 2 summarizes some of our defenses against these attacks. These defenses are described in detail in [1].

Attacks \ Defenses	Trigger Constraints	Pushback	Challenges	TTL
Eavesdropping Impersonation	✓			
Loops Confluences	✓			
Dead−ends		✓		
Node Confluence		✓	✓	
Long Chains				✓
Malicious Linking	✓		✓	
Malicious Trigger−Remove			✓	

Figure 2: Attacks and Defenses

Trigger constraints. We impose cryptographic constraints on the IDs of a trigger in such a manner that we can provably prevent topologies that are non-trees at the same time not compromising the functionality of *i3*. In particular, we divide a 256-bit ID into three fields: a 64-bit prefix, a 128-bit key, and a 64-bit suffix. Only triggers of the form (x, y), where either $y.key = h_r(x)$ or $x.key = h_l(y)$, are allowed in *i3*. h_l and h_r are well-known, one-way cryptographic functions mapping 256-bit strings to 128-bit strings. If a trigger (x, y) points to an end-host, we enforce the constraint $y.key = h_l'(x.key)$, where h_l' is an one-way cryptographic function mapping 128-bit strings to 128-bit strings, and use *y.prefix* and *y.suffix* to encode the end-host address.

Challenges. Nonce-based challenges are used to prevent gratuitous insertion of triggers by a malicious third-party.

Pushback. When a packet reaches an *i3* node where there is no matching trigger for the packet ID, the *i3* node sends back a pushback message to the node where the packet was previously matched. This would prune dead-ends in topologies.

Time-to-Live (TTL). Packets in *i3* are forwarded only for a lifetime of maximum k-hops where k is the TTL.

3. REFERENCES

[1] D. Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica. Towards a More Functional and Secure Network Infrastructure. Technical Report UCB/CSD-03-1242, UCB, 2003.

[2] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *Proc. SIGCOMM*, Aug. 2002.