Securing Information Transmission by Redundancy

Jun Li Peter Reiher Gerald Popek
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095

{lijun, reiher, popek}@cs.ucla.edu

Abstract

Many approaches have been used or proposed for providing security for information dissemination over networks, including encryption, authentication, and digital signatures. These mechanisms do not, however, necessarily help ensure that a message is delivered at all. Attacks that try to destroy or intercept security messages require other mechanisms. Authenticated acknowledgements are sometimes useful for this purpose, but do not scale well.

This paper discusses the use of redundancy to combat attempts to prevent information dissemination. Redundancy has been widely used in other areas, such as high availability data storage, file replication, and some fault-tolerant systems. The security problem has different characteristics that require different approaches to redundancy. We present one example of using redundancy to increase assurance of security updates delivery.

1 Introduction

More and more information is being shared and distributed over computer networks. Secure distribution of such information is becoming increasingly important. Conventional security approaches address many of the problems of secure information dissemination, but not all of them.

Encryption can provide secrecy, authentication can provide assurance of the source, digital signatures can provide integrity verification, firewalls can filter out dangerous transmissions, and so on. But these and other traditional mechanisms offer little assistance with interruption threats. No matter how elaborate the encryption or authentication, if the information is dropped on the floor, destroyed or transformed into a piece of garbage, blocked because of the overloading of an intermediate link, or disrupted by other malicious acts, information availability is damaged. In many cases, attackers can achieve their ends merely by ensuring that important information does not reach its destination, even if they cannot decrypt it, forge it, or alter it.

The traditional solution is to require acknowledgement of important messages. Since attackers might try to forge acknowledgements, they are typically signed (and possibly encrypted, if they contain sensitive information). If an acknowledgement is not received soon enough, the message is resent. This method works well if a relatively small number of messages require acknowledgement. If a very large number of messages must be acknowledged, then hierarchical or other load distribution methods must spread out the responsibility for checking the acknowledgements. In the general case, all nodes performing the checks must be trusted.

A further problem is that an attacker can repeatedly intercept or destroy the retransmitted message. Without other mechanisms, an attacker who has compromised a single link or router node may permanently prevent the delivery of the message, since each retransmission will probably still follow the same path through the compromised resource.

The problem is that there is only a single path for information transmission. If any point of this single path is corrupted, transmission security is corrupted. This problem can be reduced by adding redundancy to information transmission structures. Such redundancy can improve transmission resiliency and greatly improve the availability and other aspects of security. Typically such redundancy can be provided by using more than one path through the network to reach the destination.

If the redundant paths are completely disjoint, then attackers must compromise multiple resources in the network to prevent message delivery. The greater the degree of redundancy, the more resources they must compromise. Assuming that there is cost and risk in compromising each resource, increasing the degree of redundancy can thus increase the difficulty of preventing successful delivery.

Redundancy uses more resources than single-path transmission. Thus, there is a tradeoff between the degree of security achieved and the cost of providing it.

Similar arguments have demonstrated the value of redundancy for many hardware fault tolerance problems. In the networking realm, however, actually providing true redundancy may be difficult. While two distinct disks can be used for storing the same data, or two distinct processors can be loaded with the same instructions, it is not always true that two or more disjoint paths can be easily found for reaching a specific destination through a network. Such paths might not exist. Even if they do, existing network routing protocols and the desire to hide network complexities from higher levels make discovering and using the disjoint paths difficult. And it is even more difficult to know where those physical lines that a message follows are.

Nonetheless, redundancy can have some value. Even if the paths are not fully disjoint, any non-shared portions of the path limit an attacker's choice of attack points. The attacker must either find and compromise shared links or routers on the path, or must compromise the right set of non-shared elements. The volatility and obscurity that makes finding disjoint paths difficult also makes attacking them hard. While some choke points cannot be avoided, link-by-link (or segment-by-segment) redundancy may still prove very useful.

2 Interruption Threats Analysis and Transmission Primitives

2.1 Interruption threats

While information transmission latency has been shortened dramatically in the past few years, information transmission may still have to cut across several external entities or domains. A malicious entity might be able to penetrate into a place where everything seems under control. These external or maliciously penetrated places are where interruption threats will occur.

Interruption threats can be divided into two categories: path interruption and data interruption.

A path interruption happens when the information is dropped on the floor or misdirected to the wrong place. A path interruption also happens when some portion of the transmission path is flooded, causing denial of service. A data interruption happens when the information itself is damaged. Both types of interruption can happen even if the malicious entity does not know what is inside the information, so encryption or endorsement of the information cannot help. Interruption threats can be more serious if combined with other kinds of security attacks.

2.2 Transmission primitives vs. security assurance

Some transmission primitives have addressed the difficulties in transmitting information. While they are designed mainly for non-security reasons (particularly reliability in the sense of no data loss or physical error) and they do not provide a total solution, they do provide some assistance in coping with security problems in information transmission.

2.2.1 Reliable transmission – TCP

TCP [15] provides reliable one-to-one information transmission on top of the IP layer, at which an IP packet is routed to the destination along a dynamically determined physical path. If a TCP packet is lost according to acknowledgement information from the receiver, or if its own retransmission timer times out, a TCP sender retransmits the TCP packet.

If interruption attacks are sporadic, causing TCP to drop an occasional packet or sometimes damage the data, a TCP retransmission can heal the problem. But essentially TCP cannot eliminate the interruption threats if the retransmitted TCP packets encapsulated in IP packets will go through the same hostile point and be maliciously manipulated again.

In the reverse direction, the acknowledgement will also be subject to interruption threats, for instance when the one-to-one connection is symmetric and the reverse routes will also pass through the same hostile point. The acknowledgement, even with signature or other security enhancement, will not be able to reach the TCP sender smoothly.

A packet blocked by path interruption threats will cause further retransmission, since the sender may get a report that the packet is missing. Dropping or damaging of acknowledgement will also cause retransmission, since the retransmission timer at the sender side will time out before hearing any acknowledgement.

2.2.2 Reliable multicast

Reliable multicast provides reliability for information multicast. Usually it is done by negative acknowledgements or repair requests. As one example, SRM (Scalable Reliable Multicast) [5] lets each multicast recipient be responsible for information loss or error by requesting a repair from the whole multicast group (not necessarily from the sender) or by initiating a local recovery. Since more than one recipient may initiate a retransmission, the philosophy of SRM is to suppress repeated repair requests or repair itself to let only one copy be received to avoid explosion and implosion. This approach is subject to interruption threats, although it is definitely correct in normal information transmission, providing good reliability while saving bandwidth.

2.2.3 Broadcasting and flooding

Broadcasting and flooding are used to reach multiple destinations with a best effort in just one session transmission. A recipient may receive more than one copy of exactly the same information, which inadvertently gives rise to some level of redundancy (perhaps not enough) by heavy use of bandwidth. Standard broadcast and flooding methods assume that all members are following the rules, and that delivery on a link level is assured.

Reliable broadcast [3] has been proposed to deal with information loss or error caused by non-security problems, such as physical transmission errors. Obviously it cannot eliminate interruption threats for the same reason as TCP.

In a word, conventional approaches of information transmission can provide good reliability in terms of "natural" information loss or error, but provide little support in counteracting "artificial" information loss or damage. To deal with these interruption threats, we need some new approach to transmitting information in a secure fashion.

3 Redundancy for Security Assurance

We propose that redundancy in information transmission is valuable in providing security assurance. Redundancy here means that the information source, the information transmission path, or part of the path, is multiplied to avoid a single point of security corruption.

Massive redundancy in a small-scale environment may be employed to achieve best resiliency. However, lean but resilient redundancy is the fundamental goal, since brute-force redundancy will result in uncontrolled waste of resources in a large-scale environment, which in turn may overload some resources to cause denial of service.

We believe redundancy is important for security assurance in large-scale networks like the Internet.

3.1 Redundancy in other areas

Redundancy has been widely used in many areas by devoting more resources to achieve better availability. Resources redundancy is often applied to include multiple processes, multiple hardware components, and multiple data copies, usually with independent failure probabilities. Examples include high availability data storage, file replication, data backup, fault-tolerant distributed systems, mapping one web site to multiple IP addresses, and so on.

In high availability data storage, either more than one disk stores a copy of the data or the data is dispersed to more than one disk with built-in redundancy to deal with disk crashes, balance load from a hotspot disk, and provide lower latency for data access. This is normally transparent to users [13].

File replication has been used to make replicas to support easier access [9] [11] [17]. Establishing mirror web sites for lower latency is one such example.

Data backup, usually done periodically, can help restore damaged or lost files from backed-up copies.

In a fault-tolerant distributed system, replicated execution [18] may be employed to run a program concurrently at multiple places. The computation can still smoothly continue if one execution succeeds.

Mapping one web site to several different server machines, in a round robin fashion or other more sophisticated way, can prevent one single server from being overloaded and ensure that the site is accessible even if some server machines have crashed [21].

3.2 Resiliency evaluation

Given a graph with fault probability distribution of nodes, computation of the probability that there is a non-faulty path between two arbitrary nodes is known to be NP-hard in the worst case. But we still can look at some resiliency properties of a graph to get some basic understanding of what redundancy structures are good.

Let us define the resiliency of a one-to-one connection as the probability that the source S can *reach* destination D, denoted as $R_{S\text{-}D}$. Here, the word "reach" here means that, given a message, when every path from S to D is used to transmit a copy of the message at the same time, at least one authentic copy can be received. Further assume for this specific connection that there are m cutsets C_1, C_2, C_3… C_m, each containing some number of elements (a single element cutset corresponds to a choke point, for instance). Denote E_i (i=1, …, m) as the event that at least one element of C_i is not broken, then

$$R_{S\text{-}D} = \mathit{Probability}\ (E_1 \text{ and } E_2 \ldots \text{and } E_m)$$

Usually decreasing the number of cutsets, here m, can increase the resiliency of a connection. Further analysis can also show that higher resiliency can result if a cutset contains more elements, or an element has a lower probability of being subverted.

Having each path be as strong as possible by passing through the least number of corrupted nodes can decrease the number of cutsets; and having more paths, in particular as disjoint as possible, to reach a destination can make a cutset of the connection contain more elements, thus strengthening the resiliency of the connection in general.

3.3 Using redundancy in transmission

Redundancy may be improved by simply increasing the number of the sources of same information or the number of transmission paths, particularly when information corruption is detected. The increase can be linear or exponential or by other degrees.

This may not provide extra security assurance in information transmission, however, and may lead to unwise resource usage and degraded performance. For instance, if the incoming link for a receiver is maliciously flooded causing denial of service, contacting more sites for redundant information may not bring in any useful message; it may instead cause more severe overloading of the link.

To achieve best assurance with consideration of other factors, a sophisticated redundancy design is necessary. The designer should understand the stochastic distribution of interruption threats, make the best trade-off between resource usage and redundancy, build resource-saving but resilient transmission structures, use an adaptive algorithm to help choose when and how to deploy redundancy, and so on. In the above case, for instance, a receiver may also run an intrusion detection facility to find the reason for continuing information unavailability.

There are many complex issues in deploying redundancy in large-scale networks like the Internet. One problem is that each machine in the Internet is heterogeneous in terms of transmission characteristics, platform, security situations and requirements. Ideally, some of this information should be taken into account when choosing redundant paths. For example, if a particular node is suspected to be highly insecure, special care should be taken to avoid routing multiple supposedly disjoint paths through that node. Also, the security system must be adaptive in dealing with a dynamic environment in terms of location, transmission mechanism, and impact of interruption threats.

The complexity also lies in the fact that a compromised element can further compromise other intermediate elements or cause them to behave in a wrong way. For instance, while misbehaving on data traffic itself, a compromised router may cause other routers to unknowingly misbehave by sending them false routing messages [19]. Building security into the routing infrastructure is itself a challenging task [4] [8] [20] [22]. Unless routing infrastructure security is strong, two paths used to reach a destination should not only be as disjoint as possible, but also isolated within the routing infrastructure. For instance, using routers belonging to different ISPs would be preferable.

For now we will ignore the fact that two independent connections for redundancy may possibly be carried by the same cable that might be subject to physical attack.
As we pointed out earlier, resilient but lean redundancy is what we want. Obviously, in a large-scale network such as the Internet, building such a structure can only be done in a distributed fashion, adding further difficulty.

4 A System Disseminating Security Updates – Revere

4.1 Overview

Revere [10] is a system designed to disseminate security updates over the Internet to a large number of machines. The security updates can contain a new virus signature and/or its remedy, special events in a distributed intrusion detection system, offending characteristics to be filtered by a firewall, characteristics of a potential attack, certificate revocation lists, and so on. Revere has special characteristics not common to all uses of redundancy for security.

Security updates are usually of small size, disseminated infrequently, and in particular of vital importance. It is acceptable for a node to receive multiple copies of a security update. Delivering updates reasonably rapidly is important. Perfect delivery to all nodes is often not vital, but the message should be delivered to a high percentage of all nodes, and every individual node should have a high probability of receiving any given message. In particular, it should be difficult for attackers to selectively cut off particular nodes.

The overall problem Revere seeks to solve has many challenging aspects. We focus here on those that can be addressed by use of redundancy.

4.2 Revere dissemination structure

A security update structure is designed to disseminate updates in a hostile environment. Single-path dissemination or feedback is already subject to interruption threats, and the high scale and lack of trust in the vast majority of participants makes verification of positive or negative acknowledgements from the receivers even harder. The originator of the update probably cannot handle millions of acknowledgements, especially if doing so requires cryptographic authentication. Further, an acknowledgement approach would require that all participants have trusted keys, leading to a huge key distribution and management problem. Also, a receiver won't send a negative acknowledgement if it does not know that it *should* have received an update. Instead, we propose to build redundancy into the dissemination structure. Each entity interested in receiving security updates can choose to attach itself to the structure and hear multiple copies of security updates to achieve security assurance.

Furthermore, a joining node may also act as a *security update router* to forward security updates to other neighboring nodes. This characteristic assists the scaling of actual update dissemination, but since joining nodes cannot be fully trusted, it also offers another point of attack. Redundancy, however, also counters this problem.

Each Revere node has a sending table specifying a list of its children with corresponding transmission characteristics. The Revere node is likely to have multiple parent nodes that send it information.

4.3 Structure formation

There are two aspects of the formation of the dissemination structure:

4.3.1 A new applicant joins the existing structure

The applicant first contacts one or more existing Revere nodes based on out-of-band knowledge. It then either attaches itself as a child of one or more contacted nodes, or grabs the children from each contacted node, and chooses the "best" children to make further contact in a recursive fashion. The applicant runs an algorithm to find those nodes giving the best efficiency, and those children that can offer the best resiliency. Finally, the applicant will become a leaf node of the structure with multiple parents.

The most challenging issue here is to figure out which nodes are good candidates for resiliency reasons from given candidates. Some techniques [1] [6] [7] [23] can be considered to get the topology information for this purpose. In current ongoing design, each Revere node maintains its own several path vectors relative to the dissemination source, each of which contains latency information and a vector of peer Revere nodes to pass through to reach itself, as well as other information. By evaluating path vectors of every candidate, the applicant can determine which of the several candidates can collectively provide best resiliency for itself and which one is most efficient. As discussed in resiliency evaluation, the path vectors with a less overlapping degree to each other and shorter lengths are preferable (assuming these characteristics of path vector can map to the physical routes rather well).

4.3.2 Maintenance of the structure

The dissemination structure has to be maintained to accommodate changes. Heartbeat messages are used to report aliveness and to refresh the information kept at each Revere node. If necessary, the heartbeat will trigger some level of reorganization of the structure.

4.4 More on redundancy of Revere

Building a structure doing redundant push of security updates obviously cannot guarantee complete information availability. If there is only a one-time best-effort transmission, any nodes disconnected from the network during the transmission will miss the information. Also, any nodes that attackers temporarily disconnect will have no way to obtain the update later. Revere will also contain a pulling mechanism to handle such cases.

Some repository nodes will keep security updates that were disseminated in the past and provide missing security updates if queried. Redundancy also has a role here: a node can contact more than one repository node to retrieve the missed information. The node will receive the most benefit if the repository nodes it contacts will return their results along the most disjoint paths possible, within the necessary constraints of providing good performance, such as low latency.

5 Related Work

Redundancy for fault-tolerant information transmission has been studied by many people [2] [14]. Dealing with Byzantine faults has also been considered. This research has only focused on specially structured networks, such as broadcasting over complete networks or hypercube. Also, a system may tolerate Byzantine faults, but not malicious faults.

Another related research area is information dispersal [16]. It has some similarities to the RAID technology for data storage. The original information is divided into pieces with some level of redundancy before being transmitted separately. After a receiver gets the pieces, it can assemble them into the original information, even if some pieces are lost or damaged.

5 Future Research

Using redundancy to secure information transmission still has many open issues, especially when used in large-scale networks such as the Internet.

One problem is that we do not completely understand large-scale redundancy. Since centrally building a good redundant distribution structure is not feasible, we need to use distributed algorithms to build the structure on the fly. The proper methods of doing so to maximize overall resiliency are unclear. One problem Revere needs to solve is scalability with large numbers of dissemination sources.

A second problem is the security of the distributed procedure for building redundant structures. If the redundancy mechanism is compromised, the supposedly beneficial system could actually work against security. Some problems in this area and their solutions are obvious, but more subtle and indirect problems are likely to occur.

A third problem is further theoretical understanding of redundancy for better security assurance, such as how to evaluate the resiliency of a whole dissemination structure besides the resiliency of a one-to-one connection.

6 Conclusion

In this paper we discussed usage of redundancy for the purpose of security assurance in information transmission. Our analysis shows that both conventional transmission primitives and frequently used security techniques are not adequate when counteracting interruption threats. Redundancy, a widely used approach in other areas, can also improve security of information transmission. Revere, a real system performing dissemination of security updates, was discussed to illustrate the advantages of redundancy and the difficult problems in providing it. We believe redundancy has wider applicability in many areas of network security.

References

[1] J.Case, M.Fedor, M.Schoffstall and J.Davin. A Simple Network Management Protocol (SNMP), RFC 1157, May 1990.

[2] M.Castro and B.Liskov. Practical Byzantine Fault Tolerance, in Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, February 1999.

[3] J.M.Chang and N.F.Maxemchuck. Reliable Broadcast Protocols, ACM Transactions on Computing Systems, August 1984.

[4] S.Cheung and K.N.Levitt. Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection, in Proceedings of New Security Paradigms Workshop, Langdale, Cumbria, UK, 1997.

[5] S.Floyd, V.Jacobson, S.McCanne, C.G.Liu, and L.Zhang. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. In Proceedings of SIGCOMM '95, Boston, MA, Sept. 1995, ACM.

[6] P.Francis, S.Jamin, V.Paxson, L.Zhang, D.Gryniewicz, Y.Jin. An Architecture for a Global Internet Host Distance Estimation Service, in Proc. IEEE INFOCOM '99, March 1999.

[7] V.Jacobson. *traceroute*, ftp://ftp.ee.lbl.gov/traceroute.tar.Z, 1989.

[8] Y.F.Jou, F.Gong, C.Sargor, S.F.Wu, R.Cleaveland. Architecture Design of a Scalable Intrusion Detection System for the Emerging Network Infrastructure, MCNC, Technical Report CDRL A005, April 1997. http:// shang.csc.ncsu.edu//papers/jinaoArch.ps.gz.

[9] G.H.Kuenning and G.J.Popek. Automated Hoarding for Mobile Computers, in Proceeding of the 16th ACM Symposium on Operating Systems Principles, (SOSP-16) St.Malo, France, October 5-8, 1997.

[10] J.Li. Dissemination of Security Updates – Revere. http://fmg-www.cs.ucla.edu/lijun/revere.

[11] J.McDermott. Replication does survive information warfare attacks, in Proc. of the 11th Annual Working Conference on Database Security, Lake Tahoe, CA, August 1997, pp. 186-198.

[12] I.S.Moskowitz and M.H.Kang. An Insecurity Flow Model, in Proc. of New Security Paradigms Workshop, Cumbria, UK, September 1997.

[13] D.Patterson, G.Gibson, and R.Katz. A case for redundant arrays of inexpensive disks (RAID), *Proceedings of IEEE COMPCON*, pp. 112~117, Spring 1989.

[14] A.Pelc. Fault-Tolerant Broadcasting and Gossiping in Communication Networks, NETWORKS, Vol. 28 (1996) 143-156.

[15] J.Postel. Transmission Control Protocol, RFC 0793, September 1981.

[16] M.O.Rabin. Efficient dispersal of information for security, load balancing and fault tolerance. JACM 36(2) (1989) 335-348.

[17] P.Reiher. G.Popek, M.Gunter, J.Salomone, and D.Ratner, Peer-to-Peer Reconciliation-Based Replication for Mobile Computers, Proceedings of the ACM/ECOOP Workshop on Mobility and Replication, July 1996.

[18] M.Singhal, N.G.Shivaratri. Advanced Concepts in Operating Systems, Distributed, Database, and Multiprocessor Operating Systems, McGraw-Hill, Inc., 1994

[19] F.Wang, B.Vetter, S.F.Wu. Secure Routing Protocols: Theory and Practice, May 1997. http://shang.csc.ncsu.edu//papers/CCR-SecureRP2.ps.gz.

[20] S.F.Wu and C.Sargor. Deciduous: Decentralized Source Identification for Network-based Intrusions, in DARPA/ITO Next Generation Internet PI Conference, October 1998. http://shang.csc.ncsu.edu/deciduous/.

[21] *White Paper*. Enhancing Web User Experience With Global Server Load Balancing, Alteon Networks. http://www.alteon.com/products/white_papers/GSLB/index.html.

[22] http://www.net-tech.bbn.com/sbgp/sbgp-index.html. Secure BGP Project (S-BGP).

[23] http://govt.argreenhouse.com/felix/. Project Felix: Independent Monitoring for Network Survivability.