Simple Firewall Traversal Mechanisms and Their Pitfalls
draft-lear-callhome-description-03.txt

Status of this Memo

Copyright Notice

Abstract

Many devices make use of so-called "Call Home" functionality in order to be managed or updated, or to otherwise establish outbound communication in the face of NATs, firewalls, and mobility. This memo defines call home functionality, discusses the requirement for firewall traversal, some mechanisms used, and security considerations of those mechanisms. Several existing examples will be shown. This memo also contains examples of how one would make SNMP over SSH, NETCONF over SSH, and interactive terminal access call-home protocols.

1. Introduction

In the early days of the networking it was recognized that some devices would be intermittently reachable. Mechanisms such as UUCP [1] were based on this notion, and support for systems requesting that the server act as the client showed up in the Internet no later than 1982 in SMTP [2] and were formalized in Blocks Extensible eXchange Protocol (BEEP) [3] in 2001.

However, in the early days of the Internet it also largely didn't matter from a network security or transparency standpoint which device initiated communication, because there was little if any network security and everyone used public address space. With the introduction of private address space [4] and firewalls the world changed. Today a firewall with network address translator (NAT) functionality is a consumer device, not to mention an interdepartmental device.

In addition, the complexity of IT relationships and the number of vendors that support enterprises has changed the underlying assumption that the enterprise actually manages its own network and support devices, such as power distribution units. Often for small businesses, today, the situation is reversed and it is the small business that has limited access to even the network layer of their data center service provider.

All of this leads us to the conclusion that a flexible means for management applications to traverse firewalls is a useful approach in the face of devices that intercept unacknowledged SYNs or keep translation tables based on connection state.

2. What is Call Home and what is it good for?

"Call Home" refers simply to the notion of reversing the party that traditionally initiates a communication. An early example of Call Home includes the SMTP "TURN" command where the SMTP server becomes the client and the client becomes the server Various system management protocols such as Track [5][6] have offered similar functionality for quite some time. Most PCs have some means to update their operating systems and virus definitions via a similar mechanism.

Call Home is useful for devices that do not retain a stable accessible point within a network. For instance, a lap top or a wireless phone may move from one location to another, and yet it still is be desirable for that device to be managed when it is online. Imagine what would be necessary in order to manage such a

device by having the manager contact it:

1. Either the DNS would have to be updated with the mobile devices new address or the device would have to make use of MOBILE-IP [8];
2. The device would have to remain in either the global address space or within the same address space as the manager;
3. Because firewalls often only allow communications one way without prior arrangement (if they have the capability at all), they would have to be informed of the device's new location and that the device is authorized to receive requests.

Call Home also allows for more complex management relationships without the need of complex VPNs and access lists. If an enterprise wished to make use of a contract service for printer maintenance, that service could monitor printers via the MIB defined in [7]. The same scenario could be envisioned for management of uninterrupted power supplies (UPS) via [9]. In either case the vendor has little need of general remote network access, and the enterprise has a desire to limit such access.

3. How is Call Home achieved?

Call Home already exists in those session-based unicast protocols where the allowed operations and responses do not differ based on who initiated the connection. An example in the routing world would be BGP. Once the connection is established each side authenticates to the other and the same protocol operations may be executed by either end. In the application world, so-called "peer to peer" protocols that are used for (often illicit) file transfer also fit this description.

Often, however, protocols are designed with client and server roles. Examples include SMTP, and NNTP. In these cases, some additional support within the application is necessary. In SMTP's case the TURN and ETRN capabilities provide a means for ends to switch roles of client and server. In NNTP a separate mechanism to retrieve articles - NEWNEWS - allows transfer agents to retrieve articles in a similar (albeit not identical) way the IHAVE operation and a queue of messages.

The applicability of Call Home in circumstances other than those above is extremely limited. For instance, protocols that are based on atomic transactions, such as DNS queries, have no need to reverse client and server roles. Indeed one would wonder of the intent of a name server that attempted to require a client to make a query of it. Similarly, the notion of Call Home in a multicast environment is likely limited as well as it is not clear who would reverse roles.

Because TCP state is easily detected in the header via the ACK bit, call home is also most easily implemented in TCP. Because connection state is not as easily discerned for protocols based on UDP as specifics of each protocol would need to be known and the communication unencrypted, firewalls may be more reticent to pass UDP traffic and simple NAT mapping timeouts may require contrived or dummy transactions to retain the mapping, but the same principle would apply. Hence the usefulness of Call Home in a UDP environment may be limited.

4. How does Call Home change the nature of the communication?

There are several differences between the traditional connection approach and Call Home. In the traditional case of a manager and an agent, the manager would make a request of the agent at any point when the manager wishes. In the case of Call Home, the manager must wait at least until the agent has established a transport connection. This also means that control of connection frequency passes from the manager to the agent. If frequency is important either the behavior must be codified somehow or the manager must pass these parameters to the agent and the agent must use them.

A change of who is listening for new connections in the cases of TCP or SCTP further means that a potential DDOS target passes from the agent to the manager.

In the traditional case, a manager may use any local TCP or UDP port to initiate a connection but must connect to the agent on a well known (or at least prearranged) port. In the call home case, again the roles are reversed, and it is the manager that must service requests on a well known port.

In the traditional case, each agent has a stable well known address, just as it has a well known port. In the case of Call Home, the manager must maintain a stable well known address.

5. Naming Issues

One reason to make use of Call Home is that traditional names, such as domain names may not be useful to contact a device, particularly if its IP address changes, either because the device has moved or because it leases addresses from a pool. While it is possible to make use of DNS in such circumstances through mechanisms such as dynamic update [10], such use requires that tight coupling between the subsystem invoked via Call Home and the DNS, and is not particularly meaningful when the connecting device resides behind a

NAT or a firewall.

When implementing Call Home there are several possibilities for choice of naming system. In some cases, no naming system may be needed. In others, as may be the case with a consumer DSL or cable deployment, the customer username may be sufficient. In other cases, domain names may be suitable. In all cases where names are used the security of the binding between the name and the device or application must be considered.

6. Security Considerations

The nature of security of the communication is likely to change. While there are many aspects of this problem, the common traditional case requires that the agent somehow authenticate its host address or domain name (either via X.509 [11] certificate or SSH host key) and the manager authenticates via public key or username and password. Once again, with Call Home these roles are reversed: the manager authenticates its host address or domain name and the agent authenticates via public key or username and password.

Some applications might require some additional configuration, therefore, in order to accommodate Call Home. For instance, SNMP requires that the command generator be associated with a SecurityName. If the agent initiates the connection, either it must derive the security name from something like the host key or subject in the certificate of a manager, or it must be pre-configured with a username to associate the connection.

6.1. Threat / Trust Model Changes

In a more traditional client server relationship, the client connects to request some service of the server (thus the terminology). In a way, that does not change with Call Home, because in this case the client is requesting to be managed or is requesting that roles be reversed. The server must still authorize this request.

However, there are changes from the traditional model. For instance, if the client is asking to be managed, the nature of attacks change to that of mechanisms such as dictionary attacks on a request port to approaches that trick the client into connecting to a bogus management server where bogus requests could be generated. This can actually have some benefits to security by limiting dictionary and buffer overflow attacks, to centralized well protected points, provided that the communication initiated by the client is well protected with such mechanisms as SSL, TLS, and the like; and the client itself does not listen for requests.

Finally, the underlying application that makes use of Call Home will have to consider the sort of information that is being made available as a service. Each application will have different sorts of threats and mitigations. For instance, the author knows of no SMTP agent that implements TURN because while most SMTP users are comfortable with the risks of Man in the Middle attacks associated with masquerading as SMTP servers, the risk of someone masquerading as a client are considered unacceptable.

In all cases, strong authentication of either end of the communication is recommended for exchange of sensitive information, regardless of who started it.

6.2. Firewall Administration

As we discuss elsewhere in this document Call Home reverses use of well known ports and services. It is important for Call Home protocols to make use of well known ports in order to respect the legitimate wishes of firewall administrators. Such use makes more reasonable the assumption that a port is blocked for a reason. A firewall administrator may wish to allow certain communications in a single direction. Use of additional well known ports may be advised in certain circumstances. However, the ability of devices and protocols to call home exists today through SSL connections, to give but one example. Excessive barriers to inclusion of call home functionality in protocols risks inappropriate use of existing substrates.

7. Example 1: NETCONF using SSH

NETCONF [12] is a fairly simple client/server protocol. NETCONF is mapped to several protocols, including SSH.[13] In order for NETCONF agents to call home some protocol operation must be passed to the manager for this purpose, and this operation can occur in the protocol mapping layer. Thus, the simplest approach would be to have a new SSH subsystem called "netconf-turn". When the SSH client invokes this subsystem, the SSH server either will initiate the the subsystem and proceed with NETCONF capabilities exchange from the point of view of a manager or refuse to initiate the subsystem.

The nature of the NETCONF communication changes in that the manager must wait for the agent to connect, as mentioned above. There are no events explicitly defined in NETCONF at this time and so there are no explicit functions that require deferral from a protocol standpoint. However, the manager cannot configure the agent until it connects and so completion of a configuration request may be deferred when a manager is not in communication with an agent. The manager must

retain configuration requests and higher level application must be able to deal with such deferrals.

From an authentication standpoint, the SSH server must determine whether based on the credentials given the client has appropriate access to be managed. Each NETCONF management operation on the SSH server must be governed by those credentials.

On the client, it would be a configuration error for it to invoke the netconf-turn subsystem on the manager and then not allow ANY operations, but each operation must be authorized based on the server identity passed up by the SSH subsystem.

8. Example 2: SNMP over SSH

Let us again first discuss the nature of the communication. In the case of SNMP there are ostensibly two basic protocol operations - request and response. While in theory either entity may make such requests in practice only one end issues GET, SET, or GET-BULK operations while the other end issues notifications.

SNMP does not specify when GET, SET, and GET-BULK are to be executed, as these choices are left to the application or the user. Therefore, the analysis given for NETCONF regarding deferral is just as applicable to SNMP. However, in the case of notifications, SNMP does specify when these occur based on the MIB definitions. Had the designers of SNMP version 3 not allowed for the SNMP-TARGET-MIB, a change to the protocol base would have been required. But because such a MIB exists, all that remains is how it should be configured. There are two cases:

- o It is desired that no events be deferred and the agent connect to the manager, just as would be the case in RFC 3430. In this case, the SNMP-TARGET-MIB is configured externally to use (presumably) the SSHSM security model to contact the manager when a notification is to be sent. The SSHSM will define initial connection semantics.
- o It is desired that notifications be deferred until the manager contacts the agent. Here once the SSHSM subsystem is invoked by the manager, a policy is triggered to configure the SNMP-TARGET-MIB to receive events appropriate to the manager.

The following is speculative as work on [14] is not complete. That document specifies a means to extend the SNMP protocol to use SSH. SSH establishes a session and will to SNMP via SSHSM a securityName that may be used for purposes of authorization. Once established the connection may be used for any purpose, no matter the original purpose in a vein similar to that specified by RFC 3430 [15] provided

each end is properly authorized. Once again, it would be a configuration error for a device to connect for the purposes of being monitored or configured by a manager to not accept any operations. It would similarly be a configuration error for a device to connect for purposes of sending notifications but then not have any possibly allowed.

9. Example 3: Remote terminal access via Call Home SSH

Consider the case of a device that is managed by administration that resides on the other (public, if you will) side of a firewall. When the device starts it initiates an SSH connection, perhaps with the intent of starting a netconf or SSHSM session. When a problem arises, however, the administrator may want interactive access to the device to debug the problem. The administrator makes use of a tool on the network management station that causes the NMS to request a "session"connection, thus allowing the administrator interactive command line access even through the device initiated the connection.

Section 6.1 of [16] rightly encourages client implementations to reject such requests. However, if they are prepared to trust the device they are connecting to for maintenance and debugging purposes, the benefit may outweigh the risks. In all cases, the session should be properly authorized, meaning that the agent should be configured to allow appropriate access to those who have appropriate access to the NMS, and the NMS should properly authenticate and authorize that access.

In this example, a naming method must be employed at the very least by the NMS in order to properly identify the correct device to connect to.

10. IANA Considerations

While much of this is protocol specific it is within the realm of possibilities that with client/server protocols either a new port or an SSH service name or a BEEP URN will be needed to indicate the intent of the initiator of communication to "turn" it.

11. Summary

Call Home is a useful - and in some circumstances necessary - firewall and NAT traversal approach applications can use to augment their existing approach in order to establish communications with devices that sit behind NATs or firewalls, or otherwise have

intermittent connectivity.

12. Informational References

[1] Nowitz, D., Lesk, M., and G. Chesson, "A Dial-8p Network of UNIX Systems", UNIX System 7 , August 1978.

[2] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[3] Rose, M., "The Blocks Extensible Exchange Protocol Core", RFC 3080, March 2001.

[4] Rekhter, Y., Moskowitz, R., Karrenberg, D., and G. de Groot, "Address Allocation for Private Internets", RFC 1597, March 1994.

[5] Nachbar, D., "When Network File Systems Aren't Enough: Automatic Software Distribution Revisited", Proceedings of Usenix Summer 1986 , June 1986.

[6] Pleasant, M. and E. Lear, "Transcending Administrative domains by Automating System Management Tasks in a Large Heterogeneous Environment", Usenix Software Security Workshop , April 1989.

[7] Bergman, R., Lewis, H., and I. McDonald, "Printer MIB v2", RFC 3805, June 2004.

[8] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[9] Case, J., "UPS Management Information Base", RFC 1628, May 1994.

[10] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

[11] International International Telephone and Telegraph Consultative Committee, "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework", CCITT Recommendation X.509, November 1988.

[12] Enns, R., "NETCONF Configuration Protocol", draft-ietf-netconf-prot-08 (work in progress), September 2005.

[13] Wasserman, M. and T. Goddard, "Using the NETCONF Configuration Protocol over Secure Shell (SSH)", draft-ietf-netconf-ssh-04 (work in progress), April 2005.

[14] Harrington, D., "Secure Shell Security Model for SNMP", draft-harrington-isms-secshell-01 (work in progress), September 2005.

[15] Schoenwaelder, J., "Simple Network Management Protocol Over Transmission Control Protocol Transport Mapping", RFC 3430, December 2002.

[16] Lonvick, C. and T. Ylonen, "SSH Connection Protocol", draft-ietf-secsh-connect-25 (work in progress), March 2005.

Appendix A. Changes

- From -02 to -03: Added naming and interactive terminal sections.
- From -01 to -02: reworded limitations of UDP and call home. Expanded security considerations. Spell-checked.
- From -00 to -01: provided more detail on Call Home applicability in the cases of unicast session based versus other. Discussed the difference between p2p protocols versus client server. Provided more examples.

Author's Address

Eliot Lear
Cisco Systems GmbH
Glatt-com
Glattzentrum, ZH CH-8301
Switzerland

Phone: +41 1 878 7525
Email: lear@cisco.com

Intellectual Property Statement

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.