



www.fidelissecurity.com
www.threatgeek.com
@FidSecSys
+1 800.652.4020

Fidelis Threat Advisory #1005

REVERSE TUNNELS

October 13, 2011

Document Status: FINAL
Last Revised: 2011-10-13

Executive Summary

Reverse tunneling is a remote access method capable of allowing an external entity full control over an internal system while traversing network address translation (NAT) and bypassing both ingress and egress firewall filtering. This capability can be leveraged both by an internal user wishing to work from home in the absence of a legitimate means to do so, or by a malicious outsider wishing to gain control over a previously compromised system. In the former case the threat is due to a violation of the network security policy and increased risk due to an unmanaged system having direct access to the enterprise LAN. In the latter case the threat is that an initial compromise has already taken place and the attacker is strengthening their control over the enterprise, allowing them to increase the scope of the breach or exfiltrate sensitive data.

Threat Overview

Reverse tunneling is a method used for NAT traversal and to bypass both ingress and egress firewall rules for the purpose of allowing remote access to an internal system. Traditional remote access simply involves connecting to an open port on a system that is forwarded on the gateway firewall or NAT device; however this is usually tightly controlled by IT and restricted to select servers and services. Firewalls and NAT devices operate by denying all inbound traffic unless port forwarding is configured or unless the inbound traffic matches a previously established outbound session. Reverse tunneling circumvents this in one of two ways. The simpler way is that the internal system simply attempts to connect to a known external system at a given time interval; thus from a networking perspective the internal system is the client and the external system is the server. This is a reversal of the typical remote access scenario. The second method involves using the secure shell (SSH) protocol to create a tunnel from the internal to the external system, then leveraging the SSH protocol's port-forwarding flexibility to make it possible for an outsider to establish an inbound session at will, piggybacking on the previously opened connection (see "Further Reading" for an example).

Users are granted permission to copy and/or distribute this document in its original electronic form and print copies for personal use. This document cannot be modified or converted to any other electronic or machine-readable form in whole or in part without prior written approval of Fidelis Security Systems, Inc.

While we have done our best to ensure that the material found in this document is accurate, Fidelis Security Systems, Inc. makes no guarantee that the information contained herein is error free.



www.fidelissecurity.com
www.threatgeek.com
@FidSecSys
+1 800.652.4020

Risk Assessment

The two primary threats associated with reverse tunneling are network security policy violations and the potentially more serious threat of remote control of internal systems by a malicious outsider. Some sophisticated users might wish to access their work system from home, perhaps even for legitimate purposes, but are prevented from doing so by stringent firewall policies and the lack of a VPN-style solution for authorized remote access. Such a user could establish a reverse tunnel for an entirely non-malicious reason, but this would still violate network security policy. It also increases enterprise risk as that user's home system, which may not have the same security safeguards as enterprise workstations, is now logically within the security boundary. Thus if the home system is compromised, the attacker now has access to the enterprise network. The more serious threat; however, is that of a malicious outsider having remote control over a previously compromised internal system, where the initial compromise set up a reverse tunnel. This would allow an attacker complete control over an internal system, which could then be used to launch attacks against other enterprise systems or to exfiltrate stolen data.

Indicators & Mitigation Strategies

The most common protocols used for reverse tunneling are HTTP, SSL and SSH. Other tools such as Poison Ivy (Of RSA attack fame; see FTA #1001) do not use a common protocol and simply operate over TCP. HTTP and SSL are often used because those protocols are almost always allowed through a firewall with no restrictions. Traditional layer 4 firewalls are usually configured to allow any outbound traffic, regardless of protocol, on ports 80 and 443; this lack of granularity can be exploited by reverse tunnels using custom protocols. SSH is used because it comes standard on all Linux systems thus no extra software is required. SSL and SSH have the further advantage of providing an encryption layer thereby stymieing detection efforts. In a predominately Windows environment, SSH is rarely used, thus any outbound SSH traffic can be classified as anomalous and potentially suspicious. Any legitimate use of outbound SSH is most likely easily identifiable and can thus be excluded from such classification. Reverse tunnels using other protocols are much more difficult to detect and often require implementation-specific solutions, looking for protocol malformations unique to the implementation.

The Fidelis Take

Rules are available for Fidelis sensors that can detect the use of nonstandard protocols over ports 80 and 443. These rules can be used to detect tools like Poison Ivy that operate over those ports for the purpose of bypassing firewalls. Outbound SSH and IPSec traffic can also be detected; legitimate use of those protocols can be whitelisted to alert only on anomalous use of those protocols. SSL traffic can be decrypted on-the-fly and inspected using the Fidelis SSL Inspector for even greater visibility and control.

Further Reading

- HowToForge, "Reverse SSH Tunneling", kcharoen, 9/18/09. <http://goo.gl/6zrp7>
- SANS, "Reverse WWW Tunnel Backdoor", Chris Young. <http://goo.gl/AvTDA>