Ayisha Farhath Abbas
Assistant Professor
Department of Computer Science & IT

TUNNELING TECHNIQUES

Tunneling data through other protocols often bypasses these controls and may allow sensitive data to exit the network and unwanted data to enter.

The authors of SSH, the encrypted version of Telnet “on steroids,” designed it to be able to tunnel data over the connections it makes so that other applications and protocols could potentially be more secure.

Tunneling is a way to transfer arbitrary data in the payload of a protocol and then potentially interpret them differently or in some other extended way than originally intended.

A common, simple form of traffic tunneling in SSH is the tunneling of a Transmission Control Protocol (TCP) port. The protocol simply proxies a TCP connection over the SSH connection, and the content of the TCP connection does not flow directly from source to destination, but rather through the SSH connection.

Exhibit 2-4 shows how an SSH connection can tunnel a Telnet connection securely between trusted environments

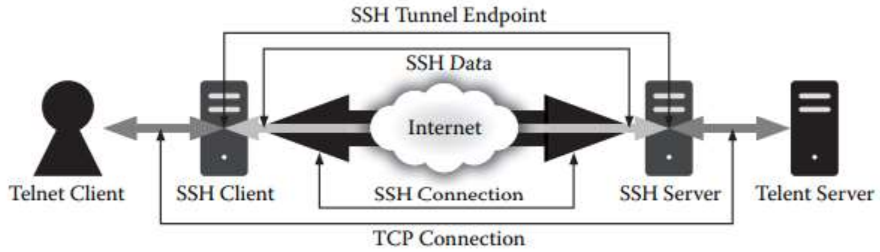

Exhibit 2-4 Telnet tunneled over a secure shell (SSH) connection.

The example tunnels traffic between two unrelated hosts that have no SSH capability.

The Internet Engineering Task Force Request for Comments (IETF RFC), which has published documents describing protocol standards, defines SSH’s port cleanly so administrators can filter it.

By writing custom applications that act as the client and server for other protocols in a given environment, malicious code can hide its activities and gain unfettered access to and from any network.

The most common unrestricted protocols in the enterprise are HTTP, the domain name system (DNS), and Internet Control Message Protocol (ICMP).

HTTP

HTTP has become the de facto high-level protocol on the Internet. It now carries audio and video streams, can transfer large files, and can even carry application-to-application remote procedure calls (RPCs).

Exhibit 2-5 shows the syntaxes of an HTTP request and reply that illustrate areas of the protocol that can contain discretionary information for data transfer.

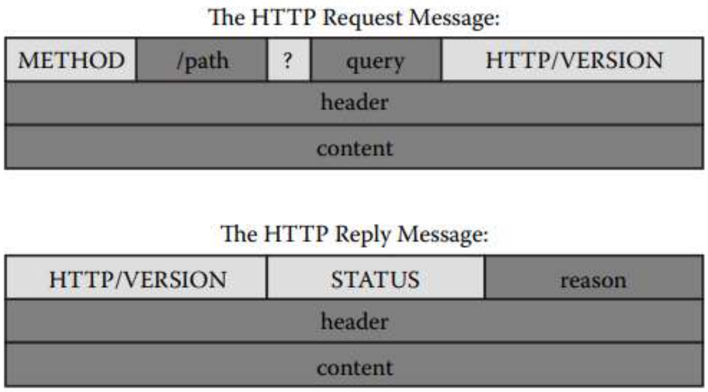

Exhibit 2-5 HTTP messages.

The protocol allows, in essence, unlimited space for content (or payload) in the request or reply message. This makes it convenient to transfer arbitrary data to and from an HTTP server.

Most malicious code already acts as a simple HTTP tunnel; in practice, it posts sensitive data to malicious Web servers for purposes other than to retrieve a Web-based resource. It also sometimes communicates over HTTP.

Attackers need a complete tunnel client and server. Some common software for the task is GNU httptunnel, JHttpTunnel, and Webtunnel.

HTTPS, which is HTTP secured over a secure socket layer (SSL) against eavesdropping and tampering, is no different from HTTP except that it makes detection harder. If a malicious actor cannot eavesdrop, he or she does not have a chance to detect known signatures of tunnels.

DNS

The DNS is the core directory service of the Internet. Without it, translations between names, such as www.verisign.com and IP addresses, could not happen. The DNS architecture is fundamentally different from that of HTTP.

DNS is a service that an administrator cannot block and must always make available. The most common delivery mechanism for DNS is the UDP, not TCP.

Exhibit 2-6 shows the layout of DNS message packets; the darker areas indicate where software can hide payloads.

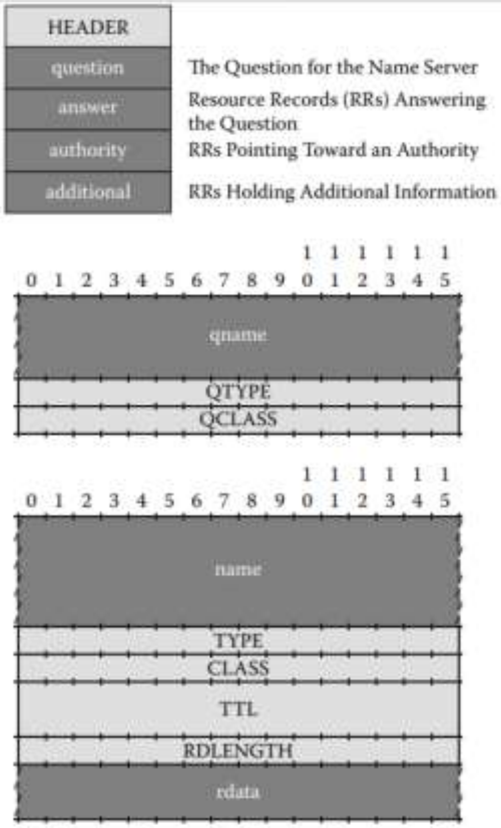

Exhibit 2-6 DNS messages. All communications inside of the domain protocol are carried in a single format called a message (top); the question section is used to carry the "question" in most queries (middle), and the answer, authority and additional sections all share the same format (bottom).

DNS tunnels need many packets to transfer large amounts of data. In most cases, the tunnel client is simple end user software that makes many requests for nonexistent hosts.

The tunnel server is generally a rogue (fake) DNS server where intermediary resolvers eventually route the questions. Thus, this setup may require more infrastructure access than just a simple Web server does because an attacker needs to delegate DNS authority to a rogue system.

ICMP

ICMP is a signaling protocol for IP. It is used mostly to deliver status and error messages when IP-based communication errors occur or to troubleshoot and test connectivity status.

ICMP echo messages, which users and administrators alike use to test the accessibility of a host, are well suited for tunneling. Ping, as the most common software that implements this ICMP mechanism, sends data to a host and expects a reply.

Exhibit 2-7 is the layout of an ICMP echo message, again showing payload areas.

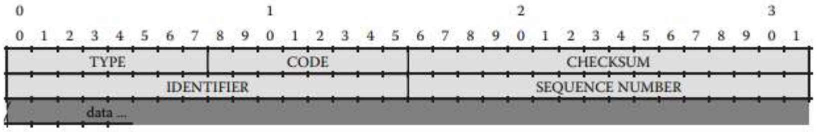

Exhibit 2-7 Internet Control Message Protocol (ICMP) echo message.

ICMP offers good throughput in both directions. ICMP tunneling was one of the earliest methods publicly available to transmit traffic over a protocol in a covert way.

Intermediaries, Steganography, and Other Concepts

Aside from the three common tunnels, hackers can modify any protocol that filters through a firewall to behave as a tunnel.

The task of tracking down specific tunnels and at least shutting down those that are readily apparent is a quick step: identify the destination and block it. This task can become much more difficult with advanced implementations such as intermediary hosts.

Once the payload arrives at the intermediate service, the destination side of the tunnel picks it up and delivers a reply.

The intermediary problem can be even more complex. Steganography is the practice of hiding messages and data in content that is not readily apparent and is a form of security through obscurity. For example, steganographic software and tools can encode messages and data into images.

Timed communication can be accomplished using lower-level protocols, such as raw IP packets.

Detection and Prevention

While the firewalls and IDS that are in place today have their roles to play, they may not be able to identify or prevent tunneling. Tunnels abuse protocols in a way that matches the syntax or the rules of the specification but not the intent, so despite the efforts of vendors using static signatures for detection.

Attackers can easily modify open-source tools to appear slightly different from the original, thus defeating a static rule. Packet inspection firewall rules and IDSs can go only so far in identifying and blocking the threats. Tunnels do have a weakness. They almost never adhere to historical or trended traffic patterns.

Tools like SilK23 and Sguil24 provide a foundation for trending the network and baselining behavior.

Covert exfiltration of information through tunnels is bound to increase as the tools to detect existing software and controls of existing methods become stronger. Broad-based dynamic analytics need to be part of any network user's strategy to ensure identification of the not-so-obvious threats that may be emerging.