White paper

How to defeat man-in-the-middle attacks

Verizon SDP

Executive summary

Man-in-the-middle attacks expose organizations to a number of vulnerabilities that can result in stolen intellectual property, business disruption and financial loss.

Adversaries become the man-in-the-middle when they intercept communication between a user and the server that the user is trying to communicate with. The attackers then obtain the unencrypted information, or cleartext, intended for that server. By placing themselves between users and servers, adversaries can gain access to (and control of) a business' valuable data.

Man-in-the-middle is a widely used type of attack which requires two main factors to successfully execute. First, the adversary intercepts communication from a victim and relays it to the server that the victim wishes to connect to. Then, the adversary obtains the unencrypted data transferred between the victim and the victim's server.

The Software-Defined Perimeter (SDP), a protocol specification created by the Cloud Security Alliance, can defeat network-based attacks. Verizon SDP, a high-performance implementation of the protocol, tackles man-in-the-middle attacks using a combination of mutual Transport Layer Security (TLS), pinned certificates that use IP addresses instead of host names and a fixed encryption suite that cannot be downgraded or altered.

This paper demonstrates various techniques adversaries use to intercept traffic and obtain cleartext, and how Verizon SDP prevents attackers from accessing cleartext.

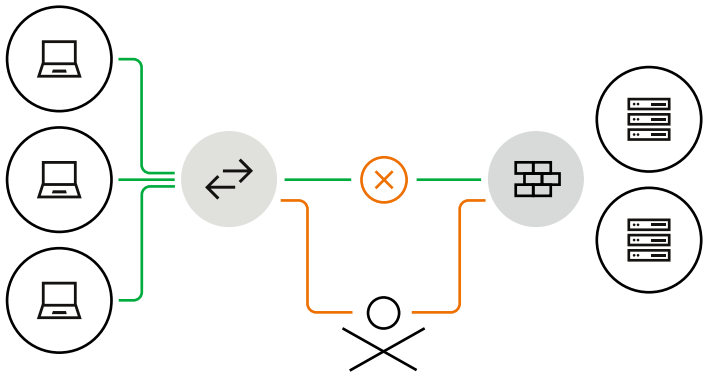

Figure 1. To execute a man-in-the-middle attack, the adversary must first redirect traffic from its normal path (shown in green) to a path that makes the traffic flow through the adversary (shown in orange). The adversary must then decrypt any encrypted traffic.

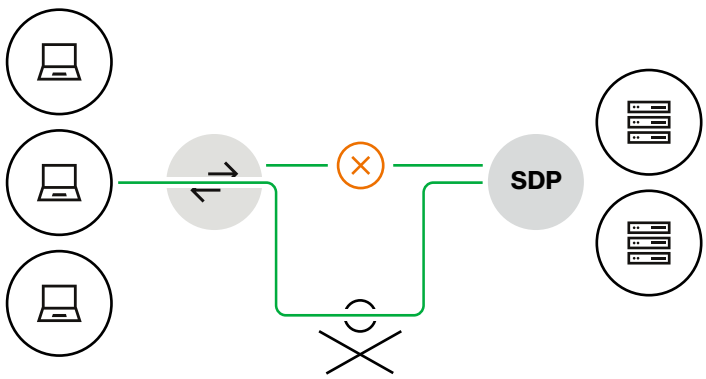

Figure 2. Verizon SDP does not defeat the redirection of traffic. That happens at the networking layer. However, Verizon SDP defeats the decryption of the traffic.

verizon

Traffic redirect

In a man-in-the-middle attack, the adversary either relays traffic from the client to the server, or becomes both the server to the victim's client device and the client to the server the victim wants to communicate with. There are several ways attackers make this happen.

Spoofing a Wi-Fi hotspot

The adversary becomes an access point of a well-known Wi-Fi name, such as the Wi-Fi network name users would expect to find at a hotel or coffee shop. Since most devices automatically reconnect to any access points they have previously connected to, an adversary can quickly become the man-in-the-middle of multiple devices, affecting many victims at once.

Another common way is for the adversary to broadcast a Wi-Fi name that is very similar to the authentic name. When the victim chooses the similar name instead of the authentic one, the adversary becomes the man in the middle.

Spoofing a website

It's relatively easy for an adversary to perform a man-in-the-middle attack via spoofing a website. The adversary registers a domain name for a website that is very similar to the domain name of a real website (e.g., varizon.com for verizon.com). Since the adversary owns the domain name, the adversary can get a legitimate SSL certificate for the bogus website domain name.

Finally, the adversary does a phishing attack, or some other type of social engineering attack, on a group of potential victims to get them to go to the bogus website – thinking it is the legitimate website. Obviously, the bogus website is the man-in-the-middle. Therefore, it then relays traffic to the real website with the adversary seeing the content.

ARP Spoofing

Address Resolution Protocol (ARP) is the protocol that client devices invoke to get a server's Ethernet MAC address from the IP address. The typical workflow for this very common networking protocol is that a user types a URL into a browser, the client's operating system uses the URL to request the IP address of the server from the Domain Name System (DNS) server, and then the operating system uses the IP address returned from DNS to request the Ethernet MAC address of the server using ARP.

Note that to reduce the number of times clients perform an ARP, clients will often listen to and store the ARP replies from servers when other clients make ARP requests. Therefore, an adversary has two ways to become a man in the middle between the client and the server:

- The adversary can be very fast at replying to an ARP request with the adversary's MAC address. With a faster reply, the client believes the adversary's host is the legitimate server
- The adversary also can generate a "gratuitous ARP," which is an ARP response that was not prompted by an ARP request. The gratuitous ARP is sent as a broadcast, as a way for a node to announce or update its IP-to-MAC mapping to the entire network. The hosts listening on the network accept the binding of an IP address with an Ethernet address as legitimate. Therefore, the hosts store the adversary's MAC address associated with whatever IP address is in the gratuitous ARP

To accomplish either attack, the adversary needs to have control of a host on the victim's LAN. Therefore, typically, the adversary will use a phishing attack to compromise a computer on the victim's network and then use this ARP attack to obtain credentials for lateral movement through the network.

Compromised DNS

In this attack, when the victim's client goes to the DNS server to get the IP address of a legitimate server, the victim instead gets the IP address of the adversary's server.

There are multiple ways to spoof a DNS server:

- The adversary acts like a fast DNS server and returns the IP address of the requested DNS name more quickly than the legitimate DNS server. This requires the adversary to be close to the victim
- The adversary compromises the local DNS server to change the IP address of certain domain names that instead point to the adversary. This allows the adversary to spoof a region of the network
- The adversary can compromise an authoritative DNS server. If a local DNS server does not know the IP address of a requested domain name, it will call upstream DNS servers to get the name. Each DNS server, in turn, will call additional DNS servers until one reaches the authoritative DNS server for that domain. Therefore, if the adversary compromises an authoritative DNS server, they can redirect everyone in the world to the adversary's fake IP address

Vulnerable infrastructure

There are many networking elements between a client and a server: switches, routers, firewalls, lots of security devices, load balancers, etc. If the adversary compromises any of those devices, then the adversary becomes a man-in-the-middle for that device.

It is also surprisingly easy to compromise a component of the infrastructure, from default passwords and poorly configured SNMP, to unpatched vulnerabilities on embedded operating systems that do not get upgraded at the same rate as servers. These are just a few of the elements that can be exploited.

Hijacked internet route

Border Gateway Protocol (BGP) is the protocol that determines the path data takes between ISPs. By broadcasting either shorter routes or more explicit routes to another ISP, an ISP can hijack the traffic. Bad actors on global levels might inject fake routes into the BGP routing tables, so they could create a man-in-the-middle attack on a huge amount of data.

This approach probably would not be used by a typical adversary, but it does go to show just how many ways there are to execute a man-in-the-middle attack.

Cleartext

As described above, the first step is for the adversary to intercept the traffic. The second step is to remove any encryption from the traffic to obtain cleartext. Verizon SDP can defeat the adversary from seeing the cleartext in a variety of attacks.

Data already in cleartext

Network protocols that do not encrypt traffic provide the cleartext to the adversary without additional effort. The most notable protocol that uses cleartext is HTTP. Importantly, not only is the data in cleartext, but so are the cookies, session tokens and other input parameters. Cookies and session tokens act as short-term credentials for accessing websites. Therefore, when the man-in-the-middle adversary obtains the cleartext cookie or session token, the adversary can impersonate the victim, connecting to the website as that person.

Verizon SDP encrypts all traffic from the user's device to the Verizon SDP Gateway – including HTTP traffic. If the traffic is cleartext, it gets encrypted, including the cookies, session tokens and other input parameters. If the traffic is cyphertext, it gets encrypted a second time.

Transforming HTTPS back to HTTP

To mitigate the attack above, most websites are now using HTTPS instead of HTTP, where HTTPS uses TLS encryption to provide secrecy and data integrity of the HTTP traffic. If an adversary becomes a man in the middle, they must decrypt the HTTPS traffic.

One of the more elegant ways the adversary can convert HTTPS traffic to HTTP traffic is to be an HTTP server to the victim and an HTTPS client to the legitimate server at the same time. Very few users check whether their device is connected to the server via HTTP or HTTPS. Therefore, the HTTP connection from the user's device to the adversary seems normal. And since the adversary's device then converts the HTTP stream to HTTPS and sends the HTTPS to the server, the server believes data is secure as well.

Verizon SDP uses mutual TLS to authenticate both the client to the server and the server to the client. The adversary does not have the private key for the mutual TLS, so the adversary cannot impersonate the user to the Verizon SDP controllers or gateways during the TLS handshake.

SSLstrip and the Mana Toolkit

The HTTP man-in-the-middle attack was first demonstrated in 2009 with a program called SSLstrip. To the user, it looked like a regular HTTP session, and all the user's data, login credentials, cookies, session tokens and other input parameters were in cleartext. The adversary could see the user's cleartext, but then encrypted the data in TLS for the connection to the server. The server sees cyphertext, as required.

In an attempt to defeat the SSLstrip attack, the browser industry created the HTTP Strict Transport Security (HSTS) protocol, a mechanism by which a website is able to inform the browser if it's supposed to be secured with Secure Sockets Layer (SSL) end-to-end. However, a more recent application that is part of the Mana Toolkit now defeats the HSTS protocol, again allowing the adversary to perform the HTTP attack.

Attacks on SSL

There have been a number of different kinds of attacks on SSL and TLS (see Appendix A for full details). Below we have identified some common themes to those attack methods and how Verizon SDP can be used to defeat those attacks.

Some attacks are based on the fact that the client and the server are verified separately, such that each step can be spoofed separately. Verizon SDP uses mutual TLS to defeat this attack. Mutual TLS authenticates both the client to the server and the server to the client in a two-way handshake at the same time.

Many attacks use JavaScript to initiate the assault on the victim's browser. Verizon SDP defeats JavaScript-based attacks because the Verizon SDP client that creates the mutual TLS connection is not a browser and does not run JavaScript.

Another set of attacks uses forged certificates. This is possible because there are so many certificate authorities in the world that are trusted by the browser. Verizon SDP defeats forged certificates by using pinned certificates. That is, instead of trusting the hundreds of certificate authorities in the world like a browser does, the Verizon SDP client only trusts certificates issued by the Verizon SDP Certificate Authority of the Verizon SDP instance. Each instance of Verizon SDP has its own unique, dedicated Public Key Infrastructure (PKI) and its own certificate authority.

A fourth set of attacks is based on the adversary's ability to downgrade the encryption cypher being used or alter other parameters of the HTTP/S protocol suite. Verizon SDP defeats this set of attacks because it controls both sides of all connections. It can therefore use one, and only one, encryption suite – one that cannot be downgraded nor have its parameters changed. And note that Verizon SDP uses one of the strongest encryption algorithms commercially available.

Finally, some attacks are possible because the server is on the internet such that an adversary can connect to it with TLS. Verizon SDP defeats this set of attacks by requiring Single Packet Authorization prior to allowing access to the TLS protocol. With Verizon SDP, the user's device cannot begin the TLS handshake before passing Single Packet Authorization.

Summary

Man-in-the-middle attacks happen behind the scenes, causing many businesses to be unaware that they are occurring. Adversaries have multiple methods they can deploy to intercept traffic, decrypt data and move laterally through the network. Verizon SDP defeats these man-in-the-middle attacks.

Verizon SDP applies a zero-trust approach to networking for remote access, internal network segmentation and cloud applications. Organizations can defeat network-based attacks and gain peace of mind with the right safeguards in place.

Contact us.

Read more about how Verizon SDP provides zero-trust remote access, internal network segmentation and cloud access.

Click here for Verizon SDP

Appendix A

Appendix A is a list of different attacks on SSL and TLS. Included is the common name of the attack, a short description of the attack and how Verizon SDP defeats the attack.

Name	Attack	Defeats Attack
SSLstrip	Man in the middle (MiTM) HTTP to HTTPS	Mutual TLS
THC-SSL-DOS	Server denial-of-service (DoS) attack	SPA
DigiNotar	MitM forged certs	Pinned certs
BEAST	MitM Java Applet oracle	PA client is not a browser
CRIME	MitM SPDY compressing oracle	No compression in cypher
Lucky 13	MitM CBC padding oracle	GCM cypher not vulnerable
TIME	MitM browser JavaScript timing oracle	PA client is not a browser
RC4 biases	MitM RC4 oracle	No cypher negotiation
BREACH	Website redirect, compression	No redirect or compression
goto fail	MitM counterfeit key via coding error	Pinned dedicated cert
Triple Handshake	MitM on client cert	Pinned dedicated cert
Heartbleed	OpenSSL bug	Not single-ended SSL
BERserk	MitM PKCS#1.5 padding	Cypher not vulnerable
POODLE	MitM SSLv3 oracle	No cypher negotiation
Poodle++	MitM JavaScript timing oracle	PA client is not a browser
FREAK	MitM negotiation 512-bit key	No key negotiation
Bar Mitzvah	MitM on RC4	No cypher negotiation
logjam	MitM downgrade to 512-bit key	No cypher negotiation
DROWN	MitM downgrade to SSLv2	No cypher negotiation
Sweet32	MitM birthday attack on 64-bit ciphers	64-bit cypher not used
SHA-1 collision	MitM collision attack on SHA-1	SHA-1 not used