ALPACA Attack Report

*Group: Lachlan Macartney (5129496), Suppanut Chaimathikul (6991678), Toby Meyrick (1998580), Jake Perkins (2713308)*

Transport Layer Security (TLS) is a layer between the transport and application layers and therefore applies to all communication over the internet, so maintaining its integrity is important. The ALPACA attack, a “content confusion attack,” takes advantage of the gap created between the transport layer and the application layer thus avoiding the key security features of TLS completely. ALPACA attacks allow attackers to intercept user traffic without disrupting the communication session which grants attackers access to session cookies and user data through weak web servers. ALPACA attack is a cross-protocol attack as it targets both the transport layer and the application layer via a security flaw in TLS.

Within the greater ALPACA attack, there are three different attack methods: upload attack, download attack, or reflection attack. During an upload attack, the attacker can obtain personal user data including session cookies and passwords; during a download attack, the attacker can execute a Cross-Site Scripting (XSS) attack they have previously stored; and during a reflection attack, the attacker can execute a reflected XSS on the target website.

ALPACA attack is made possible because of insecurities specific to TLS server authentication, explicitly, TLS does not protect the TCP connection. This means that attackers can redirect traffic between servers when their certificate matches that of the intended server, and that any server using TLS is considered at-risk of the ALPACA attack. Servers are more at-risk if they are sharing certificates with an FTP server as this leaves the server more exposed, and the more certificates the server uses, the more vulnerable they are. Servers also become more susceptible if they access an at-risk application server. Users are more at-risk if they are using Internet Explorer or outdated browsers as they respond to non-HTTP responses differently. Due to requiring “an active Man-in-the-Middle attacker,” it is accepted that the attack is less likely to occur on the wider internet as attackers are more likely to operate on a local network.

Due to their nature, cross-protocol attacks require many conditions to execute correctly. They require the certificates to match, a weak application layer, and the attacker must have the ability to upload, download, or reflect data. While execution is difficult and highly conditional, the ALPACA attack is almost impossible to prevent. Being cross-protocol, enforcing security on the transport layer is not enough, but enforcing security on the application layer is not enough either. The security must be increased in the TLS, but this is a TLS feature flaw, not a flaw of any specific programmer. There is no way for an individual or a business to block any attempt at an ALPACA attack, only methods of mitigation.

Decreasing the likelihood of a successful ALPACA attack can be done by having more extensive conditions for closing the connection which makes detecting a possible attack easier, implementing certificate exclusivity which makes it harder for the attacker to pass through undetected, application layer protocol negotiation which will prevent the clients from completing the TLS handshake if a possible attack is detected, and server name indication which checks that the hostnames match. All these concepts only decrease the likelihood - they cannot prevent an attack, even if all the steps are taken together. What would prevent the attack is an upgrade to the TLS adding a feature to prevent cross-port attacks but this would be at the cost of the rollout of a global layer extension.

References:

*https://alpaca-attack.com/*