The BEAST Wins Again: Why TLS Keeps Failing to Protect HTTP

Summary for the non-expert

The goal of this briefing was to demonstrate new attacks against HTTPS, the encryption technology used to protect all websites against eavesdropping and impersonation.

An important claim was to point out that the general public is increasingly worried about pervasive surveillance and its effects on online privacy. However, HTTPS was not designed to protect privacy (this was illustrated by a video showing how the NSA can easily infer Google search terms just by monitoring encrypted network messages). Instead, users should be increasingly worried about attackers that are actively trying to compromise them by tampering their encrypted messages. This can happen both on public networks (such as public Wi-Fi) and private networks. Notably, the technology that maps domain names, such as google.com, to network addresses, can be subverted for active tampering, an increasingly common practice that has been used by governments such as the US, UK, Turkey, China…

Another important claim to explain why new attacks have been discovered at a steady rate in the past years, despite HTTPS being a highly critical and scrutinized technology, is the fact that when new attacks are discovered, the response has often been to patch the symptoms of the attack rather than its root causes. As a result, people keep coming up with new ways of exploiting the same, ancient weaknesses (most of them are 5 to 20 years old!), because they have been improperly fixed.

As a first example, the Cookie Cutter attack shows that using a well-known vector, it is possible to disable the weak mitigations that are supposed to protect cookies, the technology famously abused for tracking users, but that is otherwise universally used to log into website. When such login cookies (also called session cookies) become known to the attacker, he is able to fully impersonate the user the session belongs to, be it on Twitter, Facebook, Google, or any other website. Thus, using the Cookie Cutter attack, an attacker is able to steal the sessions of the victim on a very large proportion of websites as soon as the login form is submitted.

The second class of attacks targets the way servers handle the page requests from clients in a cloud setting, where this server is typically responsible for not one, but many websites. These servers need to prove their identity to clients to prevent impersonation. Such identity proofs are called certificates, and are issues by the trust authorities of the Web after identity verification. When servers handle many different websites, they typically use many certificates; furthermore, each certificate can prove the identity of more than one domain. For instance, the same certificate can assess control over both twitter.com and pics.twitter.com.

The family of attacks discussed during the briefing relies on a server having credentials that are valid for both websites that it serves, and websites that it doesn't. This is a frequent occurrence on today's Web, but because of the way servers react to request sent to domains they don't handle, it can be used to defeat the security isolation between different domains (this security isolation explains why malicious websites normally cannot compromise good websites). For instance, a demo shows how by confusing a Dropbox domain that can potentially store malicious files with another where the user enters his credentials, an attacker is able to break into the victim's Dropbox account.

The briefing contains several other concrete demos of this attack pattern. First, one depicts how to steal the access tokens used by single sign-on providers (most commonly, Facebook, Twitter, Google, and LinkedIn). When users use the "Login with Facebook" feature, their identity on the website is confirmed by the access token issued by Facebook on the behalf of the user. If this token is leaked to the attacker, he becomes able to impersonate the victim. A video illustrates this attack against Pinterest, but it also applies to thousands of other websites.

Another class of attacks relies on a performance optimization that allows browsers to only check the certificate of a website when it first connects to it. Further pages and pictures loaded from the same site will reuse the same encryption key stored in a cache. Depending on their configuration, some cloud servers will store all keys in the same cache, regardless of the website the key was created for. When this occurs, an attacker may be able to bypass the certificate verification step by redirecting requests from one server to another that shares the same cache but serves different websites. A demo of this attack shows an attacker compromising a high trust Mozilla domain by redirecting requests to another Mozilla server that deliberately contains vulnerabilities that the attacker can exploit.

The last attack shown is also the one with the largest impact. It demonstrates the fact that due to the way their servers are configured, Akamai (a leading provider of cloud technologies, used by most of the top websites in the world) allows an attacker to redirect requests made to Akamai customer websites (such as Twitter, PayPal, LinkedIn, Apple, CNN…) back to himself. The impact of this attack is catastrophic: depending on the certificate stored on the Akamai server, the attacker can either steal the session cookies of the victim on these websites (this was for instance the case on Twitter and PayPal) or completely impersonate the website (this was shown on the LinkedIn, CNN, and NSA websites).

Lastly, the author discovered that a new feature in the next-generation Web protocol created by Google and adopted by Mozilla and Microsoft can potentially increase the risks of breaking the isolation between domains. Indeed, he mentions that he discovered another attack that allows server impersonation, but doesn't disclose it because it hasn't been patched yet.

The last part of the talk is targeted at specialized security experts and has little interest for the general public.