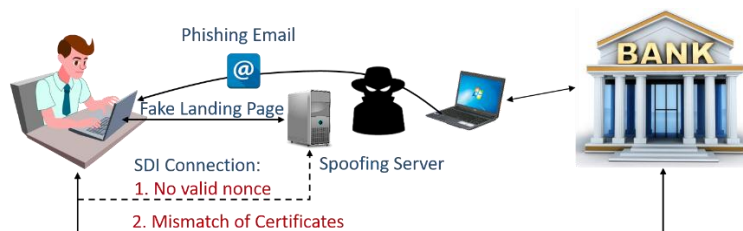


SDI and Phishing Attacks

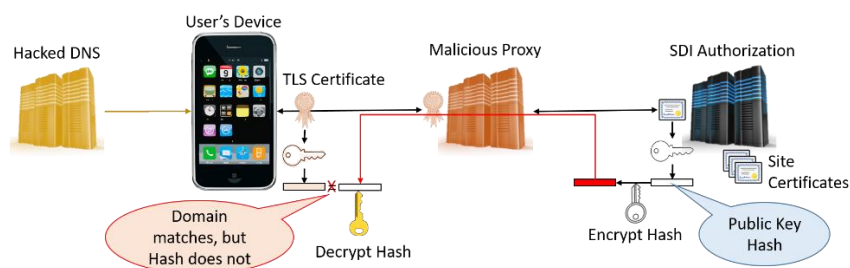
The usual scenario in phishing attacks is setting a malicious link in an authentic-looking email, an SMS message, or a web page from a search engine. By clicking or tapping that link, the user is taken to a malicious server, which presents an authentic-looking landing page. If the user types a password, the hacker can get it and impersonate the user on a real server. There are also other, more sophisticated attacks, such as hacking the DNS (Domain Name System) service so that when the user types the real URL, the routing takes them to a malicious server, which, in the worst-case scenario, could be a malicious proxy server.

Adding MFA (Multi-Factor Authentication) makes it a bit more difficult for a hacker, but not impossible to overcome. SMTP (Simple Mail Transfer Protocol) for emails that involves vulnerable hopping servers, or not secure SS7 (Signalling System No. 7) for SMSs, can be easily hacked to intercept verification codes. Dedicated authenticators improve security, but in phishing schemes they are as vulnerable. Once the user is connected to a malicious server, a hacker will receive anything required by the site to authenticate directly from the unsuspecting user.

When the user registers SDI, the user's password becomes invalid, so phishing it no longer makes sense. In addition, the SDI app must be on the same device as the browser/app that the user connects to a site to be able to get the actual URL and retrieve the same site's certificate. Also, the SDI app must receive a nonce encrypted with the user's account public key. In the initial request, the SDI app may also provide the server with a nonce encrypted with the server's public key to strongly verify that server.



Let's look at two scenarios of phishing attacks where SDI protects the connection. A simple phishing/spoofing attack where a hacker uses a server that is not a proxy. The SDI app will get the URL from a browser or from an SDI-enabled mobile/desktop app. The SDI app stores all domains where it is registered, so the URL will not match any of its domains. With hacked DNS and connecting to a given URL, the malicious server will not be able to provide a valid nonce. In either case, the phishing/spoofing will be unsuccessful.



Now let's look at a very sophisticated spoofing/phishing attack, where the DNS is hacked, and the malicious server is a proxy server re-encrypting traffic. Let's also assume that the CA (Certification Authority) was also hacked and the hacker uses a valid certificate with the site's domain. In this case, all known authentication methods, including SDI, will fail. However, because the SDI platform performs dynamic certificate pinning, it will be able to detect the attack and defeat it.

When connecting to a site, the SDI app will verify the server (destination, not the malicious proxy) with the positive result and receive a valid nonce, so no detection has happened so far. The first transaction also includes the hash of a public key of the site's TLS certificate encrypted with the user's account public key. The TLS certificate obtained during the connection will be the one that was provided by the malicious proxy server, so the domain will be verified as valid, but the hash of its public key will not match the hash obtained from the SDI authorization server. That will detect the malicious proxy and defeat it. Even though the malicious proxy has full access to the traffic, it will not be able to substitute the encrypted hash of a public key or the directives, which are signed. It will only be able to watch that traffic.

In conclusion, no matter the sophistication of the phishing/spoofing attack or its execution, SDI is able to detect and defeat it.
