Reaktive Sicherheit

Übungsblatt 2

Balduin Binder [s6babind@uni-bonn.de] Charlotte Mädler [s6chmaed@uni-bonn.de]
Bünyamin Sarikaya [s6busari@uni-bonn.de]

Aufgabe 1

Aufgabe 1.1 Erklären Sie kurz, wodurch Heartbleed ermöglicht wurde.

Das Problem lag bei OpenSSL, eine Applikation die SSL implementiert, nicht mit demProtokol SSL selbst. OpenSSL wird aber sehr viel benutzt (bei entdeckung von Heartbleed wurde vermutet, dass etwa 50% aler Websiten hiervon betrofffen waren). Server haben nur eine bidngte Anzahl an Sockets, es gibt also ein Timeout für inaktive Verbindungen. Um dies bei aktiver Nutzung zu verhindern schickt der Computer des Nutzers einen sogenannten "Heartbeat" an den Server, um Aktivität zu zeigen. Der Heartbeat kann Daten bis 64kB beinhalten un der Server antworten mit der selben Menge an Daten. OpenSSL überprüft aber nie den Wert der Größe sondern "vertraut" auf dessen richtigkeit.

Aufgabe 1.2 Erklären Sie kurz, wie Heartbleed für einen Angriff ausgenutzt werden kann.

"Heartbleed" kam zu standen wenn man bsp. 1kB Daten geschickt hat, dem Server aber gesagt hat es wären z.B. 64kB. Dann schickt der Server den gesendeten 1kB zurück und zusätzliche 63kB die er aus dem Speicher auffüült. Dies könnte unteranderem auch Nutzername/Passwörter/etc. beeinhalten. Ein Angreifer kann diese Daten dann nutzen.

Aufgabe 1.3 Wie können Sie in Ihrer Rolle als Systemadministrator erkennen ob Ihr System von einem Angriff betroffen ist, der durch Heartbleed möglich wurde? Begründen Sie Ihre Antwort.

Ein Heartbleedangriff hinterlässt keine direkten Spuren, also ist es nicht Möglich im Nachhinein konkret zu überprüfen ob Passwörter gestohlen worden sind. Es gibt aber Seiten mit den man testen kann ob es aktuell verwundabrkeiten gibt (Bsp:http://filippo.io/Heartbleed/ oder das Go-Script Heartbleed) und dadurch eine Gefahr besteht. Zu empfehlen wäre auf OpenSSL 1.0.1G (oder neuer) zu Updaten, neue Zertifikate für neu Erstellte Keys aus zustellen, sie zu installieren und zu verifizieren, die alten zurückzunehmen und sämtliche nutzernamen und passwörter (wichtig, erst danach) zu ändern (Undzwar auf allen eventuell betroffenen Systemen).

Aufgabe 1.4 Ist Heartbleed auch relevant für Clients? Welche OpenSSL-Version verwendeten damals aktuelle Android oder iOS Smartphones? Nennen Sie die von Ihnen verwendeten Quellen.

Applenutzer waren von Heartbleed nicht betroffen. OS X verwendete die ältere OpenSSL-Version 0.9.8, die Heartbleed noch nicht beinhaltete. iOS wiederum kommt standardmäßig überhaupt nicht mit OpenSSL. Android-Nutzer waren dagegen anfällig, wenn sie Version 4.1.1 (erstes Update von Jelly Bean) installiert hatten. Bei späteren Versionen hat Google dies abgeschaltet. Quelle:https://www.heise.de/mac-and-i/meldung/Heartbleed-Sicherheitsluecke-OS-X-und-iOS-nicht-betroffen-zumindest-nicht-direkt-2167417.html

Aufgabe 2

Aufgabe 2.1 Beschreiben Sie die Vorgehensweise eines Angreifers beim DNS Spoofing

Bei einer DNS Operation gibt der Nutzer eine Website in den Browser ein, der dann den lokalen DNS Server nach der Adresse fragt und dann den User zur gewünschten Website weiter leitet. Hierbei hat der DNS Server einen Cache, wodoruch die Abfrage schneller wird. Wenn die Website nicht im Chache ist, wird der "Master" DNS server via Internet gefragt. Ein Angreifer würde den lokalen DNS Anfragen und wärend dieser auf eine Antwort vom Master DNS wartet, ihn mit gefälschten Antworten überfluten, z.B. alternative Ip-adresse (IP-Spoofing). Diese wird dann in den Cache aufgenommen. Wenn nun ein nutzer die selbe Anfrage zum lokalen DNS stellt, wird er zur fake website weitergeleitet (z.b. Phishing oder Pharming) DNS Spoofing wird deshalb auch passend "Cache poisining" gennant.

Aufgabe 2.2 Beschreiben Sie mögliche Schutzmaßnahmen.

Mögliche Schutzmaßnahmen können sein, dass der DNS Sever weniger vertrauen in Inofmration von anderen Servern setzt oder Antworten ignoriert die nicht direkt relevant zur Anfrage sind.

Aufgabe 2.3 Wie konnte es zu den in „The Collateral Damage of Internet Censorship by DNS Injection" beschriebenen Kollateralschäden kommen?

arises from resolvers querying TLD name servers who's transit passes through China rather than effects due to root servers (F, I, J) located in China Collateral damage occurs when a DNS query from a recursive resolver enters a censored network, causing the censorship mechanism to react. Although intuition would suggest that this would be a rare occurrence, there exist several factors which may cause the censor to receive and react to DNS queries from outsiders. Iterative Queries (a simple "lookup" may generate numerous queries, the disruption of any by censorship would cause resolution to fail.), Redundant Servers and Anycast (multiple servers in multiple networks to increase reliability,with 13 different roots and 13 global TLD servers for .com, a resolver may experience collateral damage if a path to any one of these 26 IPs passes into a censored network Anycast:single IP address may represent a widely deployed system of servers.) ,Censored Transit and Dynamic Routing (The paths from the resolver to the authorities is dynamic, routing through a series of Autonomous Systems (AS), If one transit AS implements censorship, then all traffic which passes through that AS experiences censorship, even if both the source and destination are in non-censored networks.)