TK1 - Theorie Aufgabe 1

Arne Link (1582381), Lars Fritche (1691285)
Gruppe 14 (S217/103)

November 15, 2013

1. Aufgabe 2.1

1.1. RFCs

a). Name, origin, and purpose

RFC steht für 'Request of Comments'. Bei einem RFC handelt es sich um eine organisatorische bzw. technische Spezifikation zum Internet, wie zum Beispiel Netzprotokolle.

Begonnen wurde die RFC Dokumentation im Jahr 1969, zu Zeiten des Arpanets, einem Vorläufer des Internets. Wichtige Merkmale eines RFCs sind zum einen die Formalismen über Sprachelemente, die in RFC's oft verwendet werden. Dazu zählen beispielsweise 'REQUIRED', 'SHOULD', 'MUST' oder 'MUST NOT'. Durch die eindeutige Spezifikation dieser bleibt dem Leser eines RFC kein Spielraum für falsche Interpretationen. Darüber hinaus werden alle Veränderungen eines RFC dokumentiert. Ein einmal abgeschlossener RFC darf niemals wieder verändert werden. Er kann in dem Fall nur durch einen neuen RFC abgelöst werden.

Während der eigentliche Begriff 'Request for Comments' darauf hindeutet, dass nur Kommentare zu einer neuen Spezifikation gewünscht werden, bleibt ein RFC auch nach der Akzeptanz als Standard weiter bestehen.

Die einzelnen Stufen eines RFC's sind die folgenden:

1. Informational: Diese Stufe ist als Ansatz gedacht, der eine Grundidee beschreibt, jedoch noch nicht zur Anwendung kommt.
2. Experimental: In diesem Stadium werden erste Prototypen getestet, die den RFC implementieren.
3. Proposed Standard: Ab diesem Stadium steht die Standardisierung des RFC's zur Debatte.
4. Draft Standard: Für dieses RFC Stadium müssen zwei unabhängige Implementierungen des RFC's begutachtet und kommentiert werden.
5. Standard: Ab diesem Stadium ist ein RFC als offizieller Standard akzeptiert

b). Examples

RFC 1149 beschreibt das Senden von IP Paketen via Vogelpost, indem der Brieftaube das zu sendende Paket ans Bein gebunden wird. RFC 2549 erweitert RFC 1149 um verschiedene Qualitätslevel und Straußenvögel, Vorschläge zum Labeling, Verschlüsselungen und Queueing

Der in den zwei RFC propagierte Ansatz ist weder für P2P noch für Client/Server Kommunikation verwendbar, da der Datendurchsatz sehr gering ist. Darüber hinaus ist die Verzögerung zwischen Senden und Empfangen enorm und Stürme oder Kälteperioden beeinträchtigen die erfolgreiche Empfangsrate enorm. Darüber hinaus werden die Informationen via Papier übertragen, was bei einem plötzlichen Regenschauer zu sehr vielen Bitfehlern führen kann, wodurch die

Nachricht falsch interpretiert werden könnte. Während bei einem normalen Server/Client - Modell die Verbindungen langsam steigen, mit hinzukommenden Clients, sind in einem P2P System Peers mit vielen anderen Peers verbunden, was zu einem rasanten Zuwachs der benötigten Brieftauben bzw. Vogelsträuße führen kann. Außerdem wurde im Paper nicht hinreichend erklärt, wie die Logginginformationen aus den Vogelhäufen extrahiert werden können. Mögliche Optionen dafür wären aber beispielsweise GPS Logger oder LIDAD / RADAR Systeme, wobei sich erster negativ auf die maximale Paketgröße auswirken und letztere zu einer verringerten Haltbarkeit der Trägermedien auswirken würde.

Auch unerforscht ist außerdem, inwiefern RFC 1149 und 2549 mit RFC 2324 kompatibel sind.

1.2. DNS

a). What is a spoofing attack?

Bei einem Spoofing Angriff versucht ein Angreifer sich als eine andere Person oder ein anderes Programm auszugeben, um zum Beispiel Denial of Service Attacken auszuführen.

b). How does a spoofing attack on DNS work?

Eine Möglichkeit ist die, dass der Angreifer einem Authoritative Name Server (ANS) kleine Anfragen schickt, die deutlich größere Antworten generieren. Im Beispiel des Papers schickt der Angreifer einen Request mit 50 Bytes, der eine 500 Bytes große Antwort zurückliefert. Indem er bei der Anfrage nun nicht seine Absender IP, sondern die eines Dritten einträgt, kann er durch Häufige Anfragen zum einen die Durchsatzrate von legitimen Anfragen des DNS Dienstes schmälern, zum anderen eine DoS gegen einen Dritten durchführen, da der DNS Server Antworten mit der zehnfachen Menge an Daten sendet, als der Upload des Angreifers zulassen würde. Dadurch kann der Downlink des Angegriffenen schnell erschöpft werden.

c). What are two potential goals of an attacker when performing a spoofing attack?

Wie oben beschrieben ist es möglich, damit eine IP Adresse mit sehr vielen Anfragen zu fluten, was einen DoS zur Folge haben kann. Über 'Cache poisining attacks' lassen sich außerdem DNS-Einträge verändern. Ein Angreifer versucht in dem Fall seine Seite so nachzubilden, dass keine Unterschiede zwischen der gefälschten, ausgelieferten Seite und der eigentlich angeforderten Seite ausgemacht werden können. Dabei versucht der Angreifer, im Hintergrund Viren und andere schädliche Inhalte an Besucher der Website zu verteilen.

d). How does DNS Guard work?

Die Idee des DNS Guard ist es, Cookies einzuführen, die eine Antwort des Anfragenstellers erzwingen, was eine Erkennung und Blockierung von Spoofing Angriffen ermöglicht. Zwei Fälle werden abgedeckt:

1. Es wird eine Verweisung (NS - Name Service Record) auf den Namen eines ANS's als Antwort einer Anfrage übergeben. Der Guard fängt in dem Fall die Anfrage ab und schickt die angefragte NS an den Anfragensteller zurück, mit einem vorne angehängten Cookie. Der Anfragensteller versucht nun diese NS aufzulösen, indem er die NS mit dem Cookie an den ANS schickt. Der Guard erkennt, dass der Cookie und die IP der Anfrage zusammenpassen und leitet es weiter an den ANS, der die IP der nächsten Domain zurückliefert.

2. Die Anfrage hat eine Adresse (A) als Antwort, die nicht mehr auf einen weiteren ANS's verweist. In dem Fall würde die Authentifikation mit dem Cookie dazu führen, dass das enifache Senden des A Eintrags vom Anfragensteller als weiterer ANS interpretiert wird. Deshalb wird der NS mit dem angehängten Cookie zusammen mit einem zweiten Cookie als IP Adresse zurückgeschickt. Der Anfragensteller schickt nun seine ursprüngliche NS ohne den

Cookie zusammen mit dem Cookie für die IP Adresse an den ANS, der nun die tatsächliche IP zurückliefert, die der LRS als A Eintrag interpretiert.

e). How do the authors evaluate their approach?

Zuerst werden mögliche Attacken behandelt und erklärt wie diese abgewehrt bzw. unschädlich gemacht werden können. Ein möglicher Angriff besteht darin, den DNS Server als Bandbreitenverstärker auszunutzen, jedoch tritt bei dieser Implementierung eine Bandbreitenverstärkungsrate von 50% auf, die diesen Angriff unschädlich machen. Darüber hinaus wird die Sicherheit des Cookies besprochen, der erraten werden kann. Dies ist für kleine Netzwerke weiterhin ein Problem, dass jedoch durch eine Anfragen-Limitierung entschärft werden kann. Eine weitere Möglichkeit einen Cookie zu generieren ist, den zugrunde liegenden Hashkey zu ermitteln. In dieser Implementierung wird MD5 benutzt, der zur Zeit der Publizierung des Papers bereits als kryptographisch unsicher galt. Außerdem wird noch die Möglichkeit eines Floodings des DNS guards mit falschen Cookies besprochen, der jedoch bei der Wahl eines schnellen Hashverfahrens sehr robust ist.

Darüber hinaus wird eine Performancevalidierung präsentiert. Genutzt wurde dazu ein ANS Simulator und ein LRS Simulator, bei denen das Verfahren mit UDP und TCP Verbindungen getestet wurden. Dieser Versuch wurde zweimal ausgeführt, einmal mit Attacken auf den DNS Guard und einmal ohne.