Universität Osnabrück

SEMINARARBEIT

zum Seminar

IT-Sicherheit

im Sommersemester 2013

Thema:

IPv6 Privacy Extensions

Erstellt am 10.05.2013

Vorgelegt von:

Kevin Seidel
943147
Falkenstraße 43
49124 Georgsmarienhütte

Inhaltsverzeichnis

1 Einleitung

Diese Arbeit beschäftigt sich mit dem Nutzen und der Funktionsweise der Privacy Extensions im Internet Protocol Version 6.

2 Das Internet Protocol Version 6

2.1 Warum IPv6?

Durch den starken Anstieg von Netzwerkgeräten in den vergangen Jahren, ist es nötig geworden, einen größeren Adressraum zu schaffen. Das Internet Protocol (IP) Version 4 stellt, aufgrund der Adressgröße von 32 Bits, nur einen Adressraum von 2^{32} Adressen zur Verfügung, was in etwa 4,3Mrd. Adressen entspricht. Da diese jedoch mittlerweile alle vergeben sind, ist es notwendig geworden einen größeren Adressraum zu schaffen.

IP Version 6 setzt deshalb auf eine Adresslänge von 128 Bits, wodurch sich der Adressraum enorm vergrößert (etwa $3.4 * 10^{38}$ Adressen).

Eine weitere Veränderung von IP Version 6(IPv6) gegenüber IP Version 4 (IPv4) ist ein Header mit festgelegter Größe, welcher die zwingend notwendigen Informationen enthält. Für zusätzliche Informationen gibt es einen Extention Header, welcher jedoch nur bei Bedarf genutzt wird.

2.2 Aufbau einer IPv6-Adresse

Die IPv6-Adresse unterscheiden sich deutlich von den bisher verwendeten IPv4-Adressen. IPv4-Adressen haben eine Größe von 32 Bits und werden meist in der "dotted decimal notation", das heißt in vier Blöcken von Dezimalzahlen zwischen 0 und 255, welche durch einen Punkt separiert werden, dargestellt. Dadurch lassen sich 2^{32} (ca. 4,3 Mrd.) verschiedene Adressen darstellen. Durch die Verwendung von IPv6-Adressen erhöht sich die Anzahl der Adressen drastisch, da man hier eine Adresslänge von 128 Bits verwendet, womit die Größe des Adressraumes auf 2^{128} angehoben wird. Da eine Darstellung dieser Adressen in "dotted decimal notation" aus 16 Blöcken bestehen würde und damit sehr schwer zu lesen wäre, entschloss man sich dazu die IPv6-Adressen in 8 Blöcken zu je 4 Hexadezimalziffern zusammenzufassen. Diese Blöcke werden, durch einen Doppelpunkt getrennt, notiert.

$$2001 : 0db8 : 1aAa : 0000 : CCcc : 0000 : 0000 : 0D01$$

Abbildung 1: Beispiel einer IPv6-Adresse.

Da diese Adressen im Vergleich zu IPv4-Adressen immernoch relativ lang und unübersichtlich sind, gibt es mehrere Möglichkeiten eine IPv6-Adresse zu verkürzen. So wird in [**?**] vereinbart, dass man ein oder mehrere aufeinanderfolgende Blöcke, welche nur Nullen beinhalten durch "::" verkürzen kann. Dies jedoch nur einmal pro Adresse. Ausserdem ist es möglich auf führende Nullen innerhalb eines Blockes zu verzichten.

In [**?**] wird eine Empfehlung für eine etwas striktere Darstellung von IPv6-Adressen gemacht. So ist es dort vorgeschrieben innerhalb von Adressen nur Kleinbuchstaben zu verwenden. Dies dient der besseren Lesbarkeit und verhindert eine versehentliche Verwechselung von 8 und B sowie 0 und D, die bei gemischter Groß- und Kleinschreibung oder durchgängiger Großschreibung entstehen könnte.

Des Weiteren müssen führende Null innerhalb eines Blockes ausgelassen werden und Blöcke, welche nur aus Nullen bestehen durch eine einzelne Null repräsentiert werden.

Ausserdem ist es nun nicht mehr erlaubt einzelne Null-Felder durch "::" zu repräsentieren. Diese Schreibweise ist nurnoch auf mehrere aufeinanderfolgende Null-Felder anwendbar und muss in dem Fall auch genutzt werden. Gibt es mehrer Möglichkeiten Felder

durch "::" zu verkürzen, muss das mit dem größten Nutzen, das heißt mit den meisten aufeinanderfolgenden Nullen, gewählt werden. Gibt es mehrere gleichgroße Möglichkeiten, ist die erste zu wählen.

$$2001 : db8 : 1aaa : 0 : cccc :: d01$$

Abbildung 2: Beispieladresse aus Abb. 1 unter Berücksichtigung der Empfehlung

Durch die Befolgung dieser Regeln zu Darstellung von IPv6-Adressen in Textform hat sich die Lesbarkeit stark verbessert.

Die Netzmasken werden bei IPv6 durch Suffixe dargestellt, das heißt die Anzahl der Einsbits in der Netzmaske wird, abgetrennt durch einen Schrägstrich, hinten an die Adresse angehangen. Eine Notation der Netzmaske als Adresse, wie in IPv4 üblich, ist nun nichtmehr möglich.

$$2001 : db8 : 1aaa : 0 : cccc :: d01/64$$

Abbildung 3: Beispieladresse aus Abb. 2 mit Netzmaske

Durch diese Netzmaske wird angegeben, welcher Teil der Adresse das Netzwerk identifiziert und welcher Teil das Interface repräsentiert. Der vordere Teil der Adresse, auch Präfix genannt, dient dabei zur Bestimmung des Netzes. Der hintere Teil, Interface Identifier genannt, dient zur Indentifizierung eines Gerätes im Netz. Beim Beispiel in Abbildung 3 wäre der Präfix 2001:db8:1aaa:0::/64 und der Interface Identifier ::cccc:0:0:d01/64.

2.3 Vergabe von IPv6-Adressen

Bei der Vergabe von IPv6-Adressen in einem Netzwerk gibt es mehrere Vorgehensweisen. Zum einen die manuelle Vergabe von Adressen, was in kleinen Netzen schnell und einfach funktioniert, bei größeren Netzen jedoch einen enormen Aufwand bedeutet. Eine Alternative für größere Netze wäre die Verwendung des Dynamic Host Configuration Protocols (DHCP), welche schon für IPv4 existiert. Für IPv6 gibt es eine angepasste Version mit dem Namen DHCPv6, welches, identisch zum klassischen DHCP, dynamisch Adressen an m Netzwerk befindliche Interfaces verteilt und die Adressen auf dem DHCP-Server speichert. Durch die Speicherung der Adressen handelt es sich um eine Stateful Address Configuration im Gegensatz zur letzten Möglichkeit, der Stateless Address Autoconfiguration. Diese erzeugt IPv6-Adressen direkt auf dem Gerät, welches diese später verwendet. Die Funktionsweise wird im Folgenden näher erörtert.

3 Stateless Address Autoconfiguration

Die Stateless Address Autoconfiguration(SLAAC) ist ein Verfahren zur eigenständigen Erzeugung von IP-Adressen von Netzwerkgeräten. Dies ermöglicht eine einfache und konfigurationslose Einrichtung von Netzwerken, da zum Beispiel auf manuelle Adressvergabe oder den Einsatz von Dynamic Host Configuration Protocol(DHCP)-Servern verzichtet werden kann. Selbst bei mehreren verbundenen Netzwerken ist ein DHCP-Server nicht nötig, da mittels Router Advertisement die einzelnen Adressen für die Subnetze vergeben werden können.

3.1 Funktionsweise der Stateless Address Autoconfiguration

Für die Erzeugung einer globalen IPv6-Adresse sind mehrere Schritt nötig, welche im folgenden genauer erörtert werden. Der Autokonfigurationsprozess wird mit der Aktivierung des Netzwerkgerätes gestartet. Dazu wird zuerst eine link-local Adresse erstellt. Diese erlaubt grundlegende Kommunikation im Netzwerk. Sie setzt sich aus einem festgelegtem Präfix (fe80::/64) und dem gerätespezfischen Interface Identifier zusammen. Die Interface Identifier wird in der Regel aus der Media Access Control(MAC)-Adresse des Netzwerkgerätes gebildet. Um diese 48 Bits lange MAC-Adresse auf die für den Interface Identifier vorgesehene Länge von 64 Bits zu bringen, wird die MAC-Adresse in der Mitte geteilt und dort wird der Wert "FF:FE" eingesetzt. Ausserdem wird das siebte Bit von links invertiert, welche angibt, ob die MAC-Adresse global oder lokal administriert wird.

Bevor diese vorläufige Adresse, egal ob manuell zugewiesen oder durch DHCP oder SLAAC, an ein Interface gebunden wird, muss deren Eindeutigkeit im Netzwerk überprüft werden. Die geschieht mittels Duplicate Address Detection. Einzige Ausnahmen eine Duplicate Address Detection sind Anycast Adressen oder explizite Deaktiverierung des Vorgangs. Zur Überprüfung der Eindeutigkeit von Adresse werden sogenannte Neighbor Solicitations und Neighbor Advertisements verwendet. Neighbor Solicitations haben als Empfänger unsere generierte vorläufige Adresse und als Sender die nicht spezifizierte Adresse. Um eine Antwort auf diese Nachricht zu erhalten ist es noch nötig, dass das Interface zunächst dem all-nodes Multicast, welcher einem Broadcast in IPv4 entspricht und an alle angebundenen Adressen verschickt, und dem solicited-node Multicast, welcher speziell für die Duplicate Address Detection zuständig ist, beitreten. Falls nun schon ein Interface diese vorläufige Adresse besitzt, erhält es die Neighbor Solicitation und antwortet mit einem Neighbor Advertisement. Unser Interface empfängt dieses Neighbor Advertisement, wodurch es weiß, dass die vorläufige Adresse nicht eindeutig ist und damit nicht an das Gerät gebunden werden kann.

Falls das Interface während der Duplicate Address Detection eine Neighbor Solicitation von einem anderen Gerät erhält, versuchen zwei Geräte gleichzeitig die gleiche Adresse zu verwenden. In diesem Fall sollte keines der Geräte seine vorläufige Adresse an das Gerät binden. Es kann jedoch auch vorkommen, dass die Neighbor Solicitation von unserem Gerät kommt, falls der Multicast die Packete auch an den Sender zurückschickt.

Wenn die Duplicate Address Detection fehlschlägt und unser Interface Indentifier aus der Hardware-Adresse gebildet wurde, sollten alle IP Operationen eingestellt werden.

Nachdem die link-local Adresse gebildet wurde, ist es nun möglich eine globale Adresse zu generieren. Dies geht jedoch nur, falls sich ein Router im Netzwerk befindet, da hierfür sogenannte Router Advertisements nötig sind. Die Router Advertisements enthalten Informationen über das Netzwerk, wie zum Beispiel den zu verwendenden Präfix. Sie werden

vom Router automatisch in periodischen Abständen geschickt, können jedoch auch durch Router Solification angefragt werden.

Ein Gerät sendet also zuerst eine Router Solification und wartet auf ein Router Advertisement als Antwort. Nach einem vorgegebenen Zeitraum ohne Antwort wird dies nocheinmal ausgeführt. Dies geschieht so lange bis ein Router antwortet oder die vorgegebene Maximalanzahl von Anfragen erreicht ist. Antwortet der Router, wird aus dem in dem Router Advertisement enthaltenen Präfix und dem vom Gerät vorher gebildeteten Interface Identifier eine globale Adresse gebildet.

3.2 Probleme der Stateless Address Autoconfiguration

Ein Problem bei der Stateles Address Autoconfiguration entsteht durch die Verwendung eines konstanten Interface Identifiers für die IP-Adresse. Da der Interface Identifier in den meisten Fällen aus der MAC-Adresse generiert wird, bleibt dieser selbst bei einem Wechsel des Netzwerkes konstant und lässt sich somit dauerhaft einem Gerät zuordnen. Das heißt, selbst wenn der Präfix der IPv6-Adresse wechselt, lässt sich das Gerät, und damit meist auch der Nutzer, über den Interface Identifier mit sehr großer Trefferwahrscheinlichkeit zurückverfolgen. So lässt sich beispielsweise mittels eines Netzwerksniffers bestimmen, wann ein Gerät kommuniziert hat, mit wem es kommuniziert hat und ,durch den Präfix der IPv6-Adresse, in welchem Netz es sich befand. Besonders problematisch ist dies bei mobilen Geräten, wie zum Beispiel Smartphones oder Laptops, welche oft ihren Standort ändern. So können deren Bewegungen sehr leicht verfolgt werden, da sich zwar der Präfix der Adresse bei einem Netzwechsel ändert, der Interface Identifier jedoch konstant bleibt.

Ein weiters Problem was bei der Nutzung von MAC-Adressen als Interface Identifier entsteht, ist die dadurch entstehende Möglichkeit, die Hardware des Gerätes zu bestimmen. Da die MAC-Adressen eindeutige Herstellerkennungen enthalten, lässt sich die darunterliegende Hardware leichter bestimmen und dadurch gezielt angreifen. So kann man beispielsweise anhand der MAC-Adresse Apple Geräte erkennen und in Folge dessen gezielt angreifen.

Ein Ansatz zur Lösung des Problems geben die Privacy Extentions für IPv6.

4 IPv6 Privacy Extensions

Durch den Einsatz der IPv6 Privacy Extensions sollen die voher beschriebenen Probleme der Stateless Address Autoconfiguration gelöst werden. Dies geschieht dadurch, dass der Interface Identifier nicht mehr nur aus der bearbeiteten MAC-Adresse besteht, und damit über die gesamte Lebensdauer des Gerätes gleich bleibt, sonder dynamisch erzeugt wird. Dadurch wird eine Zuordnung von einer IPv6-Adresse zu einem bestimmten Gerät nahezu unmöglich und die vorher beschriebenen Probleme mit der Privatsphäre werden gelöst.

Da durch die Privacy Extentions jedoch nur der Interface Identifier geändert wird, ist die Gefahr einer konstanten Zuordnung einer Adresse zu einem Gerät nicht komplett gelöst. Jedes Netzwerk bekommt seinen eigenen Präfix, welcher ebenfalls konstant sein kann. So lässt sich bei kleinen Netzwerken, bestehend aus wenigen oder sogar nur einem Gerät, eine relativ genaue Zuordnung der Adresse zu einem Gerät machen und das obwohl der Interface Identifier ständig wechselt. Bei Adressen, welche über einen Domain Name System(DNS) - Namen verfügen, ist eine Verwendung der Privacy Extentions ebenfalls nutzlos, falls auch der DNS-Name gleich bleibt. Da über den DNS-Namen eine eindeutige Zuordnung möglich ist, hat der wechselnde Interface Identifier keinerlei Vorteile. Es gibt jedoch auch Geräte, welche sowohl als Client, als auch als Server aggieren. Hier ist es möglich dem Gerät zwei IPv6-Adressen zuzuweisen. So kann die Serveradresse im DNS eingetragen sein und ist dadurch öffentlich erreichbar. Der Client hingegen bekommt eine private IPv6-Adresse, auf welche die Privacy Extentions angewand werden. Dabei muss jedoch gegeben sein, dass man keinerlei Zusammenhang zwischen den beiden Adressen herstellen kann. Es ist daher nötig bei der Generierung eine Zufallsvariable mit einzubeziehen, um dies zu verhindern.

4.1 Einsatz der Privacy Extentions

Das Ziel der Privacy Extensions ist es, bei erhöhter Sicherheit und Privatsphäre, die gleiche einfache Handhabung und automatische Adressegenerierung wie bei der Stateless Address Autoconfiguration zu erhalten. Es soll somit ermöglicht werden, aus dem zufällig erstellten Identface Identifier verschiedene Adressen, mit unserschiedlichen Präfixen, zu erstellen. Eine andere Möglichkeit ist die Generierung eines spezifischen Interface Identifiers für jeden Präfix, um die keinen Zusammenhang zwischen den einzelnen Adressen zu erstellen. Dies führt jedoch möglicherweise zu einer Performancereduzierung. Aus diesem zufälligen Interface Identifier sollen sich zudem keine zukünftigen oder vergangenen Interface Identifier ableiten oder berechnen können.

4.2 Generierung des zufälligen Interface Identifiers

Bei der Generierung des randomisierten Interface Identifiers gibt es zwei verschiedene Vorgehensweisen. Die erste Vorgehensweise setzt einen persistenten Speicher für die Erzeugung des Interface Identifiers vorraus. Diese Variante verzichtet auf die Generierung eine Zufallszahl bei der Erzeugung eines neuen Interface Identifiers. Hier muss nur bei erster Inbetriebnahme eine Zufallszahl generiert werden.

Die zweite Methode wird bei nicht vorhandenem Speicher eingesetzt. Dort ist es nötig bei jeder neuen Erzeugung eines Interface Identifiers, auch eine neune Zufallszahl generieren.

Bei einem Gerät welche nach der ersten Methode vorgeht und somit über persistenten Speicher verfügt, generiert bei der ersten Inbetriebnahme eine 64 Bit Zufallszahl. Diese sollte von hoher Qualität und damit schwer zu erraten sein. Bei dieser Methodik muss immer ein 64 Bit history value vorhanden sein. Dieses wird entweder, wie oben beschrieben, bei Erst-Start generiert oder wurde aus der vorherigen Erzeugung eines Interface Identifiers gespeichert.

Der eigentliche Vorgang der Erzeugung des Interface Identifiers beginnt damit, dass man das history value an den durch die Stateless Address Autoconfiguration erzeugten Interface Identifier anhängt. Hiervon wird nun der Message-Digest Algorithm 5 (MD5)-Hash erzeugt, welcher 128 Bit lang ist. Es ist jedoch auch möglich andere Hashingverfahren zu verwenden. Für den Interface Identifier werden die linken 64 Bit verwendet und das siebte Bit wird auf 0 gesetzt, um die Adresse als lokal zu setzen. Wie bei der Stateless Address Autoconfiguration werden auch hier die Adressen auf ihre Einzigartigkeit hin getestet. Falls der Interface Identifier in einem reservierten Bereich liegt oder schon verwendet wird, ist es nötig ihn erneut zu generieren. Dies geschieht unter Verwendung der rechten 64 Bit des zuvor errechneten MD5-Hashes als history value, welche an den Interface Identifier angehängt wird. Falls kein Konflikt vorherscht werden die rechten 64 Bit des Hashes als history value auf den persistenten Speicher geschrieben.

Das Verfahren mit der Einbeziehung des statischen Interface Identifiers wird deshalb gewählt, damit es durch gleiche Zufallszahl-Generatoren nicht zu ständigen Konflikten kommt, falls zwei Geräte immer wieder die selben Zufallszahlen generieren.

Die zweite Methode, welche bei der Absenz von Speicher zur Anwendung kommt, wird ähnlich vorgehen. Statt jedoch auf das history value zurückzugreifen, ist es nötig bei jeder neuen Generierung eines Interface Identifiers auch eine neue Zufallszahl zu generieren. Danach kann das Verfahren wie aus der ersten Methode angewand werden.

5 Mögliche Verbesserungen der Privacy Extentions

6 Fazit

7 Quellen

RFC 5952 RFC 3513 RFC 4941 RFC 4291 https://supportforums.cisco.com/docs/DOC-24485