
Zugehörigkeit von
MAC- & IPv6-Adressen speichern
TÜVRheinland®
Genau. Richtig.

Vorstellung

- **Johannes Weber**
- **Master IT-Sicherheit**
- **Berater für Netzwerksicherheit TÜV Rheinland i-sec GmbH**
  - Firewall & VPN
  - Web Application Firewall
  - Mail
  - Proxy
- **IPv6 Security**

- Masterarbeit über IPv6 Security:
  http://blog.webernetz.net/2013/05/06/ipv6-security-master-thesis/
- Xing:
  https://www.xing.com/profile/Johannes_Weber65

Agenda

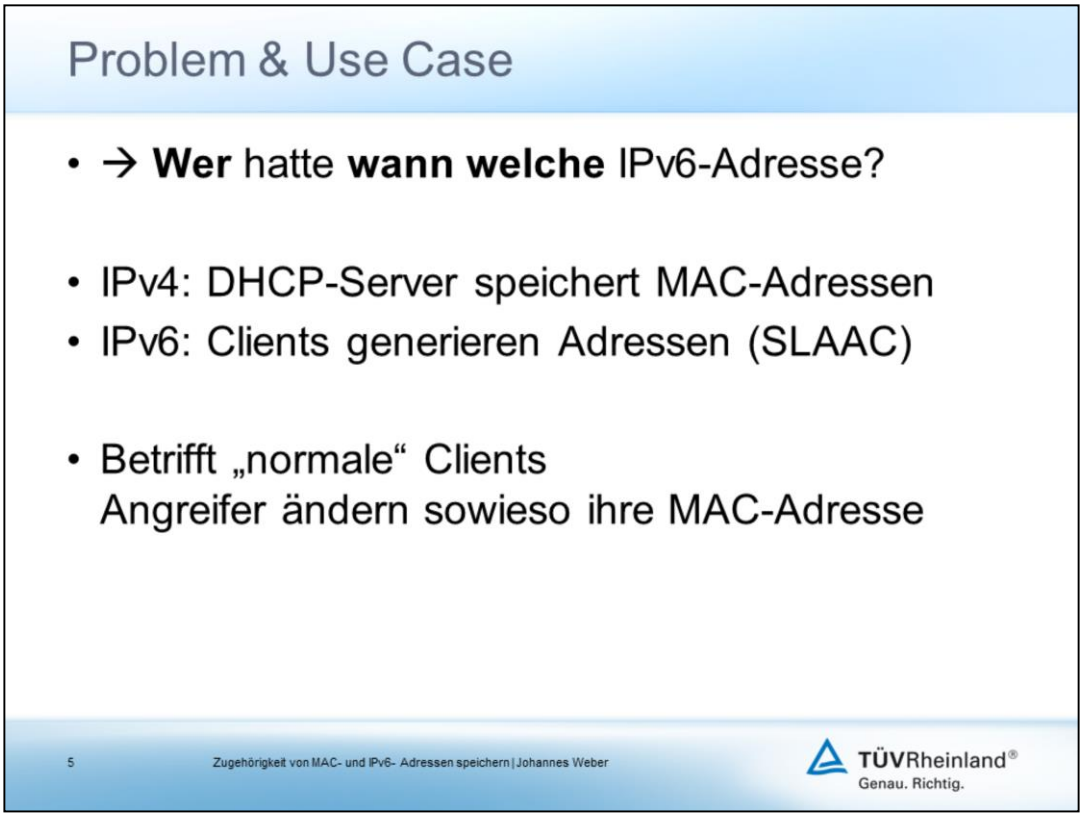
- Problem & Use Case
- Ansatz: Speichern der DAD-Nachrichten
- Tcpdump Skript & Auswertung

- Test: Verlässlichkeit der DAD-Nachrichten
- Adressstatistiken

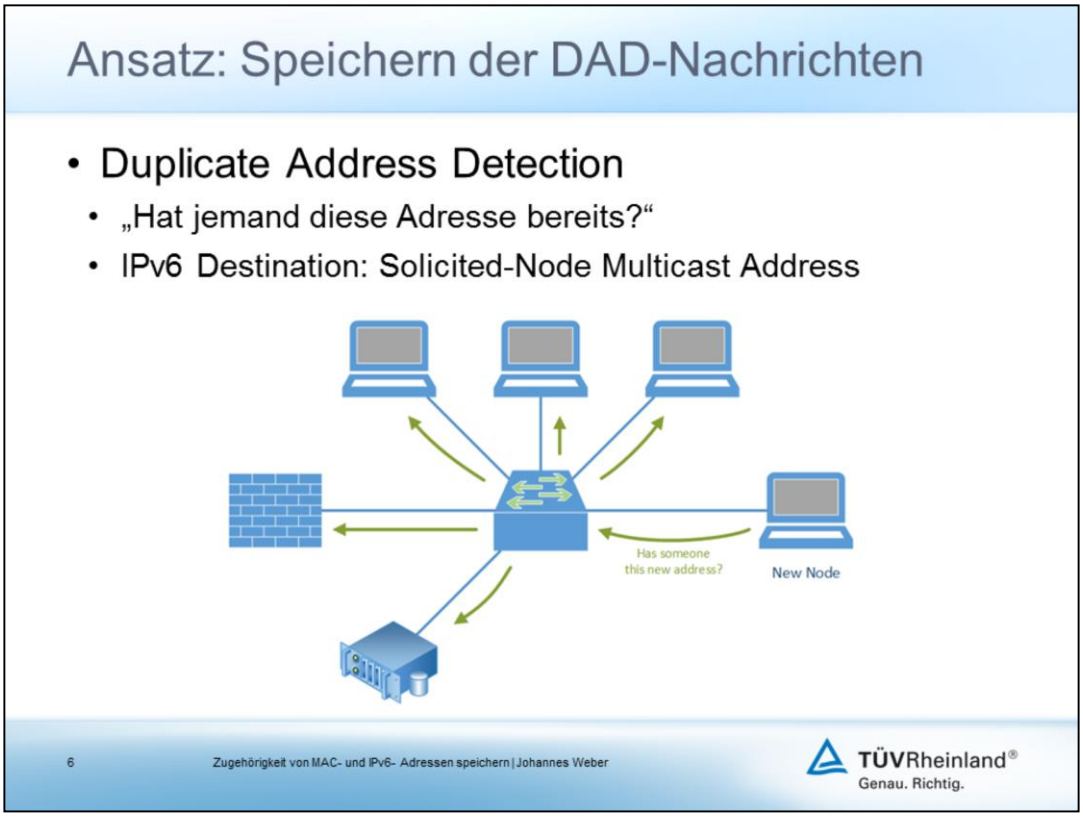
- Nachteile und Alternativen
- Fazit

Einstiegsfragen

1. IPv6 hat keine Broadcasts mehr. **Daher werden Multicast-Pakete von einem Switch immer nur an die notwendigen Ports geschickt.**
   - Falsch! Einfache Switches leiten alle Multicast-Pakete an ALLE Switchports weiter!
2. Duplicate Address Detection Nachrichten müssen immer gesendet werden, **egal ob die IPv6-Adresse selbst generiert oder manuell vergeben worden ist.**
   - Richtig.



- Kernproblem: Wer hatte wann welche IPv6-Adresse? Zum Beispiel für folgende Szenarios:
  - Zugriff innerhalb des internen Netzes auf eine unerlaubte Ressource
  - Illegales Verhalten im Internet
  - **Malware Callback**
  - → Forensische Analyse: Wer hatte die Adresse 2001:db8:8d2a:1090:903d:81ad:64f6:1199 ?
- In manchen Netzen ist DHCPv6 nicht umzusetzen, wenn beispielsweise nicht alle Clients DHCPv6 sprechen.
- Live vs. Log: Wenn man live nach einer Adresse sucht mag das zwar klappen. Aber ein Eintrag im Log ist immer besser, weil er direkt am Anfang einer Session dort gespeichert wird und auch nicht mehr verändert werden kann. (Außer wenn man den Log-Server direkt angreift…)
- **Ein Angreifer ändert sowieso seine MAC-Adresse, antwortet gar nicht auf DAD-Nachrichten, oder oder oder.** → Das Speichern der MAC-IPv6 Bindings ist also nur für legitime Clients gedacht, bei denen sich bspw. der Benutzer falsch verhält.



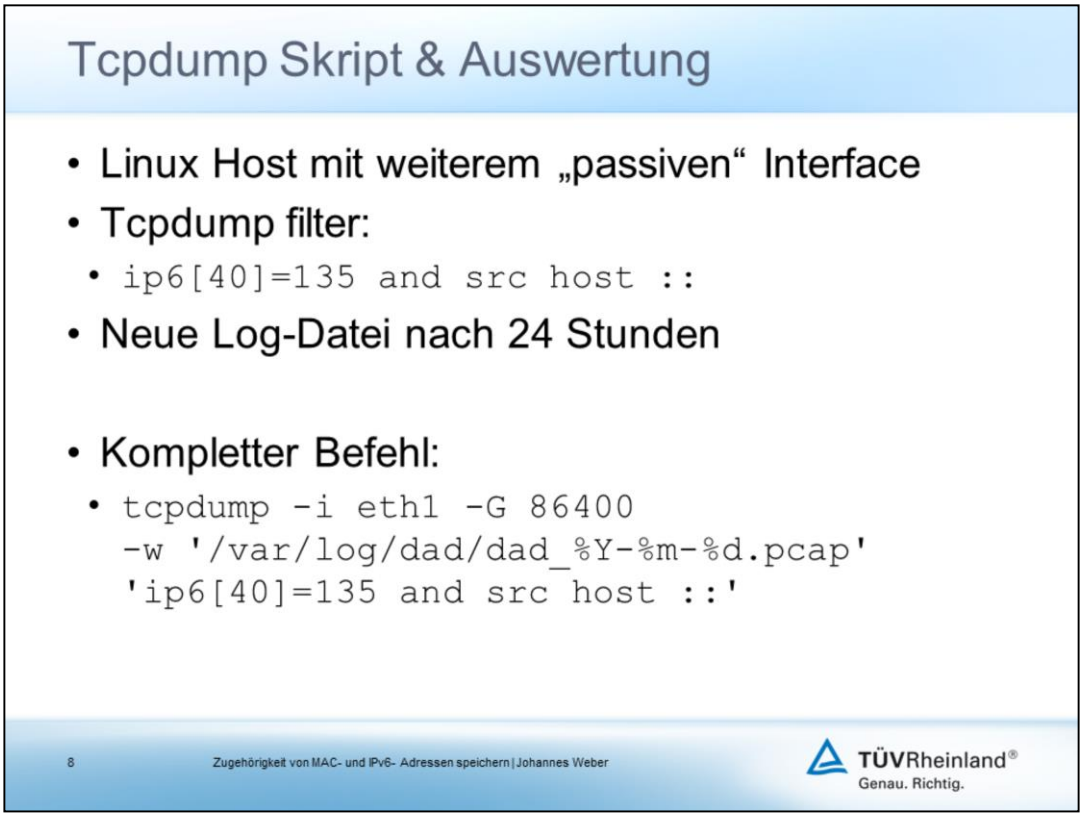
- Mein Tipp zum Lernen: In Wireshark die ersten IPv6 Pakete von einem Client genau anschauen, also ab dem Zeitpunkt, an dem man ihn ans Netzwerk hängt: Man sieht sehr schön, welche Solications er schickt, welche Multicast Gruppen er joinen möchte, etc.
- **Wenn ein Rechner eine neue IPv6-Adresse benutzen möchte, schickt er zunächst eine Duplicat Address Detection an die Solicited Node Multicast Adresse seiner selbst erzeugten IPv6-Adresse (also quasi „an sich selbst“) um zu sehen, ob nicht schon ein anderer Host diese IPv6-Adresse verwendet.**
- Das müssen auch manuell vergebene IPv6-Adressen tun. (RFC 4862, Section 5.4, „MUST be performed on all unicast […]”)
- Die *meisten* einfachen Switches leiten diese DAD-Nachricht an alle Ports weiter, da sie keine Listen über die Multicast Listener pro Port führen. **Sprich: Obwohl es ein Multicast Paket ist, leitet der Switch es an ALLE seiner Ports weiter.**
- Im Normalfall antwortet NIEMAND auf diese DAD-Nachricht, da die IPv6-Adresse entweder anhand der MAC-Adresse gebaut wurde (EUI-64), randomisiert ist (Windows) oder per Privacy Extensions erzeugt wurde. In allen Fällen sollte eine Dopplung nicht vorkommen. Ausnahme: Bei einer manuellen Vergabe von IPv6-Adressen kann es passieren.
- **Ansatz: Ein Server im Layer-2 Netz empfängt und speichert alle eintreffenden DAD-Nachrichten. Dort steht die sendende MAC-Adresse sowie die Target IPv6-Adresse drin.**

Ansatz: Speichern der DAD-Nachrichten

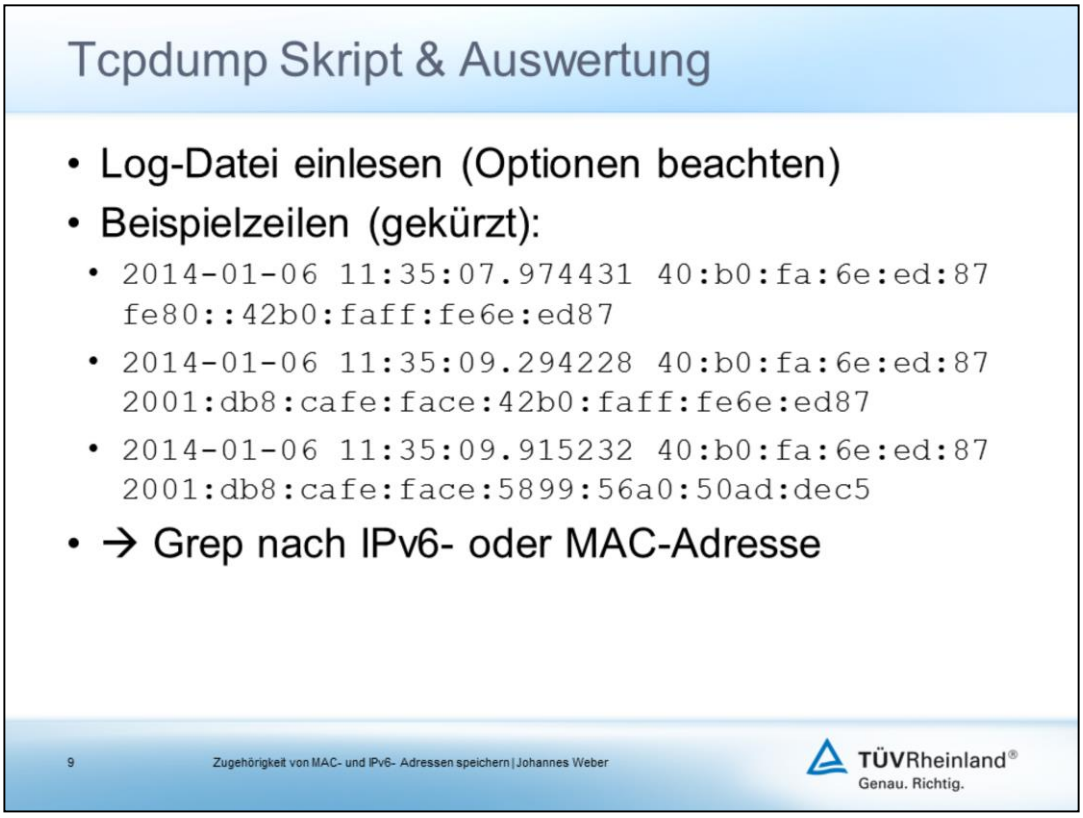
- Neighbor Solicitation (135) mit Source „::“
- Wireshark Screenshot:

```
⊞ Frame 65: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
⊟ Ethernet II, Src: Apple_4f:98:f4 (b8:ff:61:4f:98:f4), Dst: IPv6mcast_ff:4f:98:f4 (33:33:ff:4f:98:f4)
  ⊞ Destination: IPv6mcast_ff:4f:98:f4 (33:33:ff:4f:98:f4)
  ⊞ Source: Apple_4f:98:f4 (b8:ff:61:4f:98:f4)
    Type: IPv6 (0x86dd)
⊟ Internet Protocol Version 6, Src: :: (::), Dst: ff02::1:ff4f:98f4 (ff02::1:ff4f:98f4)
  ⊞ 0110 .... = Version: 6
  ⊞ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 24
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: :: (::)
    Destination: ff02::1:ff4f:98f4 (ff02::1:ff4f:98f4)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊟ Internet Control Message Protocol v6
    Type: Neighbor Solicitation (135)
    Code: 0
    Checksum: 0x2ea0 [correct]
    Reserved: 00000000
    Target Address: fe80::baff:61ff:fe4f:98f4 (fe80::baff:61ff:fe4f:98f4)
```


- Die DAD-Nachrichten sind eindeutig bestimmbar, da es nur und immer „Neighbor Solicitations“ mit einer Source IPv6-Adresse von „::“ sind
- Ethernet Frame Source: **MAC-Adresse des Clients**
- Target Address in der Neighbor Solicitation: Zukünftige **IPv6-Adresse des Clients** (falls niemand auf die DAD-Nachricht antwortet)



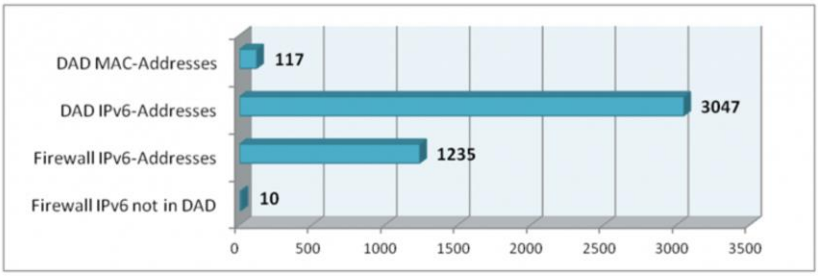
- Der Linux Host hat im besten Fall ein weiteres physikalisches Interface, auf dem NUR gelauscht wird, jedoch kein eigener Netzwerk-Verkehr drüber läuft.



- Die genauen Befehle usw. sind in meinem ersten Blog Post dokumentiert: http://blog.webernetz.net/2014/03/17/monitoring-mac-ipv6-address-bindings/
- **Beispiel hier: Der gleiche Client (MAC-Adresse) hat sich drei IPv6-Adressen erzeugt:** Eine link-local (fe80::) und eine global unicast Adresse, welche beide die gleiche Interface-ID haben die über EUI-64 erzeugt wurde, sowie eine global unicast IPv6-Adresse die zufällig aussieht, also anhand den Privacy Extensions erzeugt wurde.
- Den zugehörigen Client (User) zur MAC-Adresse sucht man dann im Inventar.

Verlässlichkeit der DAD-Nachrichten

- Vergleich DAD-Log mit Firewall-Log
- 1 Monat - BYOD-WLAN: Zoo an Geräten

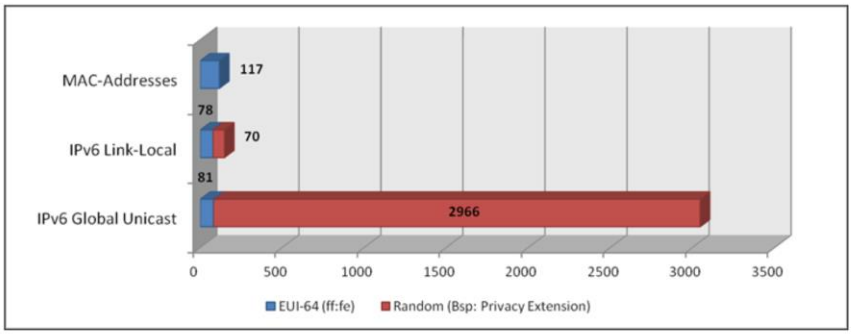


- 10/1235 IPv6-Adressen fehlten: **0,8 %**
- Worst Case: 10/127 MACs fehlten: **7,8 %**

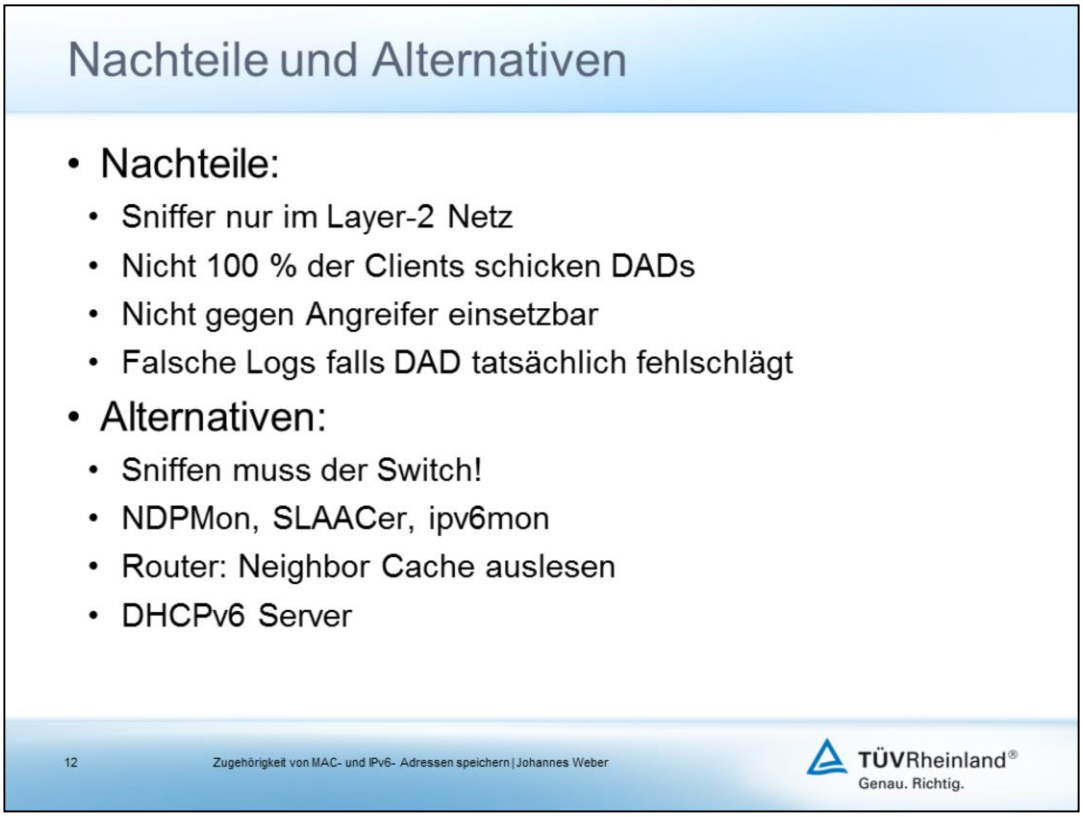
- Zugehöriger Blog Post: http://blog.webernetz.net/2014/04/24/reliability-of-ipv6-dad-message-sniffing/
- Firewall: Cisco ASA 5520, Syslog auf „Debugging“
- Geräte sind: Windows (diverse Versionen), Mac, Smartphones (Android, iPhone)
- **Ergebnis: 10 von 1235 IPv6-Adressen haben Verbindungen durch die Firewall aufgebaut und waren NICHT im DAD-Log. Macht prozentual 0,8 %.**
- **Wenn alle fehlenden Adressen von verschiedenen Clients aufgebaut wurden (Worst Case), dann ist die Fehlerrate 10 von 127 MAC-Adressen = 7,8 %.** Man kann aber vermuten, dass die 10 fehlenden IPv6-Adressen von weniger als 10 verschiedenen Geräten stammten.

Adressstatistiken

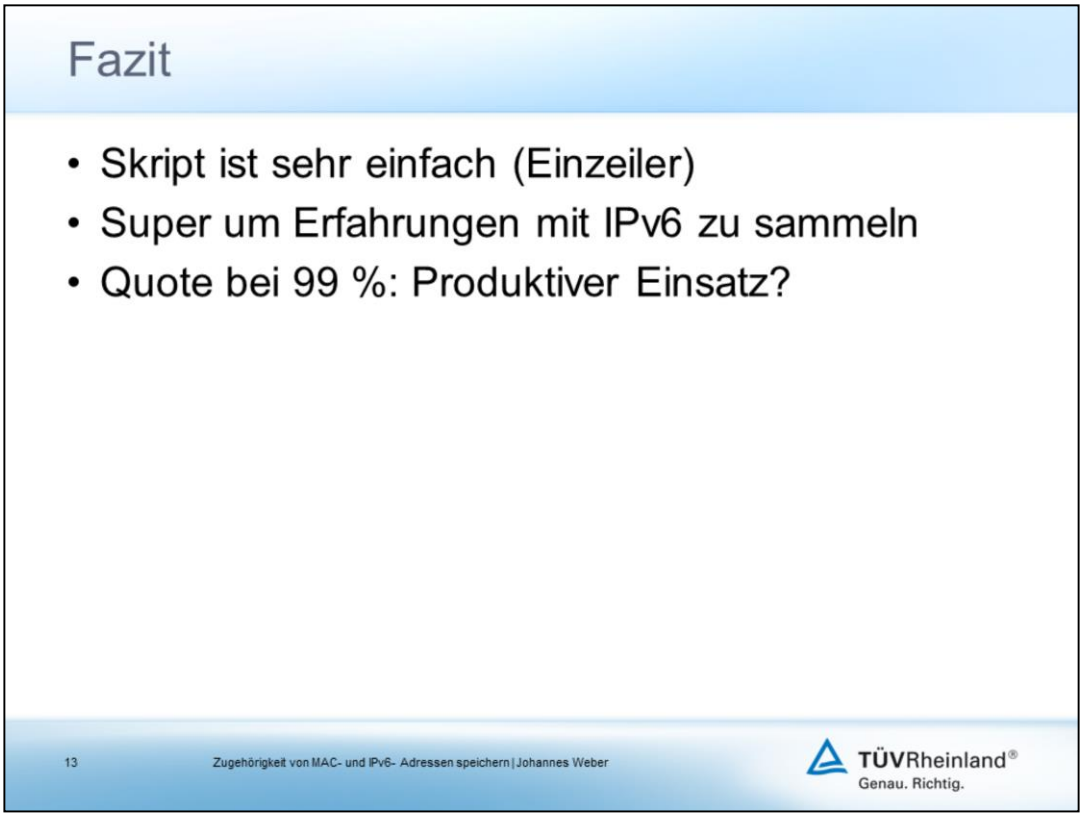
- Auswertung DAD-Log per Skript
  - Einfache Linux Tools (sed, sort, uniq, grep)
  - Ausgabe: Anzahl der Adressen



- Skript verfügbar auf meinem Blog: http://blog.webernetz.net/2014/05/02/ipv6-address-statistics-based-on-dad-messages/
- Das Skript gibt die Anzahl der Adresse aus. Hier in der Präsentation ist es zusätzlich grafisch aufbereitet.



- Der hier vorgestellte Ansatz speichert **nur die DAD-Nachrichten aus dem Layer-2 Netz, in dem er hängt**. Für mehrere Netze bräuchte man jeweils einen neuen Sniffer.
- Der Test hat gezeigt: Nicht 100 % aller IPv6-Adresse werden gespeichert.
- Falsche Logs falls DAD fehlschlägt: Wenn PC 2 eine IPv6-Adresse anfragt die PC 1 bereits besitzt, so speichert das Tcpdump Skript diese IPv6-Adresse mit der MAC-Adresse von PC 2 (!) ab, obwohl er sie gar nicht bekommt. Da der Logeintrag von PC 1 ebenfalls vorhanden sein sollte, kann man diesen Fehler zumindest erkennen. Man muss in diesem Fall also aufpassen.
- **Sniffen muss der Switch**, da er 100 % des Verkehrs sieht. Das heißt, er kann zu jedem IPv6-Paket die MAC-Adresse des Ethernet Frames auswerten. Die entsprechenden Features heißen unter anderem: **IPv6 Guard, IPv6 Snooping, IPv6 ND Inspection, …**
- NDPMon: http://ndpmon.sourceforge.net/
- SLAACer: http://www.digriz.org.uk/slaacer
- ipv6mon: http://www.si6networks.com/tools/ipv6mon/
- **Router Neighbor Cache:** Automatisches Auslesen des Caches in kürzeren Intervallen als das automatische Entfernen von Einträgen. Nachteil: Enthält keine Einträge für Kommunikationspartner innerhalb eines Subnetzes.



- Einfach in jedes Segment einen Raspberry Pi reinpacken ;)
- Ob es ernsthaft im produktiven Umfeld genutzt werden sollte, bleibt jedem selbst überlassen. Im Zweifelsfall möchte man wohl eher eine *komplette* Übersicht über das eigene Netzwerk von einem kommerziellen Anbieter.

Quellen

- Drei Blog Posts:
  - Monitoring MAC-IPv6 Address Bindings:
    http://blog.webernetz.net/2014/03/17/monitoring-mac-ipv6-address-bindings/
  - Reliability of IPv6 DAD Message Sniffing:
    http://blog.webernetz.net/2014/04/24/reliability-of-ipv6-dad-message-sniffing/
  - IPv6 Address Statistics based on DAD Messages:
    http://blog.webernetz.net/2014/05/02/ipv6-address-statistics-based-on-dad-messages/
- RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
- RFC 4862: IPv6 Stateless Address Autoconfiguration

- Bei Fragen: Einfach mailen
- Mein Blog: Infos/Tutorials/Erfahrungen im Bereich Netzwerk-Sicherheit, Firewalls, VPN, Passwörter, etc.

VIELEN DANK FÜR IHRE AUFMERKSAMKEIT!

Regelmäßig aktuelle Informationen im
Newsletter und unter **www.tuv.com/informationssicherheit**