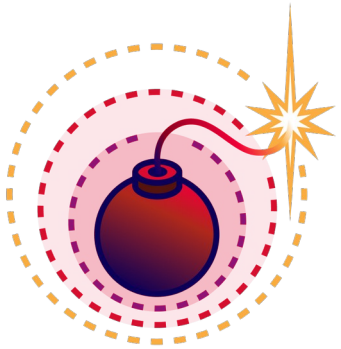


Blast-RADIUS – Hintergrund und Ausblick CVE-2024-3596

81. DFN-Betriebstagung | 9. Oktober 2024

Jan-Frederik “Janfred” Rieckers

Die Blast-RADIUS-Verwundbarkeit

- Entdeckt von Forschenden an der Boston University, UC San Diego u.a.
- Verwundbarkeit erlaubt es Angreifer:innen mit vollem Zugriff auf die Kommunikation, eigene RADIUS-Pakete mit korrektem RADIUS Authenticator zu senden → RADIUS-Clients würden diese Pakete akzeptieren

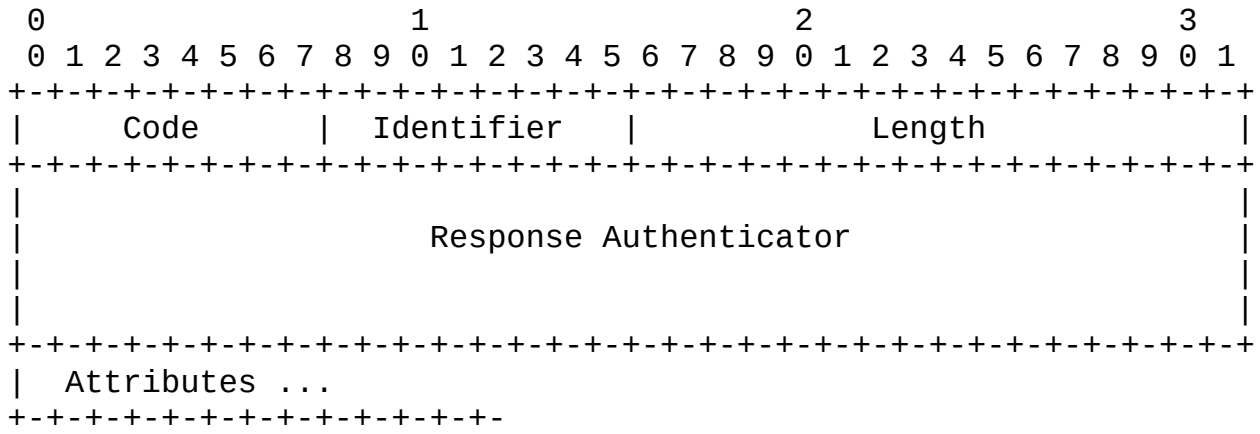
Spoiler

- Verwundbarkeit existiert nur bei RADIUS/UDP bzw. RADIUS/TCP, nicht bei RADIUS/(D)TLS
  - Angriff allerdings möglich, sobald ein Glied in der Proxy-Kette RADIUS/UDP nutzt und verwundbar ist
- eduroam ist nicht verwundbar, selbst wenn RADIUS/UDP verwendet wird
- Angriff ist trotz CVSS-Score von 9.0 (Temporal score 8.3) i.d.R. unwahrscheinlich

Zum Beginn – Ein Blick auf RADIUS-Pakete

RADIUS REQUEST

RADIUS RESPONSE

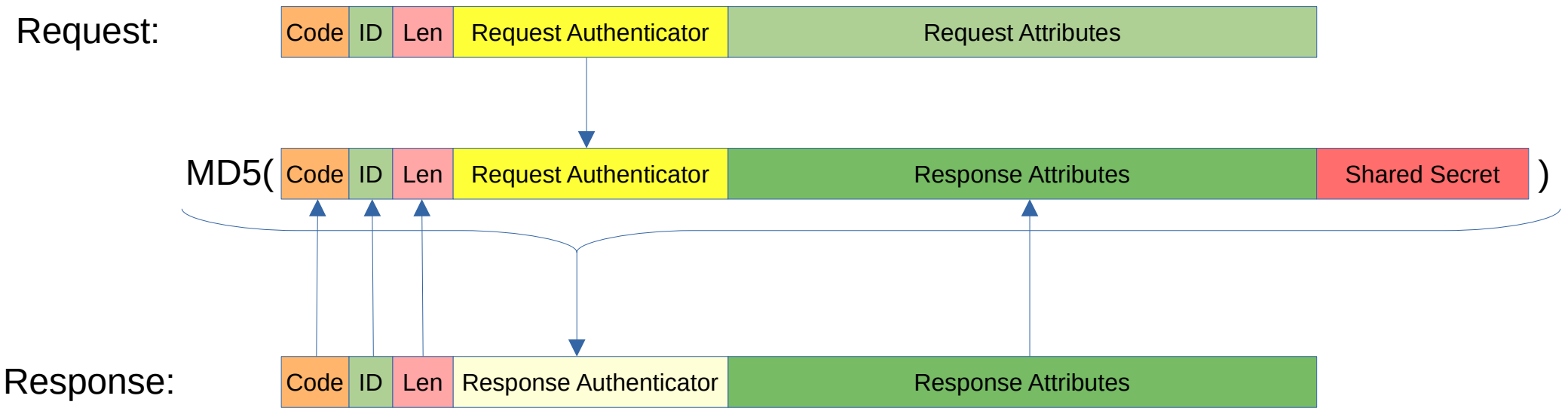


- RADIUS Paket-Format beinhaltet einen „Authenticator“
  - Request Authenticator: Zufallszahl, vom Client generiert
  - Response Authenticator: Integritätsschutz, berechnet auf Basis von MD5, shared secret wird integriert
- RADIUS-Requests haben keinen Integritätsschutz

Zum Beginn – Ein Blick auf RADIUS-Pakete

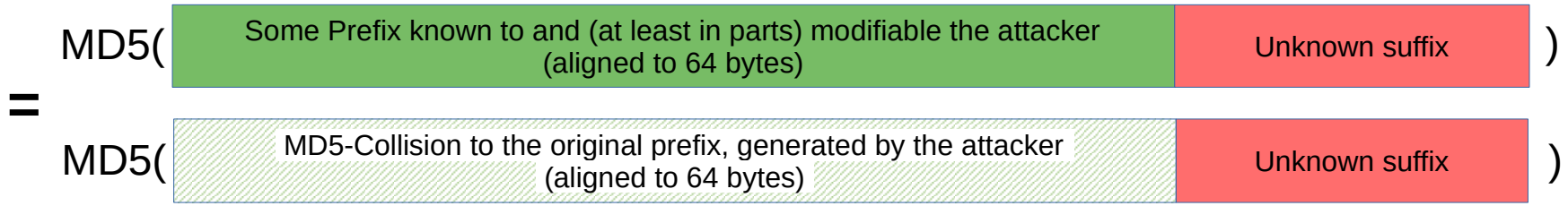
- Berechnung des Reponse-Authenticator ist definiert als: (+ ist Konkatenation)

  MD5(Code+ID+Length+RequestAuth+Attribute+Secret)



MD5-Prefix-Kollision

- Kollisions-Resistenz von MD5 ist gebrochen
- Durch die Block-Architektur von MD5 beinhaltet das Prefix-Kollisionen
- MD5(A + suffix) = MD5(B + suffix) wenn
  - MD5(A) = MD5(B)
  - Länge(A) = Länge(B)
  - Die Länge von A/B ist mit der MD5-Block-Größe ausgerichtet (512 bit = 64 byte)



Der Übeltäter – Das Proxy-State-Attribut

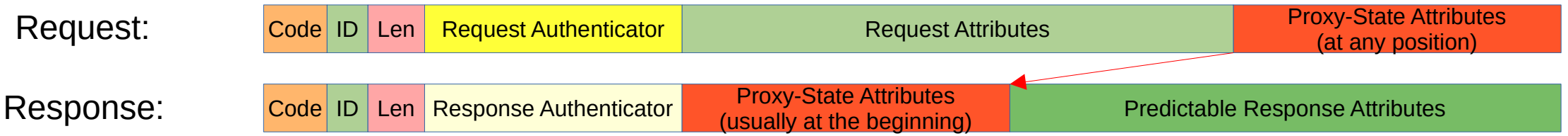
- Spezifikation in RFC 2865, Section 2

```
If any Proxy-State attributes were present in the Access-Request,
they MUST be copied unmodified and in order into the response packet.
Other Attributes can be placed before, after, or even between the
Proxy-State attributes.
```

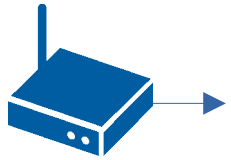
- Jedes Proxy-State Attribute wird in der Antwort zurückgespiegelt

  → da Requests nicht integritätsgeschützt sind, kann Angreifer sie hinzufügen

- Die meisten Implementierungen packen Proxy-State an den Anfang der Antwort (aber Position für Angriff irrelevant, solange sie vorhersagbar ist)



Beispiel-Angriff



1: Client-Request beobachten und Paket unterdrücken

2: Template für von Angreifer:in gewünschte Antwort erstellen (Accept-Gibberish ist veränderlich)

MD5(

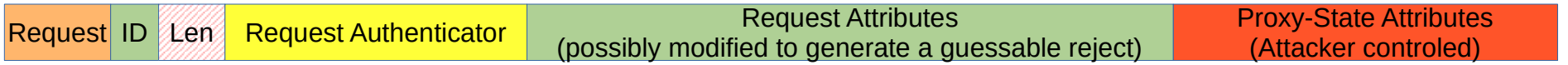
)

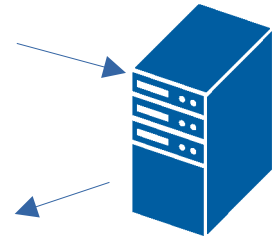
3: MD5-Kollision berechnen mit der Antwort, die vom Server erwartet wird (Proxy-State Attribute veränderlich)

MD5(

)

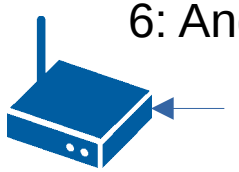
4: Angepassten Request an den Server senden



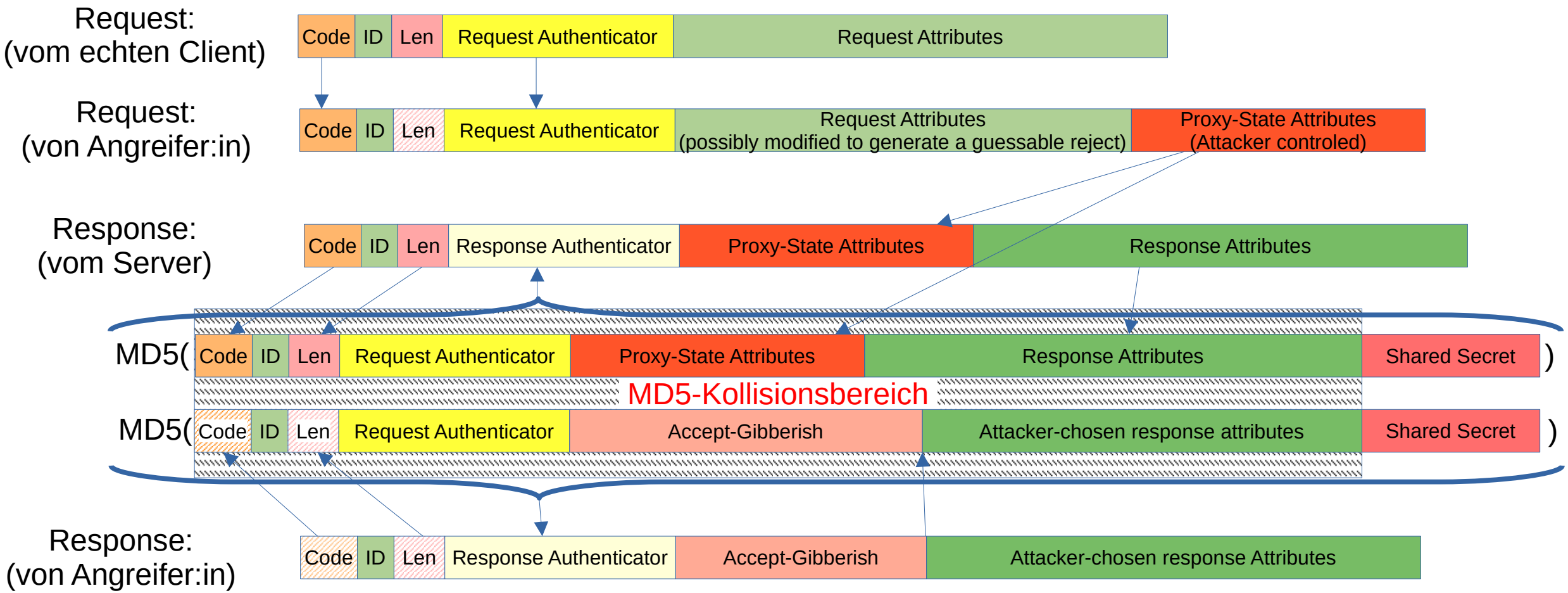


5: Antwort des Servers mit Access-Reject beobachten und unterdrücken

6: Angepasste Antwort an den Client senden, Response Authenticator aus der Server-Antwort kopieren



MD5-Prefix-Kollision in größerem Detail



Angriffswahrscheinlichkeit und -aufwand

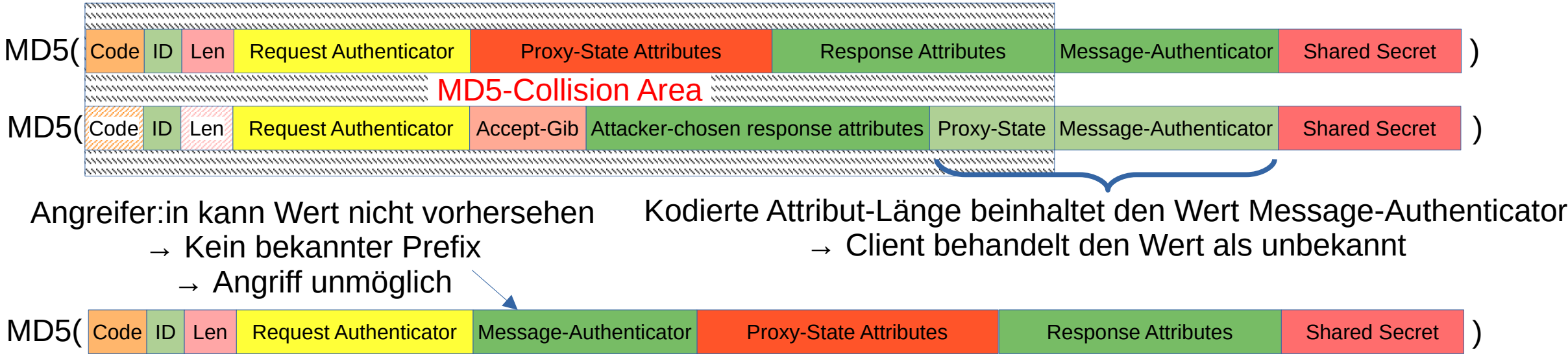
- Angreifer:in benötigt Zeit um die MD5-Kollision zu berechnen
  - Online-Angriff, da Details aus dem Request benötigt werden
  - Proof-of-Concept der Forschenden: Cluster mit ca. 50 Nodes mit Intel-Xeon CPUs (20-40 Kerne) und eine Node mit 4 GPUs: Berechnungszeit von meist 5-10 min → vermutlich länger als die meisten Timeouts
    - Vermutung der Forschenden: 2% erfolgreiche Angriffe in < 4 min
  - Mit entsprechenden Resourcen und spezialisierter Hardware möglicherweise in nahezu Echtzeit möglich
- Angreifer:in braucht Zugriff auf Traffic mit Fähigkeit Pakete zu beobachten und zu unterdrücken

Gegenmaßnahmen – Message Authenticator

- RADIUS-Attribute Message-Authenticator (RFC 2869)
- Integritätsschutz basierend auf HMAC-MD5 (HMAC ist nicht anfällig für Kollisions-Angriffe, also nicht perfekt, aber HMAC-MD5 ist nicht gebrochen)
- Wenn das RADIUS Paket ein **EAP-Message** Attribut enthält, MUSS Message-Authenticator vorhanden sein, Pakete ohne MÜSSEN weggeschmissen werden
  - Die meisten Implementierungen machen das korrekt → EAP ist nicht betroffen
- Für alles andere als EAP: Angreifende könnten das Message-Authenticator Attribut einfach entfernen → kein Integritätsschutz für den Request

Gegenmaßnahmen – Message Authenticator

- Die meisten RADIUS-Server fügen/fügten Message-Authenticator am Ende hinzu
- Angreifer:in könnte Message-Authenticator Attribut maskiren durch Attribute mit einer Länge, die das gesammte Message-Authenticator-Attribut „schluckt“



Reaktive Gegenmaßnahmen

- Wenn RADIUS/UDP weiter genutzt werden soll
  - Message-Authenticator in allen gesendeten RADIUS-Paketen hinzufügen
  - Message-Authenticator in allen eingehenden RADIUS-Paketen erfordern (Gegenstelle muss es dann auch senden)
  - Für Server-Implementierungen: Message-Authenticator an den Anfang setzen

    → Angreifer:in Kann den Prefix nicht raten, verhindert den Angriff selbst dann, wenn Client Message-Authenticator nicht prüft.

- Beispiel FreeRADIUS: Neue Konfigurationsoption **`limit_proxy_state`** (seit v3.0.27/v3.2.5)
  - RADIUS-Requests mit Proxy-State Attributen werden nur akzeptiert, wenn sie Message-Authenticator enthalten
    - Verhindert den Angriff, solange alle Elemente der Proxy-Kette das umsetzen
    - Ermöglicht die sichere Weiterverwendung von legacy-Clients, die keinen Message-Authenticator unterstützen

Die Perspektive

- Migration auf RADIUS/(D)TLS (RFC6614 RADIUS/TLS bzw. RFC7360 RADIUS/DTLS)
  - Integritätsschutz wird von TLS bereitgestellt → Angreifende haben keine Chance
  - Verschlüsselung gibt’s kostenlos oben drauf
  - Demnächst™ auch als „Proposed Standard“ und nicht mehr als „Experimental“
- RADIUS/(D)TLS ist auch mit TLS-PSK nutzbar → Zertifikate nicht notwendig für kleinere Use-Cases
- RADIUS/UDP wird von der IETF in naher Zukunft offiziell als „Deprecated“ deklariert
  - Sämtliche Verwendungen von RADIUS/UDP außerhalb von geschützten internen Netzen sind damit laut IETF veraltet
- Weiterer Ausblick: RADIUS/1.1
  - Auch (D)TLS basiert, nutzt RADIUS/(D)TLS-Port weiter, Aushandlung per TLS ALPN
  - Entfernt die Benutzung von MD5 aus RADIUS (Integritätsschutz wird von TLS bereits bereitgestellt, daher ist Authenticator auf MD5-Basis und Message-Authenticator nicht mehr nötig)

Links und Resourcen

https://www.blastradius.fail/

https://blog.cloudflare.com/radius-udp-vulnerable-md5-attack/

https://kb.cert.org/vuls/id/456537

https://www.cve.org/CVERecord?id=CVE-2024-3596

https://www.dfn.de/blastradius-newsmeldung/

https://datatracker.ietf.org/wg/radext/about/

https://datatracker.ietf.org/doc/draft-ietf-radext-deprecating-radius/

https://datatracker.ietf.org/doc/draft-ietf-radext-radiusdtls-bis/

https://datatracker.ietf.org/doc/draft-ietf-radext-radiusv11/

Discussion/Questions?

- Contact
  - Jan-Frederik Rieckers

Mail: rieckers@dfn.de
Phone: 0049 30 884299-339
Fax: 0049 30 884299-370

Address:
DFN-Verein, Geschäftsstelle
Alexanderplatz1
10178 Berlin