

1 Allgemeine Aussagen zur IT-Sicherheit

1.1 Verteilte Systeme (optional)

"Ein Verteiltes System (VS) ist eine durch ein Kommunikationssystem lose gekoppelte Menge von Knoten, wobei

- die Knoten kooperieren, um Systemfunktionen auszuführen (verteilte systemweite Kontrolle)
- keine zwei Prozesse dieselbe Sicht des Systemzustands besitzen und insbesondere kein zentraler Prozess existiert, der andere Prozesse mit einer konsistenten, identischen Sicht des globalen Systemzustands versorgen kann."

(Quelle: Prof. Dr. rer. nat. Bernd E. Wolfinger - Datenkommunikation und Rechnernetze (DKR) Skript, S. 15)

"Die 3 wichtigsten Aspekte eines verteilten Systems sind:

- lose gekoppelte Knoten (kein gemeinsamer Speicher; Kommunikation nur durch Nachrichten)
- verteilte systemweite Kontrolle
- kein globaler Systemzustand"

(Quelle: Prof. Dr. rer. nat. Bernd E. Wolfinger - Datenkommunikation und Rechnernetze (DKR) Foliensatz 1, Folie 44)

3 Beispiele für verteilte Systeme:

- Terminalnetze (bestehend aus Terminals und Terminalkonzentratoren)
- Mobile Systeme (bestehend z.B. aus Endgerät und Peripherie-Komponenten wie Kamera, Drucker, Wifi-Festplatte,...)
- ein beliebiger Abschnitt des Internets, der aus mehr als einem Rechner besteht

1.2 Sicherheit verteilter Systeme (optional)

Vorteile:

- Wenn ein Verteiltes System mit Verfügbarkeitsverbund vorliegt, kann nach einem erfolgreichen Angriff auf einen Knoten die Funktionalität des Gesamtsystems weiterhin gewährleistet werden.
- Wenn ein Verteiltes System mit Datenverbund vorliegt, kann nach einem erfolgreichen Angriff auf einen Knoten mit einhergehendem Datenverlust die Verfügbarkeit der Daten durch einen anderen Knoten weiterhin gewährleistet werden.
- ...

Nachteile:

- Durch Infektion eines Rechners können die anderen Rechner des verteilten Systems leichter infiziert werden, u.a. da zwischen ihnen regelmäßig Nachrichten ausgetauscht werden

GSS-Übungsblatt 1

Knudsen, Rasch, Runge, Titov · SoSe 2018

- Umfang der Sicherheitsvorkehrungen und Einfachheit des Zugriffs stehen in Konkurrenz zu einander
- ...

1.3 Ursachen (Pflicht, 6 Punkte)

Vermutungen:

- Kosten
- Unwissen
- Fahrlässigkeit

Tatsächlich eher:

- Defizite in der Zusammenarbeit von IT und Sicherheitsverantwortlichen
- Mangelnde oder mangelhafte Sicherheitskonzepte für neue Technologien (z.B. Mobile Endgeräte, Cloud, ...)
- Mangel an Collaboration und Ressourcen
- immer mehr verwundbare Endpunkte,
- unwirksame Strategien in der technologischen Implementierung und organisatorischen Priorisierung von IT-Sicherheit
- sowie die Unfähigkeit, Mitarbeitern Best Practices nahezubringen.

(Quelle: https://www.ponemon.org/local/upload/file/Third_Annual_Study_Patient_Privacy_FINAL.pdf)

Referenzen: Studien, z.B. von <kes> – Die Zeitschrift für Informations-Sicherheit, die Microsoft-Sicherheitsstudien, oder auch die oben verlinkte Studie vom Ponemon Institute.

Komplikationen:

- Unternehmen werden unter Umständen nicht dran interessiert sein, Probleme ihres Umgangs mit IT-Sicherheit zu offenbaren, zumal der Punkt "Unwissen" die kritische Auseinandersetzung damit erschwert bzw. evtl. sogar unmöglich macht.
- Studien werden evtl. durch die Auftraggeber oder die Durchführenden selbst insofern kompromittiert, als dass die dargestellten Ergebnisse nicht (ganz) der Realität entsprechen.
- Befragte haben schlichtweg keine Fachkenntnisse und bieten somit ein verzerrtes Bild der Realität.

GSS-Übungsblatt 1

Knudsen, Rasch, Runge, Titov · SoSe 2018

1.4 Digitale Signaturen (optional)

Im Gegensatz zu einer Signatur im Sinne einer physischen Unterschrift mit Stift auf Papier handelt es sich "bei der digitalen Signatur (DSig) [...] um einen asymmetrischen elektronischen Schlüssel, der die Identität des Benutzers sicherstellt. Der Schlüssel wird mit dem privaten Schlüssel des Absenders verschlüsselt und vom Empfänger mit dem öffentlichen Schlüssel gelesen."

(Quelle: <https://www.itwissen.info/Digitale-Signatur-digital-signature-DSig.html>)

2 Schutzziele

2.1 Abgrenzung I (Pflicht, 14 Punkte)

a) Anonymität, Pseudonymität und Unbeobachtbarkeit

Während die Identität eines Akteur bei gewahrter *Anonymität* lediglich nicht preisgegeben wird, wird bei *Pseudonymität* eine falsche Identität vorgeschoben, wobei die wahre Identität bei Eintritt besonderer Rahmenbedingungen ermittelt werden kann, und bei *Unbeobachtbarkeit* ist weder seine wahre, noch eine vorgetäuschte Identität ersichtlich. Seine Handlungen können nicht beobachtet werden oder es kann nicht festgestellt werden, dass die Handlungen einem einzelnen Akteur zuzuordnen sind.

b) Vertraulichkeit und Verdecktheit

Während bei *Vertraulichkeit* die Daten eines Akteurs vor Blicken Dritter geschützt sind, ist bei *Verdecktheit* das Wissen über das Stattfinden einer Übertragung an sich vor Dritten geschützt.

c) Integrität und Zurechenbarkeit

Bei *Integrität* wird versucht, eine Veränderung der Daten zu verhindern, während bei *Zurechenbarkeit* die Veränderung der Daten einem Akteur nachgewiesen werden kann.

2.2 Techniken (optional)

3 Angreifermodell

3.1 Angreifermodell (optional)

Das Angreifermodell definiert die maximal berücksichtigte Stärke eines Angreifers, gegen den ein Schutzmechanismus gerade noch wirkt.

Es beschreibt:

- Rollen des Angreifers (Außenstehender, Benutzer, Betreiber, Wartungsdienst, Produzent, Entwerfer ...), auch kombiniert
- Verbreitung des Angreifers (Stellen im System, an denen der Angreifer Informationen gewinnen oder Systemzustände verändern kann)
- Verhalten des Angreifers
 - passiv / aktiv, beobachtend / verändernd
- Rechenkapazität des Angreifers
 - unbeschränkt: informationstheoretisch
 - beschränkt: komplexitätstheoretisch

(Quelle: Prof. Dr. Hannes Federrath - Sicherheit in verteilten Systemen (SVS)-Foliensatz 1 - "Einführung in die IT-Sicherheit", Folie 26)

3.2 Praxisbeispiel (Pflicht, 10 Punkte)

Angreifermodell für das Abheben von Bargeld an Geldautomaten mit einer EC-Karte

Angreifermodell 1

Rolle: Außenstehender, weiterer Benutzer (aber auch andere Insider)

Verbreitung: Sichtkontakt zum Automaten (PIN-Eingabe)/ ggf. über Kamera/Spiegel

oder Anbringen einer Vorrichtung zur Ermittlung der PIN Verhalten: passiv, beobachtend / aktiv, beobachtend (Sicherheitsdienst) / aktiv, verändernd (Spiegel, Vorrichtung, unerlaubte Kameranutzung)

Rechenkapazitäten: unbeschränkt.

Angreifermodell 2

Rolle: Außenstehender, weiterer Benutzer (aber auch andere Insider)

Verbreitung: Besitz einer betrügerisch erworbenen gültigen EC-Karte oder Erlangen der EC-Kartendaten durch unerlaubte Handlungen

Verhalten: aktiv, verändernd

Rechenkapazitäten: beschränkt.

4 Angriffsformen

4.1 Essenslieferungen (Pflicht, 7 Punkte)

a) passive Angriffe: Bereits das Schließen von erhöhten Essenslieferungen auf einen bevorstehenden Kampfeinsatz gefährdet die Vertraulichkeit dieser Information. Durchführen lässt sich der Angriff durch das Beobachten der Nahrungslieferanten oder des Verteidigungsministeriums (Traffic Analyse). Eine Gegenmaßnahme wäre eine Kantine im Verteidigungsministerium, die vergleichbares Essen liefert.

b1) aktive Angriffe: Durch das Vergiften der Lebensmittellieferungen kann Einfluss auf die Verfügbarkeit der Soldaten genommen werden. Der Angriff kann auch Einfluss auf die Integrität der Planung haben, da unter Drogen stehende Soldaten die Planung manipulieren könnten. Alle relevanten Restaurant oder Lieferdienstfahrer müssten unter Kontrolle gebracht werden. Eine Gegenmaßnahme wäre eine Kantine im Verteidigungsministerium, die vergleichbares Essen liefert.

b2) Wenn alle relevanten Restaurant oder Lieferdienstfahrer unter Kontrolle gebracht wurden, sind auch weitere Angriffe auf die Verfügbarkeit per Denail of Service oder Flooding möglich.

4.2 Liste bekannter WLAN-AP SSIDs (optional)

a) Der Hauptangriffsvektor ist über eine Man-in-the-Middle-Attacke, es ist sowohl möglich die Verkehrsdaten zu analysieren, wie auch Einfluss auf die Integrität und Verfügbarkeit der Daten zu nehmen. Ein Angriff kann durchgeführt werden, wenn man einen Sender mit der gleichen SSID und größerer Signalstärke installiert. Als Gegenmaßnahme zum Schutze der Vertraulichkeit und Integrität kann ein verschlüsseltes VPN verwendet werden. Die Verfügbarkeit lässt sich so nicht schützen.

b) Eine Denail of Service Attacke ist möglich, in dem der WLAN-AP mit vielen Verbindungsanfragen überlastet wird. Ein Schutz ist nicht möglich, außer es wird auf eine andere Technik wie UMTS gewechselt.

GSS-Übungsblatt 1

Knudsen, Rasch, Runge, Titov · SoSe 2018

5 Passwortsicherheit

5.1. Eigenschaften kryptographischer Hashfunktionen (optional)

Erläutern Sie drei Eigenschaften kryptographischer Hashfunktionen!

Antwort:

- a) Zeichenfolge beliebiger Länge (Eingabewert) wird auf eine Zeichenfolge mit fester Länge (Hashwert) abbildet.
- b) Einwegfunktion
- c) Kollisionsresistenz

5.2. Einfaches Hash-Verfahren (optional)

Durch kryptographische Hash-Funktionen können Kennwörter sicherer als im Klartext hinterlegt werden. Nennen Sie zwei Gründe, warum die Kennwörter in einem IT-System nicht im Klartext abgespeichert werden sollten. Wie funktionieren Kennwortspeicherung und -prüfung unter Verwendung einer Hash-Funktion im einfachsten Fall? Warum ist dieses Verfahren sicherer als die Abspeicherung im Klartext?

Antwort:

- a) Wenn jemand Zugriff auf das System hat, kann er auch das Passwort nutzen.
 - i) Ggf. wird das gleiche Passwort auch in anderen Systemen verwendet.
 - ii) Oder jemand kann sich ggü. dem System als Jemand anderes Identifizieren.
- b) Um nicht auf <http://plaintextoffenders.com/> bloßgestellt zu werden.
 - Aus dem Passwort des Users wird (am besten lokal auf dem Gerät des Users) ein Hash des Passworts erzeugt, dieser wird mit dem Hash in der Passwortdatenbank verglichen.
 - Siehe a.i und a.ii

5.3. Brute-Force-Angriff (Pflicht, 6 Punkte) Bei vielen Unix-Betriebssystemen wurden früher lediglich die ersten acht Stellen eines Kennwortes verwendet. Wie lange benötigt ein Passwort-Cracking-Tool in diesem Fall maximal, das eine Million Passwörter pro Sekunde prüfen kann, wenn bekannt ist, dass das Passwort lediglich aus alphanumerischen Zeichen besteht. Wäre es im Vergleich dazu für das Passwort-Cracking-Tool aufwendiger, wenn das Betriebssystem keine Beschränkung der Kennwortlänge hat und bekannt ist, dass das Passwort nur aus Zahlen und maximal 16 Stellen besteht? **Antwort:**

- a) alphanumerischen Zeichen, 8 Stellen:
52 Buchstaben + 10 Ziffern = 62 Zeichen
62⁸ Möglichkeiten für das Passwort
 $(62^8 / 1000000) / 60 / 60 / 24 = 2527$ Tage

GSS-Übungsblatt 1

Knudsen, Rasch, Runge, Titov · SoSe 2018

b) Zahlen und maximal 16 Stellen:

10 Zeichen

10^{16} Möglichkeiten

$(10^{16}/1000000)/60/60/24 = 11570$ Tage

5.4. Time-Memory-Trade-Off (optional) Eine leistungsfähige Technik, mit der auf Basis eines Hashwerts ein dazu passendes Passwort ermittelt werden kann, stellen sogenannte Rainbow Tables dar. Zur Beantwortung können Sie <http://www.h-online.com/security/features/Cheap-Cracks-Of-dictionaries-and-rainbows-746217.html> heranziehen. Was versteht man in diesem Zusammenhang unter dem Begriff Time-Memory-Trade-Off? Was ist die grundsätzliche Idee von Rainbow Tables bzw. den Vorgänger-Verfahren? Was haben Rainbow Tables mit dem Regenbogen zu tun?

Antwort: Die grundlegende Idee ist bei Passworthashes ohne Salt eine Datenstruktur möglicher Passwörter und Hashes vorher zu erstellen, um beim Entschlüsseln von den Hashes auf die Passwörter schließen zu können. Die schnellste Version wäre eine Tabelle direkt im Arbeitsspeicher, diese Umsetzung benötigt jedoch viel von diesem. Andere Umsetzungen wären langsamer, jedoch weniger Arbeitsspeicher-hungrig (Time-Memory-Trade-Off). Eine solche Datenstruktur sind Rainbow-Tables.

5.5. Salting (optional) Einen wirksamen Schutz gegen Rainbow Tables stellt das Hinzufügen einer zufälligen Zeichenkette (auch „Salt“ genannt) vor dem Anwenden der Hash-Funktion h auf ein Kennwort dar, was als $h(\text{SALT}|\text{PASSWORD})$ ausgedrückt werden kann. Warum?

Antwort: Eine weitere Methode, die Generierung von Rainbow Tables unwirtschaftlich zu machen, ist der Einsatz von Salt. Dabei wird an das Passwort vor dem Hashen ein – im Idealfall – zufällig generierter Wert, das Salt, angehängt. Das Salt wird zusammen mit dem Hashwert gespeichert, um das Passwort später überprüfen zu können, es ist also kein Geheimnis. Der Salt vergrößert entsprechend die Menge der möglichen Passwörter.

5.6. Dictionary-Attack (Pflicht, 10 Punkte) Schreiben Sie ein Programm (z. B. in Java, Ruby, Python oder Perl) zum Ermitteln eines Kennworts anhand eines Hashwerts mit einem Wörterbuch-Angriff. Besorgen Sie sich dazu ein deutsches Wörterbuch aus dem Internet. Testen Sie Ihre Implementierung mit den unten angegebenen Daten. Über das gespeicherte Kennwort sind die folgenden Fakten bekannt: Es ist ein deutsches Wort, steht im Wörterbuch, ist kleingeschrieben und nicht länger als 6 Zeichen. Der Salt wird beim Hashen vor das Passwort gestellt. Bei der Hash-Funktion handelt es sich um MD5.

```
# user:salt:hash
berta;x0hth4dew5p8:14146888a9cb5e924987691876fb4252
```

Welches Kennwort konnten Sie ermitteln? Skizzieren Sie die Funktionsweise Ihres Programms anhand der wichtigsten Stellen Ihres Programms (bitte nicht separat abgeben, sondern ins PDF einfügen). Achten Sie dabei auf Verständlichkeit und Übersichtlichkeit. Wie müsste das Programm erweitert werden, wenn der Salt im Vorfeld nicht bekannt wäre?

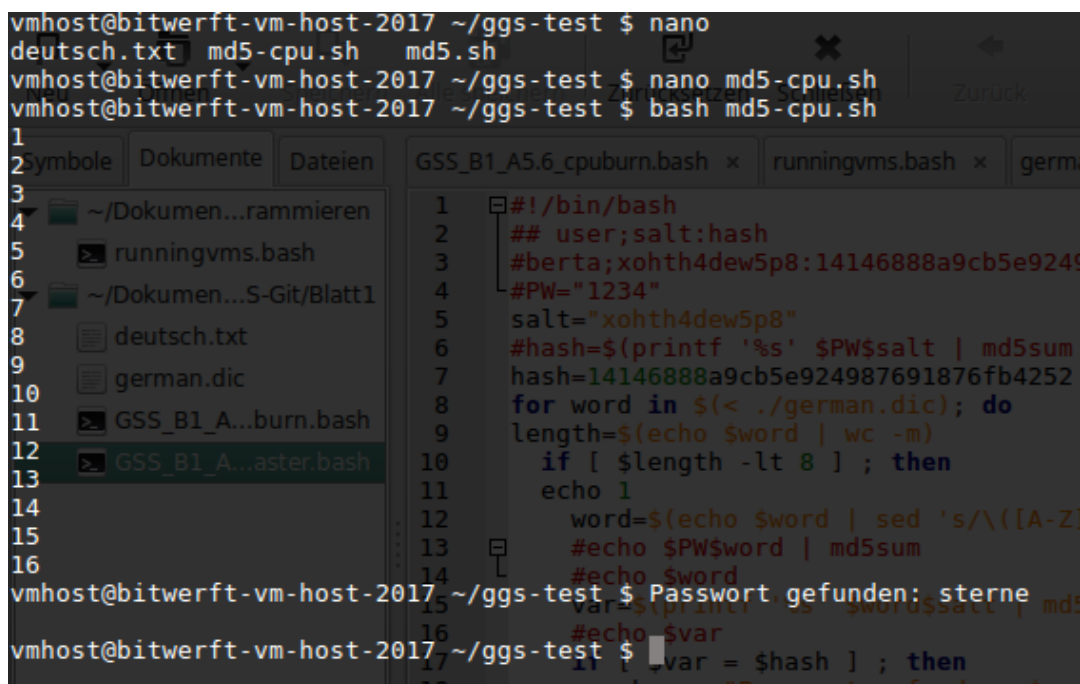
Antwort: Gefundenes Passwort: sterne

GSS-Übungsblatt 1

Knudsen, Rasch, Runge, Titov · SoSe 2018

```
#!/bin/bash
## user:salt:hash
#berta;xohth4dew5p8:14146888a9cb5e924987691876fb4252
#PW="1234"
salt="xohth4dew5p8"
#hash=$(printf '%s' $PW$salt | md5sum | cut -d ' ' -f 1)
hash=14146888a9cb5e924987691876fb4252
#numofcpu=4
numofcpu=$(cat /proc/cpuinfo | grep processor | wc -l)
# Anzahl der CPU-Kerne = Anzahl der Threads.

for ((z=1;z<=$numofcpu;z++)); do
  for word in $(split --number=1/$z/$numofcpu ./deutsch.txt); do
    length=$(echo $word | wc -m) #Berechnen der Wortlaenge
    if [ $length -lt 8 ] ; then #wenn Wort + EOL < 8 ...
      word=$(echo $word | sed 's/\([A-Z]\)/\L\1/g')
      #Wort aus Woerterbuch alle Zeichen klein
      var=$(printf '%s' $salt$word | md5sum | cut -d ' ' -f 1)
      if [ $var = $hash ] ; then
        echo -e "Passwort gefunden: $word"
      fi
    fi
  done & echo -e "Thread $z of $numofcpu started"
done
```



```
vmhost@bitwerft-vm-host-2017 ~/ggs-test $ nano
deutsch.txt md5-cpu.sh md5.sh
vmhost@bitwerft-vm-host-2017 ~/ggs-test $ nano md5-cpu.sh
vmhost@bitwerft-vm-host-2017 ~/ggs-test $ bash md5-cpu.sh

1
2 symbole Dokumente Dateien GSS_B1_A5.6_cpuburn.bash x runningvms.bash x germ
3
4 ~/Dokumen...rammieren
5 runningvms.bash
6 ~/Dokumen...S-Git/Blatt1
7 deutsch.txt
8 german.dic
9 GSS_B1_A...burn.bash
10 GSS_B1_A...aster.bash
11
12
13
14
15
16
vmhost@bitwerft-vm-host-2017 ~/ggs-test $ Passwort gefunden: sterne
vmhost@bitwerft-vm-host-2017 ~/ggs-test $
```

Wenn der Salt nicht bekannt wäre, müsste nach einem Passwort in der maximalen Länge von Passwortlänge + Saltlänge gesucht werden.