Bristande autentisering och sessionshantering

Daniel Bosk[1]

Avdelningen för informations- och kommunikationssytem (IKS),
Mittuniversitetet, Sundsvall.

auth.tex 1282 2013-09-10 19:11:58Z danbos

Mittuniversitetet
MID SWEDEN UNIVERSITY

Översikt

Litteratur

Föreläsningen utgår från avsnittet *A2 Broken Authentication and Session Management* i [Pro13]. Innehållet i föreläsningen tar även upp delar från kapitlen ”Authentication” och ”Session Management” i [owa05], kapitel 3 ”Protocols” i [And08] samt avsnitten ”V1: Authentication Verification Requirements” och ”V2: Session Management Verification Requirements” i [owa13]. För att behandla lösenordspolicyer bör även artikeln ”Of passwords and people” [KSK+11] läsas.

Översikt

Vad säger ”A2 Broken authentication and session management”?

Autentisering Handlar om att vi vill identifiera olika principals, eller låta olika principals identifiera sig till oss. Det vill säga, knyta en identitet till en principal.

Auktorisation Handlar om åtkomstkontroll. Vi vill tillåta vissa principals men hindra andra. Detta kräver att vi autentiserat dem på ett korrekt sätt.

Sessionshantering Krävs för att autentiserad principal inte ska behöva autentisera sig för varje anrop.

Vad säger ”A2 Broken authentication and session management”?

Hotagenter Externa anonyma utan och interna användare med konton är båda möjliga hot.

Attackvektorer Läckor och buggar i autentiserings- eller sessionshanteringssystemet.

Svagheter Egenutvecklade mekanismer för autentisering och sessionshantering, att konstruera sådana är svårt. Använd färdiga välutvecklade bibliotek istället.

Konsekvenser Angriparen kan utge sig för att vara den legitima användare vars konto angripits.

Exempel

Exempel

Scenario 1

```
http://foo.bar/sale/saleitems?jsessionid=
    2P0OC2JSNDLPSKHCJUN2JV?dest=Bahamas
```

Användaren kopierar sin URL och skickar till en kollega för att visa resorna. Kollegan kan beställa resor med detta konto om kortuppgifter etc. finns lagrade.

Exempel

Exempel

Scenario 2 Användaren loggar in på en publik dator. Istället för att logga ut stänger denne webbläsaren när denne lämnar datorn. Eftervarande är fortfarande inloggad om timeouts inte fungerar som de borde.

Exempel

Exempel

Scenario 3 Angripare kommer åt lösenordsdatabasen. Denne kan senare autentisera sig som vilken användare som helst i systemet. (Mer om detta i en senare föreläsning.)

Översikt

Vad är ett protokoll?

- Ett system består av en uppsättning principals.
- Ett protokoll är en uppsättning regler som styr hur dessa kommunicerar.

Exempel (Tentamen MIUN)

1. Tentamensvakten öppnar salen och ger varje tentand ett nummer.
2. Tentanden går in och sätter sig vid sin tilldelade plats.
3. Efter att tentan börjat jämför tentamensvakten tentandens legitimation och nummer.
4. Vid inlämning av skrivning jämförs legitimationen och numret.

Vad är ett protokoll?

- Bör vara designade för att motstå attacker.
- Både oavsiktligt och avsiktligt brott mot protokollet.

Vad är ett protokoll?

Exempel

Beställa vin

1. Hovmästaren visar vinlistan för värden.
2. Värden väljer vin, hovmästaren hämtar.
3. Värden provsmakar vin, det serveras till gästerna.

Egenskaper

Konfidentialitet Gästerna får ej veta priset.

Riktighet Hovmästaren kan inte byta ut vinet.

Oavvislighet Värden kan inte falskt klaga på vinet i efterhand.

Vad är ett protokoll?

Exempel (Tentamen MIUN)

1. Tentamensvakten öppnar salen och ger varje tentand ett nummer.
2. Tentanden går in och sätter sig vid sin tilldelade plats.
3. Efter att tentan börjat jämför tentamensvakten tentandens legitimation och nummer.
4. Vid inlämning av skrivning jämförs legitimationen och numret.

Egenskaper

Anonymitet Varje tentand förblir anonym.

Autenticitet Skrivningen är garanterat associerad med tentanden.

Mittuniversitetet
MID SWEDEN UNIVERSITY

Vad är ett protokoll?

- Konstrueras utifrån grundläggande antaganden.
  - Exempelvis att kortägaren kan mata in PIN-koden direkt i terminalen.
- Analysera om hoten är rimliga.
- Analysera om protokollet hanterar dem.

Ibland blir det fel

Tentamen

Exempel (Tentamen KTH)

KTH:s tentamensprotokoll har inte med den sista punkten[a].

1. Tentamensvakten öppnar salen.
2. Tentanden går in och sätter sig.
3. Efter att tentan börjat undersöker tentamensvakten tentandens legitimation och antecknar plats i salen.
4. Tentanden lämnar in skrivningen.

[a] KTH har sedan dess uppdaterat sitt förfarande.

Egenskaper

Autentiseringen brister, tentanden kan skriva vilket namn och personnumer som helst på inlämnad tentamen. Tentanden är inte garanterad anonymitet om den inte explicit ber om att tentan ska skrivas anonymt.

Ibland blir det fel

Uttagsautomat

Autentisera uttag

- Banker lagrade kontonummer på magnetremsan.
- PIN-koden skickades till centrala systemet för verifiering.

”Förbättring”

Kryptera PIN-koden och lagra den på magnetremsan så att uttagsautomaten kan verifiera den när den inte får kontakt med centrala systemet.

Ibland blir det fel

Uttagsautomat

Problem

- Jag kan byta ut kontonumret men inte ändra koden.
- Ändra kontonummer och använd min egen kod för att ta ut från annans konto.

Autentisering

- Lösenord och PIN-koder är fundamentala metoder för autentisiering.
- Exempelvis 90-talets fjärrlås till bilen:
  - Nyckeln skickade serienumret.
  - Bilen kontrollerade allt den mottog och jämförde med sitt serienummer.
- Kunde angripas med en inspelningsattack.
  - Spela in allt som sänds (serienummer).
  - Spela upp serienummer för att låsa upp.

Autentisering

- Ibland kan ett enkelt lösenord vara rätt väg att gå.
- Matkuponger på lunchställen:
  - Papperslapp med serienummer.
  - Kan framställas i kopiatorn.
- Hotet är inte värt högre säkerhet.

Formell notation

Exempel (Protokollbeskrivning)

Två principals P, P' ska kommunicera.

1. P skickar sitt namn till P'.
2. P' svarar med ett token t_P för vidare användning, detta är krypterat med P:s kryptonyckel k_P.

Exempel (Formell beskrivning)

Principals P, P', token t_P, P:s kryptonyckel k_P.

$$P \rightarrow P' : P$$

$$P' \rightarrow P : \{t_P\}_{k_P}$$

Formell notation
Tentamen

Exempel (Autentisering MIUN)

Låt T vara tentanden, V tentamensvakten, n_T det unika numret för T och S skrivningen. Vidare låt k vara en kryptonyckel delad mellan legitimationsutfärdaren och tentamensvakten (legitimation).

$$V \rightarrow T : n_T$$
$$T \rightarrow V : \{T\}_k, n_T$$
$$T \rightarrow V : \{T\}_k, n_T, S$$

Formell notation
Tentamen

Exempel (Autentisering KTH)

Låt T vara tentanden, V tentamensvakten och S skrivningen. Vidare låt k vara en kryptonyckel delad mellan legitimationsutfärdaren och tentamensvakten (legitimation).

$$T \rightarrow V: T, \{T\}_k$$
$$T \rightarrow V: T, S$$

Formell notation

Exempel (Needham-Schröder Shared-Key)

Låt A, B, S vara principals, S en server, n_A, n_B vara nonces[a].

$$A \to S: A, B, n_A$$
$$S \to A: \{n_A, B, k_{AB}, \{k_{AB}, A\}_{k_{BS}}\}_{k_{AB}}$$
$$A \to B: \{k_{AB}, A\}_{k_{BS}}$$
$$B \to A: \{n_B\}_{k_{AB}}$$
$$A \to B: \{n_B - 1\}_{k_{AB}}$$

[a] Number used once.

Formell notation

- För att verifiera protokoll från säkerhetsbrister används formella metoder.
- En av dessa formella metoder är BAN-logik.
- Detta är ett formellt sätt att resonera kring antaganden och vad man kan tro på i protokollen.
- Med BAN-logiken kan man hitta en brist i Needham–Schröder-protokollet.
  - $A \rightarrow B : \{k_{AB}, A\}_{k_{BS}}$
  - B måste anta att detta är färskt, även om så inte är fallet.

Översikt

Enkel autentisering

En bättre metod för fjärrlås

Exempel (Fjärrlås)

Låt A, B vara principals, n nonce, k_A en nyckel unik för A.

$$A \to B : A, \{A, n\}_{k_A}$$

Egenskaper

- Nonce n för färskhet.
- Krypteringen för identifiering.

Enkel autentisering
Nyckelhantering

- Måste hantera nycklarna k_i för alla enheter i.
- *Nyckeldiversifiering*: huvudnyckel k_M och generera $k_i = \{i\}_{k_M}$.
- Måste tänka efter:
  - 128-bitar nyckel krypterar 16-bitar ID, mindre lämpligt för diversifiering.
  - Svagt chiffer ger också dåligt resultat.
  - $k_i = i \oplus k_M$?

Enkel autentisering
Kolla nonces

Kolla nonces långt tillbaka i tiden.

- Jämför med senaste nonce.
- Spela in två och spela upp dem varannan gång.
- Förbetalda elmätare, köp två laddningar och använd dem om vartannat.

Enkel autentisering
Betjäntattacken

- Hur genereras nonces?
- En person som har tillfällig åtkomst att generera tokens.
- Generera ett antal, använd dem senare.
- Exempelvis engångskoder för att logga in hos internetbanken.
- Attacken fungerar om nonces är (pseudo)slumptal.

Enkel autentisering
Kontra betjäntattacken

Förbättring

- Använd en räknare c som successivt ökas på.
- $A \rightarrow B$: $A, \{A, c+1\}_{k_A}$, $c = c + 1$.
- Inget $c' \leq c$ accepteras.

Problem

- Får inte ha jämförelsen $c' = c$, ger synkroniseringsproblem.
- $c \notin \mathbb{Z}_+$ utan $c \in \mathbb{Z}_{2^x}$, för något $x \in \mathbb{N}$: vid något tillfälle blir då $c + 1 < c \pmod{2^x}$.

Challenge–response

Grundläggande princip

Två principals A, B med gemensam nyckel k och nonce n.

$$A \to B\colon n$$
$$B \to A\colon \{B, n\}_k$$

Problem

- Dåliga (pseudo)slumptalsgeneratorer, ger förutsägbara n.

Challenge–response

Tvåfaktorautentisering

- Ha användarnamn och lösenord.
- Komplettera med extern kod; exempelvis genererad av koddosa, SMS till mobiltelefonen.
- Finns många varianter, kombinera två:
  - Något du vet (lösenord),
  - något du har (koddosa, mobiltelefon),
  - något du är (biometrik).

Challenge–response

Tvåfaktorautentisering

Protokoll (tvåfaktorautentisering med koddosa)

Låt A, B, D vara principals, D är koddosa, k är nyckel delad mellan B, D och p är A:s PIN-kod.

$$A \to B: A$$
$$B \to A: n$$
$$A \to D: n, p$$
$$D \to A: \{n\}_k$$
$$A \to B: \{n\}_k$$

Challenge–response
Tvåkanalsautentisering

Protokoll (tvåkanalsautentisering med mobiltelefon)

Låt A, B, M vara principals, M är mobiltelefon och p är A:s lösenord.

$$A \to B: A, p$$
$$B \to M: n$$
$$M \to A: n$$
$$A \to B: n$$

Miljöbyte

- Betalkortsystemet designades för en pålitlig miljö.
- Kraftigt reglerad miljö inbyggd i bankens fasad.
- Tillämpas i den mindre pålitliga miljön i samtliga affärer.
- Skimming.

Miljöbyte
Personen i mitten

- ”Det är enkelt att spela oavgjort mot en schackstormästare i postschack: spela bara mot två stormästare samtidigt, en som vit och en som svart, och skicka deras brev mellan varandra.” (John Convey, från [And08], egen översättning)
- Problem med pålitliga användargränssnitt: hur vet du att inte kortterminalen ljuger?

Internetbanken och betalkort

Olika former av bankdosor

Swedbank

- Individuell dosa, förkonfigurerad av banken.
- Kan generera engångskod.
- Kan hantera challenge–response.

Nordea

- Oberoende smartkortläsare, använder individuellt betalkort.
- Kan generera engångskod.
- Kan hantera challenge–response.

Internetbanken och betalkort

Problem som kan uppstå

Problem

- Om bankkort och dosa förvaras tillsammans kan PIN-koden utläsas från de slitna knapparna på bankdosan.
- Om kortet används i en dålig terminal har angriparna allt som behövs för att logga in till ditt bankkonto.

Förbättringar

- Använd inte samma säkerhetsmekanism i flera sammanhang.
- Ha separata oberoende mekanismer.
- Ha ett pålitligt användargränssnitt.

Översikt

Sessioner

- HTTP är ”stateless”.
- Varje anrop är nytt och kan ses som att det kommer från en ny användare.
- För att användaren ska förbli ”inloggad” krävs ”state”.
- För detta har vi sessioner.

Sessioner

För sessionshantering, använd alltid färdiga implementationer!

Autentisering

- Autentisering är bara så stark som användarhanteringen.
- Ingen legitimation, inte verifierad person.
- Om personer tillåts skapa konton själva, då behöver vi inte göra mer än att skydda kontona från alla som inte skapat kontot.

Autentisering

- Använd en autentiseringsmetod som är lämplig för den skyddade informationen.
- Jämför e-post, bank och Skatteverket.
- Man behöver inte logga in enbart en gång, verifiera användaren igen vid kritiska operationer; exempelvis vid lösenordsbyte, överföringar av resurser.
  - Autentisera då transaktionen, inte användaren.
  - Personen-i-mitten.

Autentisering

- Använd färdiga implementationer där möjligt; exempelvis OAuth, Google- eller Facebook-koppling.

Referenser

[And08] Ross J. Anderson. *Security engineering : a guide to building dependable distributed systems*. Wiley, Indianapolis, IN, 2 utgåvan, 2008.

[KSK+11] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Christin Nicolas, Lorrie Faith Cranor och Serge Egelman. Of passwords and people: Measuring the effect of password-composition policies. I: *CHI*, 2011.

[owa05] A guide to building secure web applications and web services, 2005.

[owa13] Owasp application security verification standard 2013, 2013.

[Pro13] The Open Web Application Security Project. OWASP Top 10 - 2013: The Ten Most Critical Web Application Security Risks, Jun 2013.