Grundläggande lösenordsanalys

Daniel Bosk*

pwdanalysis.tex 1674 2014-03-19 14:39:35Z danbos

Innehåll

1 Val av lösenord och en enkel metrik för lösenordsstyrka 1

2 Att angripa lösenord 2
2.1 Forcering applicerad på Exempel 2 3
2.2 Effekten av en lösenordspolicy 3
2.3 Forskning på området 4

1 Val av lösenord och en enkel metrik för lösenordsstyrka

Vi ska nu titta på hur detta kan användas för att undersöka säkerheten hos lösenord. Vi inleder med att definiera styrkan hos ett lösenord.

Definition 1. *Styrkan* hos ett lösenord av längd n som väljs från ett alfabet A är $|A|^n$, det vill säga antalet tecken i vårt alfabet upphöjt till längden av lösenordet.

Anledningen till att vi väljer Definition 1 är för att detta är antalet möjliga lösenord tillika antalet gissningar som krävs, i värsta fall, för att gissa rätt lösenord. Det vill säga, desto fler antal gissningar ett lösenord kräver ju längre tid tar det att gissa och alltså är det säkrare. Rent statistiskt sett kommer antalet gissningar som krävs att vara hälften av vårt mått för lösenordsstyrkan eftersom att i medel kommer vi att behöva gå igenom hälften av antalet lösenord.

Det finns flera angreppssätt för att skapa lösenord, exempelvis genom att välja en kombination av tecken; då har vi bokstäver, siffror och specialtecken som alfabet. Det går också att skapa lösenord genom att slumpmässigt välja några ord som kombineras till ett lösenord; i detta fall utgör ordlistan som vi väljer från vårt alfabet, exempelvis Svenska Akademiens ordlista.

Vi börjar med det första fallet, där vi skapar ett lösenord genom att kombinera tecken.

Exempel 1. Om vi ska skapa ett lösenord som är fem tecken långt och får innehålla bokstäverna A-Z, både gemener och versaler, siffrorna 0-9 samt specialtecknen "!@.%&", då kan vi se att vårt alfabet

$$A = \{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}, \mathrm{a}, \mathrm{b}, \ldots, \mathrm{z}, 0, 1, \ldots, 9, !, @, ., \%, \&\}.$$

Vårt alfabet innehåller således $|A| = 26 \cdot 2 + 10 + 5 = 67$ tecken. Styrkan hos ett sådant lösenord är följaktligen $67^5 = 1350125107$.

Nu fortsätter vi med att titta på fallet med att välja lösenord genom att slumpmässigt välja några ord.

Exempel 2. Vi bestämmer oss för att använda ett lösenord med fyra slumpmässigt valda ord från svenska språket. Detta ger oss ett alfabet

$$A = \{\mathrm{apa}, \mathrm{banan}, \mathrm{hej}, \mathrm{dator}, \ldots\},$$

där A alltså innehåller samtliga ord i Svenska Akademiens ordlista. Enligt Svenska Akademien innehåller ordlistan ungefär 125000 ord [**SAOL**], detta ger oss $|A| \approx 125000$. Enligt Definition 1 får vi då att styrkan för ett lösenord av denna typ är

$$|A|^4 \approx 125000^4 = (2^3 \cdot 5^6)^4 = 2^{12} \cdot 5^{24} = 2^{12} \cdot 25^{12} = (2 \cdot 25)^{12} = 50^{12}, \quad (1)$$

det vill säga 244140625000000000000.

Det senaste fallet kan också ses ur det första fallets perspektiv. Om vi tittar på det sista ledet i likheten i ekvation (1) ser vi direkt att detta skulle exakt motsvara ett alfabet med 50 tecken och en lösenordslängd av 12 tecken.

2 Att angripa lösenord

Den egentliga skillnaden mellan Exempel 1 och Exempel 2 är de möjliga angreppssätten. Exempelvis går det att angripa Exempel 2 på ett sätt som gör att dess styrka blir större.

För att kunna knäcka ett lösenord måste en gissning kunna testas. Detta kan göras på flera sätt. Exempelvis kan vi skicka en gissning till inloggningsprogrammet, exempelvis inloggningsgränssnittet för en webbmail. Detta tar oftast lång tid och gör systemet i fråga varse om ett angreppsförsök. Alternativet är att vi har tillgång till lösenordsdatabasen och kan direkt testa mot den. Detta är inte ovanligt [**Cubrilovic2009rhf**, **Oberheide2010bao**, **Hunt2011abs**, **Cluley2012twp**]. I många system lagras inte själva lösenorden utan ett hashvärde av lösenordet. Det vi gör för att testa vår gissning är att vi beräknar hashvärdet för gissningen och jämför detta med det värde som finns i lösenordsdatabasen. Notera att eftersom många användare återanvänder sina lösenord i flera system behöver vi inte nödvändigtvis ha lösenordsdatabasen för det system vi är intresserade av att angripa, det räcker med att det finns användare som har konton i båda systemen.

Det finns huvudsakligen tre olika former av lösenordsknäckning. Dessa är

- forcering (brute-force attack),
- ordlistemetoden (dictionary-attack), och

- i övrigt kvalificerade gissningar.

Social engineering är egentligen inte en lösenordsknäckningsstrategi utan en generell teknik för att ta sig förbi åtkomstkontroll [**Anderson2008sea**], men eftersom att åtkomstkontroll vanligtvis implementeras med lösenordsskydd är den värd att nämna i sammanhanget. Varför knäcka lösenordet när du kan be användaren att utföra attacken åt dig?

Forcering innebär att vi låter ett program gå igenom alla möjliga teckenkombinationer i vårt alfabet för att finna lösenordet i fråga. Med denna metod är vi garanterade att finna lösenordet, men det kan dock ta väldigt lång tid.

Ordlistemetoden är en effektivare metod, med denna metod använder vi en ordlista med de vanligast förekommande lösenorden och vi låter dessa vara våra gissningar. Det är vanligt att lösenordslistor skapas av publicerade hackade databaser, vilket är fallet i analyserna gjorda av **Cubrilovic2009rhf**, **Oberheide2010bao**, **Hunt2011abs**, **Cluley2012twp** Detta ger färre antal gissningar och går således snabbare, men om lösenordet vi försöker att knäcka inte finns med i ordlistan kommer vi aldrig att kunna knäcka det med denna metod.

2.1 Forcering applicerad på Exempel 2

Låt oss anta att medellängden av orden i Svenska Akademiens ordlista är fem tecken och att dessa tecken enbart är gemener från det svenska språket. Det innebär att Exempel 2 ger oss en lösenordsstyrka på

$$29^{5\cdot 4} = 29^{20} = 176994576151109753197786640401. \tag{2}$$

Med hjälp av logaritmen kan vi enkelt se vilken av dessa som är störst. Vi har att $\log(29^{20}) \approx 29$ medan $\log(50^{12}) \approx 20$. Att konstruera lösenordet utifrån fyra slumpmässigt valda ord är alltså mycket starkare sett ur detta perspektiv.

Hur spelar denna representation någon roll, vad betyder skillnaden mellan ekvation (1) och ekvation (2)? Det första som bör påpekas är att i ekvation (2) tas även teckenkombinationer som ej är svenska ord med. Detta eftersom att valet var att välja fyra uppsättningar av fem tecken. Betydelsen av detta är att om vi låter en dator bara slumpa fram 20 bokstäver (gemener), då kommer det att resultera i antalet gissningar från ekvation (2). Men det är inte ens säkert att datorn kommer att hitta rätt lösenord om vi råkade välja fyra långa ord som alla var minst sex bokstäver långa. Detta är ett problem med denna uppskattning.

Om vi däremot använder Svenska Akademiens ordlista som alfabet när vi tillämpar forcering då kommer antalet gissningar att maximalt bli de från ekvation (1) och vi kommer garanterat att finna lösenordet.

Utifrån detta kan vi konstatera att denna enkla modell är för enkel för att säkert kunna resonera om styrkan hos olika typer av lösenord. Vår modell här kan användas för att på ett enkelt sätt översiktligt jämföra styrkan hos olika typer av lösenord. Det har däremot forskats fram mer formella modeller, vilket vi ser i nästa avsnitt, som kan användas för att resonera kring starka och svaga lösenord.

2.2 Effekten av en lösenordspolicy

Den något enkla lösenordspolicyn som kräver minst åtta tecken med gemener, versaler och siffror – utan krav på något antal inom de olika kategorierna – ger

$(26 + 26 + 10)^8 = 62^8 \approx 2^{48}$ antal lösenord. Denna policy har inga krav på giltighetstid hos ett lösenord.

Universitetets lösenordspolicy kräver minst åtta tecken. Dessa tecken ska vara minst tre gemener, tre versaler och två siffror – dessutom måste dessa finnas bland de första åtta tecknen i lösenordet. Detta ger $26^3 26^3 10^2 = 26^6 10^2 \approx 2^{35}$ antal möjliga lösenord. Lösenordet måste dessutom bytas var tredje månad, vilket i sin tur ger risken för lösenordssystem där användaren baserar det nya lösenordet på det gamla.

Resultatet av detta är en reduktion av komplexiteten från 62^8 ned till $26^6 10^2$. Detta utgör en relativ minskning av komplexiteten med $1 - \frac{26^6 10^2}{62^8} = 0.99986$, alltså 99.99 procent. Om vi bortser från journalistformuleringen i föregående mening och tar det akademiska perspektivet ser vi att den första policyn är cirka $2^{13} = 8192$ gånger mer komplex.

Oavsett vilken av ovan givna policyer som används får användarna (sannolikt) svaga lösenord.

Övning 1. Förklara hur dessa lösenordspolicyer kan angripas.

Övning 2. Ge ett förslag på en riktigt bra lösenordspolicy. Glöm inte att en lösenordspolicy är meningslös utan tillhörande analys av den.

2.3 Forskning på området

Angreppsmetoder mot och hur användare väljer lösenord och -fraser är ett aktivt forskningsområde. Metoderna blir alltmer avancerade och exempelvis undersöker **Bonneau2012lpo** hur lingvistiken påverkar valet av lösenord bestående av flera ord. **Bonneau2012lpo** finner att användarna inte väljer slumpmässiga ord utan föredrar att välja dem anpassade efter naturligt språk. Exempelvis XKCD:s "correct horse battery staple"[1] föredras framför "horse correct battery staple" på grund av att det första alternativet är mer grammatiskt korrekt.

Kuo2006hso gjorde en undersökning av hur användare skapar lösenord som är lätta att komma ihåg. **Kuo2006hso** undersökte styrkan hos frasbaserade lösenord. Det vill säga lösenord som skapas utifrån en mening, exempelvis Googles exempel "To be or not to be, that is the question"[2] som ger lösenordet "2bon2btitq". Det visade sig då i undersökningen att denna typ av lösenord är lite säkrare än medellösenordet, men användare väljer fortfarande lösenord som är lätta att gissa.

Komanduri2011opa genomförde också en undersökning om lösenordsstyrka och hur lösenordspolicyer påverkar valet av lösenord. Dessa använder Shannons entropi [**Shannon1948amt**] som metrik för lösenordsstyrka. Denna metrik är väl anpassad för denna typ av undersökning, den går dock inte att tillämpa utan tillgång till en större samling av valda lösenord. De fann att den lösenordspolicy som gav starkast valda lösenord var den enkla policyn att lösenordet ska vara minst 16 tecken långt, inga andra krav. Denna policy visade sig även vara den som gav lösenord som var enklast att komma ihåg.

Utöver de ytliga analyserna av läckta lösenordsdatabaser som gjordes av **Cubrilovic2009rhf**, **Oberheide2010bao**, **Hunt2011abs**, **Cluley2012twp**

[1] URL: `http://xkcd.com/936/`.

[2] URL: `http://www.lightbluetouchpaper.org/2011/11/08/want-to-create-a-really-strong-password-dont-ask-google/`.

har **Bonneau2012sog** gjort en mer formell och djupgående analys. **Bonneau2012sog** har tittat på lösenorden hos nästan 70 miljoner Yahoo!-användare och har då kunnat undersöka skillnader mellan olika demografiska grupper. I artikeln utvecklas en metrik för styrkan hos lösenord, en mer formell och ingående än den mycket simpla metrik some ges i Definition 1.

Bonneau2012ghs har skrivit sin avhandling om tillvägagångssätt för att gissa hemligheter valda av människor, där lösenord är en självklar sådan hemlighet. I avhandlingen presenteras en matematisk modell för mänskligt val och en metrik för att modellera motståndskraften mot olika gissningsattacker.