Mittuniversitetet
MID SWEDEN UNIVERSITY

Laboration: Lösenordsknäckning och *advanced persistent threats*

Daniel Bosk och Lennart Franked*

passwd.tex 1635 2014-02-25 08:19:09Z danbos

Innehåll

1 Introduktion **1**

2 Syfte **2**

3 Läsanvisningar **2**

4 Genomförande **2**
4.1 Inledande reflektion kring lösenord 2
4.1.1 Knäckning av lösenord 2
4.1.2 Ophcrack 3
4.1.3 John the Ripper 4
4.2 Social engineering och APT 5

5 Examination **5**

1 Introduktion

Denna laboration handlar om styrkan hos olika lösenord, hur dessa bör utformas för att vara säkra och praktiskt testa att knäcka olika lösenord.

I vissa fall spelar styrkan hos lösenordet ingen roll, varför knäcka lösenordet när det är enklare att manipulera användaren att ge det till dig? Detta faller under kategorin social engineering. Laborationen tar även upp grundläggande aspekter av detta och *advanced persistent threats* (APT:s).

*Inspirerat av en tidigare laboration av Curt-Olof Klasson och Rahim Rahmani.

2 Syfte

Syftet med denna laboration är:

- Att du ska ha erfarenhet av hur utformandet av lösenord påverkar enkelheten att knäcka dem.
- Att du ska kunna redogöra för olika tillvägagångssätt för att komma åt lösenordsskyddade resurser utan att ha tillgång till själva lösenordet.
- Att du ska få en uppfattning om *advanced persistent threats* (APT) och vad detta kan innebära.

3 Läsanvisningar

Innan du genomför denna laboration ska du ha läst kapitel 2 "Usability and Psychology" i Anderson [1]. Du ska också ha läst kompendiet av Bosk [2] och artikeln av Kuo m.fl. [8].

Sedan ska även några artiklar om lösenordsincidenter läsas [6; 3; 9; 4].

Du ska även läsa artikeln av Fisher [5] som behandlar hur attacken mot RSA genomfördes i början av 2011 samt artikeln av Juels och Yen [7] som diskuterar *advanced persistent threats*.

4 Genomförande

Genomförandet kommer att bestå av två moment. I det första ska du fundera över säkra lösenord och sedan försöka att knäcka två lösenord för Windows och ett för ett UNIX-likt system. I det andra momentet ska du spekulera kring möjligheterna för en social engineering-baserad attack.

4.1 Inledande reflektion kring lösenord

Du ska nu reflektera över styrkan hos olika utformade lösenord. Till din hjälp har du Anderson [1] och Bosk [2] enligt ovan.

Börja med att jämföra hur många tecken långt ett lösenord med gemener, versaler och specialtecken måste vara föra vara lika starkt som ett lösenord bestående av fyra slumpmässigt valda ord.

Om orden inte är slumpmässigt valda utan kanske snarare är ett välkänt citat, exempelvis ett filmcitat eller citat från en TV-serie, hur tror du att detta påverkar styrkan hos lösenordet? Glöm inte en övertygande motivering.

Var tror du att gränsen går för ett säkert lösenord idag? Gör en uppskattning för hur starkt ett lösenord behöver vara för att vara säkert. Glöm inte en övertygande motivering.

4.1.1 Knäckning av lösenord

På sidan

`http://sectools.org/tag/crackers/`

hittar du en lista med program för att knäcka lösenord. De två program som rekommenderas är *John the Ripper* och *Ophcrack*. Det är naturligtvis tillåtet att använda valfritt program, men dessa två är testade och ska fungera. Dessa program finns installerade i datorsalen L209, campus Sundsvall.

För ett UNIX-likt operativsystem finns lösenordens hash- med tillhörande salt lagrade i filen

`/etc/shadow`,

under Linux-baserade system som Ubuntu, alternativt i filen

`/etc/master.passwd`,

under BSD-baserade operativsystem som OpenBSD och FreeBSD. Superuserrättigheter (root) krävs för att kunna läsa denna fil. Lösenordens hashvärden för Windows kan utvinnas genom programmet `fgdump`. Programmet finns tillgängligt på

`http://www.foofus.net/~fizzgig/fgdump/`.

De hashvärden som ni ska hitta lösenord för kommer från både Windows- och UNIX-lika system. Lösenordsfilerna som ska användas under denna laboration finns tillgängliga på URL:erna

`http://ver.miun.se/courses/infosakb/labs/win-pwd.txt`

och

`http://ver.miun.se/courses/infosakb/labs/unix-passwd.txt`.

Du behöver alltså inte använda något som `fgdump` eller `unshadow(8)` eftersom att hashvärdena redan är utvunna åt dig.

4.1.2 Ophcrack

`ophcrack(1)` använder en teknik som kallas regnbågstabeller (eng. *rainbow tables*). Denna metod går ut på att det i förväg har beräknats hashvärdena för väldigt många lösenord och sedan spara hashvärde och tillhörande lösenord. En hash för ett okänt lösenord kan då enkelt slås upp i denna enorma lista. Därför sparas enormt mycket tid vid knäckningen av lösenordet, det tar dock mycket resurser att skapa och sedan lagra själva tabellen. Helst ska hela tabellen få plats i datorns arbetsminne när den används. Dessutom fungerar denna metod enbart när lösenorden inte kombinerats med ett saltvärde innan det hashas, vilket är fallet med Windows. Unix-baserade system använder traditionellt saltvärden när lösenord lagras.

Du hittar `ophcrack(1)` i pakethanteraren för Ubuntu eller på

`http://ophcrack.sourceforge.net/`,

alternativt redan installerat om du sitter i sal L209, campus Sundsvall.

När du laddat hem programmet behöver du ladda hem några hashtabeller, dessa finns ej förinstallerade. Dessa finns på webbsidan given ovan under rubriken "Tables". Vid respektive tabell står även hur den har genererats. Välj tabell med omsorg.

4.1.3 John the Ripper

John the Ripper är ett terminalbaserat program som använder flera olika metoder för att knäcka lösenord. Bland annat används ordlistor, och det finns möjlighet att skriva regler för att exempelvis modifiera orden i ordlistan till att använda siffran noll istället för bokstaven O etcetera, samt ren forcering (eng. *brute force*). Dessa metoder tar naturligtvis längre tid att genomföra än om en regnbågstabell används, då alla beräkningar måste göras i realtid.

Du hittar programmet i pakethanteraren för Ubuntu eller på

`http://www.openwall.com/john/`,

eller redan förinstallerat om du är i sal L209, campus Sundsvall. Det är "Community enhanced version" som rekommenderas.

För att få en kort sammanfattning av argumenten till programmet kör du igång programmet genom att bara köra kommandot `john(1)` i terminalen utan några argument:

```
$ john
John the Ripper password cracker, version 1.7.8
Copyright (c) 1996-2011 by Solar Designer
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single                    "single crack" mode
--wordlist=FILE --stdin     wordlist mode, read words from FILE or stdin
--rules                     enable word mangling rules for wordlist mode
--incremental[=MODE]        "incremental" mode [using section MODE]
--external=MODE             external mode or word filter
--stdout[=LENGTH]           just output candidate passwords [cut at LENGTH]
--restore[=NAME]            restore an interrupted session [called NAME]
--session=NAME              give a new session the NAME
--status[=NAME]             print status of a session [called NAME]
--make-charset=FILE         make a charset, FILE will be overwritten
--show                      show cracked passwords
--test[=TIME]               run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..]   [do not] load this (these) user(s) only
--groups=[-]GID[,..]        load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]      load users with[out] this (these) shell(s) only
--salts=[-]COUNT            load salts with[out] at least COUNT passwords only
```

```
--format=NAME               force hash type NAME:
    DES/BSDI/MD5/BF/AFS/LM/crypt
--save-memory=LEVEL         enable memory saving, at LEVEL
    1..3
$
```

De intressantaste argumenten är `--show`, `--wordlist`, `--rules` och `--incremental` med mode *all*. Notera att `--wordlist` och `--stdin` är skilda argument. Den första läser ord från en fil, medan den senare läser ord från tangentbordet. Du kan läsa mer om dessa i manualsidan `john(1)` som du hittar genom att skriva `man 1 john` i terminalen.

Om du sitter i sal L209, campus Sundsvall, måste du skriva hela sökvägen till `john(1)`, det vill säga

```
$ /usr/sbin/john
[...]
$
```

Du hittar länkar till ordlistor att komplettera med på bland annat följande URL:

`http://sectools.org/tag/crackers/`,

under respektive verktyg.

4.2 Social engineering och APT

Du ska i denna uppgift hjälpa universitetets säkerhetsgrupp att granska säkerheten för social engineering-baserade attacker. Du har i uppdrag att ta fram ett realistiskt attackscenario för en social engineering-baserad attack som kan användas som exempel vid utbildning av universitetets personal.

Du har som inspiration i ditt arbete artikeln av Fisher [5], artikeln av Juels och Yen [7] och kurslitteraturen [1, kapitel 2]. Ha i åtanke gällande svensk lagstiftning, exempelvis offentlighetsprincipen. Spekulera kring hur en realistisk attack skulle kunna gå till.

Notera att detta *inte* är ett uppmuntrande till genomförande av denna attack! Detta är alltså en rent teoretisk uppgift. Du gör ditt bidrag till wikin kallad *Advanced persistent threats (APTs)* i lärplattformen.

5 Examination

Laborationen får genomföras i grupper om upp till två personer. För att examineras på denna laboration ska gruppen lämna in en rapport innehållandes:

1. Samtliga lösenord vars hashvärden bifogats på kurswebbplatsen samt en kort beskrivning av hur du gick tillväga.

2. Med utgångspunkt i dina reflektioner kring lösenordsstyrka, teorin, och tiden det tog att knäcka lösenorden, praktiken, ska du uppskatta styrkan hos de knäckta lösenorden och diskutera skillnader och likheter mellan teori och praktik.

3. Ditt social engineering-baserade attackscenario, du gör detta genom att bifoga en länk till avsnittet i wikin där du gjort ditt största bidrag.

Rapporten ska lämnas in i lärplattformen i PDF-format, med korrekta referenser, skriven på akademisk svenska eller engelska, och den ska vara i godkännbart skick.

Om rapporten inte uppfyller ovanstående krav eller inlämnas sent underkänns den med F, utan kommentarer, och får lämnas in igen vid nästa uppsamlingstillfälle.

Referenser

[1] Anderson, Ross J. *Security engineering : a guide to building dependable distributed systems*. Wiley, Indianapolis, IN, 2 utgåvan, 2008. ISBN 978-0-470-06852-6 (hbk.). URL `http://www.cl.cam.ac.uk/~rja14/book.html`.

[2] Bosk, Daniel. Grundläggande lösenordsanalys. URL `http://ver.miun.se/courses/infosakc/compendii/pwdanalysis.pdf`. 2013.

[3] Cluley, Graham. The worst passwords you could ever choose exposed by Yahoo Voices hack, jul 2012. URL `http://nakedsecurity.sophos.com/2012/07/13/yahoo-voices-poor-passwords/`.

[4] Cubrilovic, Nik. RockYou Hack: From Bad to Worse, dec 2009. URL `http://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/`.

[5] Fisher, Dennis. RSA: SecurID attack was phishing via an Excel spreadsheet. URL `https://threatpost.com/en_us/blogs/rsa-securid-attack-was-phishing-excel-spreadsheet-040111`. apr 2011.

[6] Hunt, Troy. A brief Sony password analysis, jun 2011. URL `http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html`.

[7] Juels, Ari och Yen, Ting-Fang. Sherlock Holmes and The Case of the Advanced Persistent Threat. I: *LEET*, 2012. URL `https://www.rsa.com/rsalabs/staff/bios/ajuels/publications/SherlockHolmes.pdf`.

[8] Kuo, Cynthia, Romanosky, Sasha, och Cranor, Lorrie Faith. Human Selection of Mnemonic Phrase-based Passwords. Teknisk rapport 36, Institute of Software Research, 2006. URL `http://repository.cmu.edu/isr/36/`.

[9] Oberheide, Jon. Brief analysis of the gawker password dump, dec 2010. URL `https://blog.duosecurity.com/2010/12/brief-analysis-of-the-gawker-password-dump/`.