Háskóli Íslands

Öryggi Tölvukerfa (HBV602M)

P455w0rd Cr4ck3r 3000

Authors:
Agnar Pétursson 300497-2929
Daníel Þór Guðmundsson
240297-2649

Professor:
Kristján Valur Jónsson

March 15, 2020

Contents

1 Inngangur

Í þessari skýrslu munum við fjalla um lykilorða brjótara (e. password cracker) sem við smíðuðum eftir beinagrind frá kennara. Ástæðan fyrir því að velja þetta tiltekna verkefni fram yfir önnur er sú að okkur þótti það það mun áhugaverðar heldur en skanninn. Það er sífellt í umræðunni um hversu mikilvægt það er að hafa sterkt lykilorð og hvað fólk stendur sig almennt illa í því. Við vildum því sjá hversu erfitt (eða auðvelt) það væri fyrir tvö amatöra eins og okkur að brjóta þekkt lykilorð sem margir eflaust nota.

Cracker-inn okkar notar tvær aðferðir til að brjóta lykilorð. Hann getur notað regnbogatöflur og/eða brute force aðferð. Búa þarf regbogatöflurnar sérstaklega til en við höfum útfært kóða og leiðbeiningar sem gerir það á handhægan máta.

Það finnst aragrúi af alls kyns cracker-um á netinu með notendavænu viðmóti. Einn af þeim kallast *John the Ripper* sem margir á allskyns net-spjallborðum virtust mæla með. *John* er þróað fyrir Linux umhverfi en hægt er að finna útgáfur fyrir önnur stýrikerfi. Hugbúnaðurinn sameinar nokkra cracker-a í einn pakka sem gerir hann nokkuð öflugan. Meðal aðferða sem *John* notar er ofbeldisaðferð (e. brute force) en hún snýst um að prófa alla möguleika. Þessi aðferð er bæði tímafrek og óhagstæð. Einnig er hægt að bæta orðabóksáras (e. dictonary attack) en hún byggir á því að vinna með lista af orðum (t.d. orðabók) og prófa hvert og eitt þeirra þangað til rétta lykilorðið finnst. [1]

2 Hönnun og aðferðarfræði

Áður en við byrjuðum á verkefninu lögðumst við á smá rannsóknarvinnu á því hvaða leiðir standa til boða þegar það kemur að því að brjóta hakkastrengi. Þær helstu eru eftirfarandi ásamt kostum og göllum:

- Nota regnbogatöflur
  - Tökum lista af þekktum lykilorðum, höshum þau og sjáum hvort við finnum það hash sem við ætlum að brjóta í töflunni. Helsti ókostur við regnbogatöflur er sá að það er tímafrekt að búa þær til og þær geta tekið mikið pláss. Kosturinn er hinsvegar að þegar þær eru tilbúnar kostar nánast ekki neitt að fletta upp í þeim.
- Orðabóks árás
  - Eins og fjallað var um í inngangi byggir þessi aðferð á því að nýta lista af alvöru orðum úr t.d. orðabókum og bera saman hashað lykilorð við hösh af orðum í orðabókinni. Þannig séð er þessi aðferð mjög svipuð regnbogatöflum og hefur því sömu kosti og galla. Annar galli við þessa aðferð er að mörg lykilorð nota tákn eða tölustafi inn á milli sem gera það að verkum að þau eru ekki til í orðabókum.
- Brute Force
  - Þessi aðferð snýst um að prófa allar mögulegar samsetningar tákna og tölustafa til að brjóta hakkastreng. Þannig séð mun þessi aðferð alltaf takast að brjóta alla hakkastrengi. Gallin er bara sá að spurningin er hvort það taki nokkrar mínútur eða áratugi.
- Social Engineering
  - Aðferðin snýst um að bæta samfélagslegri þekkingu ásamt hæfni í mannlegum samskiptum til að veiða upp upplýsingar um lykilorðið eða jafnvel lykilorðið sjálft upp úr fórnarlambinu. Í okkar tilfelli má segja að Kristján kennari sé fórnarlambið og töldum við hæpið að við gætum veitt upp úr honum meiri upplýsingar, svo að þessi möguleiki kemur ekki til greina.
- Hash-collisions
  - Hakkaföll eru "one-way" föll og ekki er hægt að snúa þeim við. Hinsvegar þá er möguleiki á því að tveir ólíkir strengir varpist yfir í sama hakkastreng og þá er um að ræða svokallað hash collision. Samvkæmt grein frá John D. Cook [2] er líkurnar á að það verði a.m.k. eitt hash

[1] https://en.wikipedia.org/wiki/John_the_Ripper
[2] https://www.johndcook.com/blog/2017/01/10/probability-of-secure-hash-collisions/

collision 40% sem er sláandi há tala. Hinsvegar þegar hakkastrengirnir eru orðnir margir bitar að lengd þá þarf að hasha gríðarlega margar strengi til að fá a.m.k. eitt collision. Svo að þessi aðferð er ekki hagkvæm í okkar tilfelli.

Eftir að hafa skoðað þessa möguleika vel ákváðum við að aðeins tveir þeirra kæmu mögulega til greina fyrir þetta verkefni. Regnbogatöflur og brute force.

Samvkæmt upplýsingum frá kennara vitum við hvernig lykilorðin eru byggð upp. Flest eru byggð á listanum passwords.txt sem inniheldur 10,000 lykilorð. Sum þeirra eru óbreytt, á sum hefur veirð beitt "leet-speak" umbreytingu og á önnur er búið að bæta við einu tákni aftast í lykilorðið. Það er afar auðvelt að búa til skriftu sem sér um að útbúa þessar útgáfur af lykilorðunum á fljótlegan máta. Hinsvegar er hluti af lykilorðunum búinn til á handahófskenndan máta. Þessi handahófskendu lykilorð eru 4-6 eða 8-12 stafir að lengd og eru 69 tákn sem koma til greina fyrir hvern staf í þessum lykilorðunum. Skoðum aðeins hvað þetta þýða mörg möguleg lykilorð:

Fjöldi tákna	**4**	**5**	**6**	**8**	**9**	**10**	**11**	**12**
Möguleikar	$2.27 * 10^7$	$1.56 * 10^9$	$1.08 * 10^{11}$	$5.14 * 10^{14}$	$3.55 * 10^{16}$	$2.45 * 10^{18}$	$1.69 * 10^{20}$	$1.16 * 10^{22}$

Við sjáum að fjöldi mögulegra lykilorða verður því fljótt afar há tala. Við vorum ekki vissir hversu hátt við gætum farið svo við ákváðum að byrja að kóða og sjá hvar við myndum lenda á vegg. Við notuðum MD5 hakkafallið ti að prófa okkur áfram til að byrja með.

Við byrjuðum á því að útfæra regnbogatöflu fyrir ósöltuð lykilorð. Við bjuggum til skriftu sem tekur inn lykilorða skrána passwords.txt sem skilar annari skrá þar sem búið er að bæta við leet-umbreytingu og táknum fyrir aftan ásamt fjögurra stafa handahófskenndum strengjum. Tölvan átti í litlum vandræðum með það svo við ákváðum að bæta við fimm stafa handahófskenndum strengjum. Þá kom fyrsti skellur. Tölvan var afar lengi að keyra og endaði það með því að hún kláraði minnið sitt og drap því forritið. Það var því ljóst að handahófskennd lykilorð lengri en 5 stafir voru ekki möguleikir miðað við minnispláss á tölvunum okkar.

Við prófuðum síðan að búa til regnbogatöflu úr þessari skrá og keyrðum cracker.py á hana. Niðurstöðurnar voru ágætar og tókst okkur að brjóta rétt rúmlega 40 lykilorð.

Við endurtókum síðan leikinn nema núna fyrir söltuð lykilorð og með 4 handahófskenndum lykilorðum. Í hverri skrá sem við eigum að bjóta eru 55 lykilorð og þar með 55 ólík sölt. Við þurfum því aðeins að prófa þessi 55 sölt en ekki öll sölt sem koma til greina. Við skrifuðum aðra skriftu sem tekur inn passwords.txt og eina saltaða lykilorða skrá sem skilar síðan annari skrá þar sem búið er að taka öll sölt úr söltuðu skránni og búa til mögulegar samsetningar af lykilorðum með söltum.

Næsta skref var þá að búa til regnbogatöflu út úr þessari lykilorða skrá. Við gerðum smá breytingu á rainbow.py svo að hún tæki tillit til þess að búið væri að salt lykilorðin áður. Þegar kom að því að keyra rainbow.py áttuðum við okkur hinsvegar á raunverulegri stærð á þessari regnbogatöflu. Samkvæmt grófum útreikningum okkar myndi þessi skrá verða hátt í 60GB að stærð. Þessi skrá passar hvorti í vinnsluminnið okkar né á harða diskinn. Svo að við sáum fljótt að þetta var ekki möguleiki. Við ákváðum þá í staðinn að prófa að sleppa handahófskenndum lykilorðum alveg og sjá hversu langt við kæmumst með það. Regnbogataflan fyrir það reyndist mun minni og viðráðanlegri eða rétt rúmlega 2GB.

Núna var komið að því að keyra cracker.py á söltuðum lykilorðum og sjá hversu langt við kæmumst. Niðurstaðan kom okkur verulega á óvart því við fengum ekki nema örfáum lykilorðum færra heldur en ósöltuð lykilorð sem innihéldu handahófskennd lykilorð. Það var því ljóst að það voru ekki nema örfá handahófskennd lykilorð sem voru 4 tákn að lengd.

Núna vorum við komnir með aðferð sem leysti bæði u.þ.b. 75% af bæði söltuðum og ósöltuðum lykilorðum sem verður að teljast ansi gott. Hinsvegar þá vildum við skoða hvort hægt væri að gera betur.

2.1 Brute force

Eins og við fengum að kynnast í seinasta kafla þá hafa regnbogatöflur sín takmörk út af minnisplássi. Það var því ljóst að ef við ætluðum að útfæra brute force aðferð þá þyrfti hún að vera hagkvæm þegar það kemur að minni. Þar sem brute force aðferðin er mun hægari heldur en það að fletta upp í regnbogatöflu þá ákváðum við að hanna crackerinn með það í huga að fyrst byrjar hann að leita í regnbogatöflum en reynir síðan að brjóta þau lykilorð sem eru eftir með brute force.

Brute force aðferðin okkar byggir á því að geyma vísa í lista af táknum. Þessir vísar eru síðan geymdir í lista og út frá listanum er hægt að búa til orð. Lengd listans er hámarkslengd orðs sem við viljum reyna að brjóta með brute force, þ.e. ef listinn er af lengd 6 þá reynum við aðeins að brjóta lykilorð sem eru 6 eða færri stafir að lengd. Best er að útskýra með dæmi hvernig þessi aðferð virkar:

- Við vitum að þau tákn sem koma til greina eru 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQR
  ?. Við getum geymt þessi tákn í lista og síðan vísað í þann lista til að fá eitthvað tiltekið tákn.
- Búum til annan lista sem hefur ákveðna lengd t.d. af lengd 5. Þessi listi inniheldur tölur frá -1 og til og með 68. -1 þýðir að það er enginn stafur í þessu sæti. Annars þá er talan vísun í táknalistann.
- T.d. ef listinn er [11, 13,68, 10, -1] þá myndi þetta varpast yfir í orðið **bd?a**.
- Þetta orð er síðan hashað og borið saman við þau hösh sem eru að finna í lykilorða skránni sem við ætlum að brjóta. Ef hashið finnst þar þá höfum við tekist að brjóta það. Ef lykilorðið væri saltað þá myndum við prófa allar samsetingar við söltin og gera það sama.
- Það sem gerist næst er að við uppfærum listann okkar með því að "hækka" hann um 1. Svo að listinn verður næst [11, 12, 68, 11, -1]. Þetta endurtekið sig síðan aftur og aftur. Ef við ætlum að reyna að hækka sæti sem inniheldur 68 þá verður það sæti að 0 og við hækkum sætið til vinstri um einn í staðinn. Þegar fyrst stakið í listanum verður hærra en 68 þá lengjum við orðið og setjum 0 í fyrsta sætið sem inniheldur -1. Þ.e. ef við höfum t.d. listann [68, 68, 68, 68, -1] og ætlum að hækka um 1 þá endum við með listann [0, 0, 0, 0, 0]

Þessi aðferð gerir það að verkum að við þurfum aldrei að geyma milliniðustöðurnar sem kostar þar af leiðandi ekkert aukaminni. Aftur á móti þá verður hún hinsvegar hægari. Við komumst að því að við getum brotið allt að 5 stafa handahófskennd ósöltuð lykilorð á viðráðanlegum tíma og 4 söltuð lykilorð.

3 Virkni og notkun

Crackerinn byggir að miklu leiti á því að beita regnbogatöflu árásum á lykilorðin. Svo að kjarni hans byggir á því að búa til þessar regnbogatöflur til að byrja með. Við skrifuðum 3 skriftur sem sjá um að búa þær til og gerðum breytingu á rainbow.py

- **passwordGenerator.py**
  - Þessi skrifta tekur inn skrá af þekktum lykilorðum og bætir við leet-umbreytingu af þeim, táknum afast við þau og fjögurra stafa handahófskenndum lykilorðum mögulega.
  - passwords [Textaskrá af lykilorðum sem nota á sem grunn]
  - output [Úttak úr forriti sett í þessa skrá]
  - -r [Bæta við fjögurra stafa handahófskenndum lykilorðum]
- **saltedPasswordGenerator.py**
  - Tekur inn in skrá af þekktum lykilorðum og skrá sem inniheldur höshuð söltuð lykilorð (users_salted.*) og skilar út nýrri skrá þar sem búið búa til allar útgáfur af lykilorðunum með söltum fyrir framan
  - passwords [Textaskrá af lykilorðum sem nota á sem grunn]
  - output [Úttak úr forriti sett í þessa skrá]

  - salt [söltuð lykilorðaskrá (users_salted.*]

- **rainbow.py**
  - Skráin er að mestu leiti óbreytt frá því áður nema að núna er búið að bæta við möguleikanum á því að að hakka lykilorð sem þegar er búið að bæta við salti.
  - -s [búið er að bæta við salti]

3.1 Brjóta ósaltaða skrá

Aðferðin er sú sama óháð því hvaða hakkafall er notað. Svo við sýnum virknina miðað við eitt fall og sambærilega aðferð má nota til að brjóta hin hakkaföllin.

Við byrjum á því að búa til öll möguleg lykilorð sem við viljum skoða og vistum í nýrri skrá passwordsAll.txt. Athugum að hér er -r rofinn virkur en þá er handahófskenndum fjögurra tákna lykilorðum bætt við.

- python3 passwordGenerator.py passwords.txt passwordsAll.txt -r

Næst búum við til regnbogatöflu út frá þessari skrá

- python3 rainbow.py passwordsAll.txt md5.rainbow -a md5

Að lokum skulum við keyra cracker.py þar sem hann byrjar á því að nota regnbogatöflu og skiptir síðan yfir í brute force. Crackerinn hefur nokkra rofa:

- -f [Password file]
- -r [rainbow table]
- -o [output file]
- -b [use brute force]
- -m [min brute force length] (Á aðeins við ef -b rofinn er notaður)
- -n [max brute force length] (Á aðeins við ef -b rofinn er notaður)
- -a [hash algorithm]
- -i [info]
- -v [verbose]

Brjótum lykilorðin með eftirfarandi skipun:

- python3 cracker.py -f users.md5 -r md5.rainbow -o users_md5.out -b -m 5 -n 5

Við náum að brute force brjóta 5 stafa lykilorð á skikkanlegum tíma. Niðurstöðurnar eru prentaðar út og einnig skrifaðar í nýja skrá, users_md5.out.

3.2 Brjóta saltaða skrá

Að brjóta saltaða skrá er ögn flóknara og krefst auka milliskrefs. Við byrjum á því að búa til þau lykilorð sem við viljum skoða nema að núna notum við engin handahófskennd lykilorð vegna skorts á minni (og tíma).

- python3 passwordGenerator.py passwords.txt passwordsAll_noRandom.txt

Næsta skref er að bæta við þeim söltum sem koma til greina við lykilorðin. Það er gert með saltedPasswordGenerator.py.

- python3 saltedPasswordGenerator.py passwordsAll_noRandom.txt passwords_salted.md5 users_salted.md5

Núna erum við komin með þau lykilorð sem við ætlum að skoða ásamt viðeigandi söltum. Núna getum við hafist handa við að búa til regnbogatöfluna. Við notum rainbow.py eins og áður nema að núna notum við rofann -s svo að það sé tekið tillit til saltsins.

- python3 rainbow.py passwords_salted.md5 md5_salted.rainbow -a md5 -s

Þá er ekkert annað eftir en að keyra cracker.py.

- python3 cracker.py -f users_salted.md5 -r md5_salted.rainbow -o users_salted_md5.out -b -m 4 -n 4 users_salted_md5.out -a md5

Við náum að brjóta 4 stafa slembin lykilorð með brute force á skikkanlegum tíma.

4 Niðustöður

4.1 Ósöltuð lykilorð

Í eftirfarandi töflum má sjá helstu niðurstöður og mælingar.

Skrá	Tími sem tekur að búa til (sek)	Stærð (kb)
passwordsAll.txt	18.44424	116,367
md5.rainbow	80.0413	1,234,710
sha1.rainbow	76.75404	1.508.590
sha256.rainbow	97.02094	2,284,583

Table 1: Tímamælingar á gerð nauðsynlegra skráa

Lykilorðaskrá	Fjöldi lykilorða	Fjöldi brotinna lykilorða	Hlutfall	Tími (sek)
users.md5	55	43	78.2%	7671.940212
users.sha1	55	43	78.2%	7185.566628
users.sha256	55	43	78.2%	8549.058721

Table 2: Niðurstöður fyrir ósöltuð lykilorð

Við náum að brjóta u.þ.b. 78% af öllum lykilorðunum. Það kom okkur hinsvegar á óvart að við náum að brjóta einum færri lykilorð í sha256 safninu. Skýringin á því er líklegast sú að skrárnar innihalda ólík lykilorð.

4.2 Söltuð lykilorð

Skrá	Tími sem tekur að búa til (sek)	Stærð (kb)
passwordsAll_noRandom.txt	0.329753	5,688
passwordsAll_salted.md5	16.56208	464,050
passwordsAll_salted.sha1	16.84038	464,050
passwordsAll_salted.sha256	16.65808	464,050
md5_salted.rainbow	123.466	2,354,581
sha1_salted.rainbow	133.7101	2,808,308
sha256_salted.rainbow	147.0524	4,093,869

Table 3: Tímamælingar á gerð nauðsynlegra skráa

Lykilorðaskrá	Fjöldi lykilorða	Fjöldi brotinna lykilorða	Hlutfall	Tími (sek)
users_salted.md5	55	42	76.4%	3870.878703
users_salted.sha1	55	42	76.4%	4434.170853
users_salted.sha256	55	41	74.5%	4422.230582

Table 4: Niðurstöður fyrir söltuð lykilorð

Við áttum vona á verri niðurstöðum fyrir söltuð lykilorð þar sem við erum ekki að nota nein slembin lykilorð sem eru meiri en 4 stafir. Það kom okkur því á óvart að fjöldinn sem við náum að brjóta er nánast sá sami, ekki nema 1-2 lykilorðum færri í öllum tilvikum. Þetta þýðir að það eru mjög fá 5 stafa slembin lykilorð í skránum.

4.3 Áhrif algríma og salta

Þar sem við nýttum okkur regnbogatöflu árasir og brute force til að brjóta lykilorðin urðum við lítið var við mun á erfiðleikastigi við að brjóta lykilorðin. Eini munurinn sem við fundum fyrir er sá að tíminn sem það tók að búa til hverja regnbogatöflu og skrár var lengri eftir því sem algrímið var flóknari þ.e. MD5 var styst (fyrir utan ósöltuð lykilorð), SHA1 í miðjunni og SHA256 lengst, og hver skrá varð að auki stærri.

Söltin höfðu mun meiri áhrif á okkur en algrímin. Eins og fjallað var um í hönnunarkaflanum þá lentum við strax í vandræðum þegar það kom að því að búa til regnbogatöflu út frá söltuðum lykilorðum. Við höfðum 55 sölt svo að söltuð lykilorða skrá varð þar af leiðandi 55 sinnum stærri en ósöltuð lykilorðaskrá. Þetta hafði þau áhrif að það var ekki hagkvæmt, hvorki fyrir tíma né minni miðað við okkar tölvur, að búa til regnbogatöflu með handahófskenndum lykilorðum. Crackerinn var síðan hlutfallslega mun lengur að keyra þar sem vinnan hans við hver lykilorð sem hann prófar var 55 sinnum meiri en fyrir ósöltuð lykilorð. Þetta sýndi hvað salt getur verið öflug leið til að gera vinnuna örlítið erfiðari fyrir vondu kallana.

4.4 Styrkur slembra lykilorða

Í þessu verkefni sjáum við svart á hvítu hversu hættulegt það getur verið að nota alvöru orð sem lykilorð. Fyrir reyndan aðila tæki það líklegast ekki nema augnablik til að brjóta það. Einnig sjáum við að þótt það sé búið að skipta stöfum út fyrir tákn þá þýðir það ekki að lykilorðið sé mikið öruggara. Hér kemur styrkur slembra lykilorða vel fram. Við lentum í miklu basli með slembin lykilorð, þá sérstakleg ef þau voru löng. Þetta sýnir að tegund lykilorðs og lengd þess hefur gríðarlega mikil áhrif á styrkleika þess.

5 Næstu skref

Eins og við fjölluðum um í upphafi skýrslunnar eru margar aðferðir sem geta komið til greina. Við ákváðum að nýta okkur regnbogatöflur en eins og við fengum að kynnast hér hafa þær sínar takmarkanir. Ef við hefðum meiri tíma ásamt öflugri tölvu væri gaman að útfæra "hagkvæmt" brute force cracker sem myndi einfaldlega prófa sig áfram þangað til hnan finnur rétta lausn. Það sem gæti hinsvegar gert brute force aðferðina þægilegri eru hash collisions.

Þrátt fyrir að líkurnar á collisions séu nánast hverfandi þá er möguleiki á þeim samkvæmt afmælis mótsæðunii (e. birthday paradox). Hægt væri þá að búa til gríðarlega mikinn fjölda af hakkastrengjum og þá með einhverjum líkum (mis miklir eftir tegund algríms) ætti a.m.k. eitt collision að eiga sér stað. Hinsegar, rétt eins og með brute force aðferðina þá kostar þetta mikinn tíma og minni.

6 Lokaorð

Í þessari skýrslu höfum við farið yfir aðferðafræðina sem við studdumst við til að brjóta gefið lykilorða safn. Við nýttum okkur regnbogatöflur og brute force aðferir til að brjóta lykilorðin og fengum við að kynnast styrkleikum og veikleikum þeirra. Okkur tókst að brjóta tæp 80% af ósöltuðum lykilorðum og í kringum 75% af söltuðum lykilorðum. Þau lykilorð sem við gátum ekki brotið voru slembin lykilorð að lengd 6 eða meira fyrir ósöltuð lykilorð en fyrir söltuð lykilorð gátum við aðeins brotið 4 stafa slembin lykilorð. Fjöldi mögulegra lykilorða sem komu til greina sem slembin lykilorð var of mikill til að hægt væri að búa til regnbogatöflur miðað við þau aðföng sem við höfum til staðar. Niðurstöðurnar okkar sýna að löng slembin lykilorð eru margfallt öruggari heldur en stutt þekkt orð.