UNIX Password Security

Walter Belgers

walter@giga.win.tue.nl
6 december 1993

Inleiding

Dit document is bedoeld om systeembeheerders te wijzen op het belang van goede passwords. Slechte passwords zijn vaak een mogelijkheid voor hackers[1] om een systeem binnen te dringen. Er zijn steeds meer computers op het wereldwijde Internet aangesloten (de meest recente schattingen spreken over zo'n anderhalf miljoen systemen). Dat houdt in dat er steeds meer gebruikers en ook meer hackers komen. Door middel van een goede password-beveiliging kunnen beginnende hackers geweerd worden.

Er zijn vele soorten systemen, en per soort systeem vele beveiligingsaspecten. Ik beperk me hier tot password-beveiliging van UNIX systemen. De reden hiervoor is dat UNIX systemen populair zijn, in het bijzonder in een educatieve omgeving, waar men een verhoogde concentratie hackers kan verwachten. Dat is het gevolg van de openheid die in een onderzoeksomgeving gewaardeerd wordt. Dit in tegenstelling tot een commerciële omgeving, waar gegevens beschermd moeten worden tegen (onder andere) concurrenten. Er zijn vele manieren om een UNIX systeem te hacken, en voor het vinden van passwords van gebruikers zijn verschillende programma's in omloop die gebruikt kunnen worden door mensen die weinig kennis van UNIX hebben. Goede passwords kunnen dus beginnende hackers van een systeem weren. ('Gevorderde hackers' kunnen vaak een systeem binnen dringen zonder gebruikmaking van passwords. Dat wil zeggen dat de beveiliging van een systeem van meer dan alleen goede passwords afhangt).

Naast het belang van goede (dat wil zeggen niet door een willekeurig persoon te raden) passwords komt in dit artikel ook de werking van passwords aan de orde. Daarna volgt een praktijkvoorbeeld waaruit blijkt dat het met beveiliging af en toe slecht gesteld is. Tot slot volgen er enkele methodes om een goed password te kiezen.

Het belang van goede passwords

Het doel van een hacker is meestal het verkrijgen van de superuser-status ('root'). Dat gebeurt normaal door gebruik te maken van slecht geïnstalleerde software, bugs in (systeem)software en menselijke fouten. Er zijn verschillende manieren om een computer te hacken zonder in te loggen, maar dat vereist enige kennis van zaken. Een (relatief) eenvoudige methode is inloggen als een reguliere gebruiker en dan het systeem afzoeken op bugs om zo de superuser-status te verkrijgen. Daarvoor zal de hacker dus eerst een geldige usercode/password combinatie moeten hebben.

Het is dus van belang dat alle(!) gebruikers op een systeem een password kiezen dat niet eenvoudig te raden is. De beveiliging van de gebruikers afzonderlijk heeft gevolgen voor de beveiliging

[1] 'cracker' is een betere benaming, daar 'hacker' van oudsher de benaming is voor iemand die veel uit zijn/haar computer haalt vanwege een grote kennis van de soft- dan wel hardware. Ik zal de term 'hacker' blijven gebruiken omdat dit algemeen gangbaar is

van het hele systeem. Gebruikers hebben vaak geen notie van de werking van een multi-user systeem en staan er niet bij stil dat zij door het kiezen van een gemakkelijk te raden password indirect een buitenstaander de mogelijkheid geven het hele systeem te manipuleren. Het is dus zaak om gebruikers goed voor te lichten om houdingen als beschreven in [Muf] te voorkomen: "It doesn't matter what password I use on **my** account, after all, I only use it for laserprinting…". De beveiliging van een systeem is ook een zaak van de gebruikers. Zo staat in [Pet]: "Users have a responsibility to employ available security mechanisms and procedures for protecting their own data. They also have a responsibility for assisting in the protection of the systems they use". Overigens staat in [Pet] óók dat het belangrijk is de gebruikers op de hoogte te stellen van de beveiligingsrichtlijnen. Een mogelijkheid is om nieuwe gebruikers een beperkte cursus te geven, of op zijn minst op de hoogste te stellen dat, en vooral **waarom** een goed password van belang is. Dat kan bijvoorbeeld gebeuren als de gebruiker zijn initiële password bij de systeembeheerder komt opvragen.

Hoe een hacker een password vindt

Op de meeste UNIX systemen wordt geen gebruik gemaakt van zogenaamde shadow password-files. De passwords worden versleuteld ('gecrypt') opgeslagen in de file `/etc/passwd`, of, als het een client betreft, op de server. Om dan aan de passwordfile te komen kan men het commando `ypcat passwd` uitvoeren.

Een regel uit deze passwordfile ziet er als volgt uit:

```
account:versleuteld password:uid:gid:GCOS-field:homedir:shell
```

Een gebruiker met account gigawalt, versleuteld password `fURfuu4.4hY0U`, userid 129 (een gebruiker met userid 0 (waar er meerdere van kunnen zijn) is superuser), groupid 129, informatie (GCOS) `Walter Belgers`, homedir `/home/gigawalt` en shell `/bin/csh` heeft zo'n entry in `/etc/passwd`:

```
gigawalt:fURfuu4.4hY0U:129:129:Walter Belgers:/home/gigawalt:/bin/csh
```

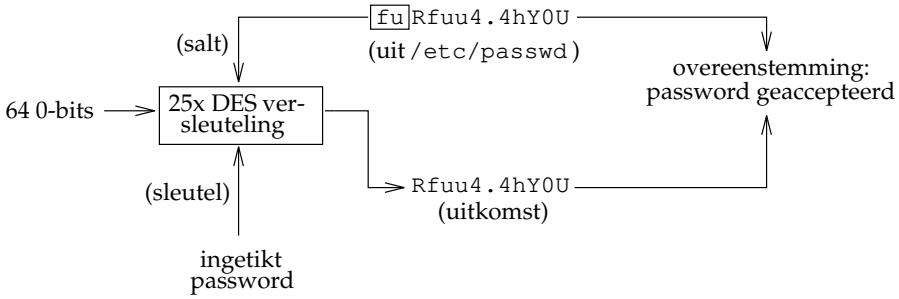
Als er een shadow passwordfile geïnstalleerd is ligt de zaak iets anders. Bij het gebruik van een shadow passwordfile staan alle gebruikelijke gegevens, behalve het versleutelde password, in `/etc/passwd`. In plaats van het versleutelde password staat er een `*`. In een tweede file (de shadowfile), die alleen met privileges te bekijken is, staan de versleutelde passwords. Hierdoor kan een normale gebruiker niet aan de versleutelde passwords komen.

Het password wordt versleuteld met een op DES gebaseerd algoritme. DES ('Data Encryption Standard') is een Amerikaanse encryptiestandaard sinds 1979. Met DES is het mogelijk gevegens te versleutelen en te ontcijferen met behulp van een sleutel. UNIX password-versleuteling gebruikt het DES algoritme 25 keer na elkaar. De eerste DES ronde gebruikt 64 0-bits als invoer en versleutelt die met het ingetypte password, waarbij tijdens het proces ook nog bits worden gepermuteerd op één van 4096 mogelijke methodes. Welke permutatie gebruikt wordt is willekeurig per gebruiker. De gebruikte permutatie wordt gecodeerd in twee bytes die men 'salt' noemt. De salt wordt in de passwordfile opgeslagen. De verkregen uitvoer wordt gebruikt als invoer voor de volgende DES ronde die weer het password als sleutel gebruikt en dezelfde permutaties uitvoert. Dit herhaalt zich tot er uiteindelijk een uitvoer komt na de 25^e DES ronde. Deze uitvoer wordt als elf karakters in de passwordfile gezet. In de passwordfile komen dus dertien bytes te staan, eerst de salt en daarna het versleutelde password.

Deze methode van encryptie is bijna onomkeerbaar, dat wil zeggen dat het gemakkelijk is een string te versleutelen, maar dat het onmogelijk is bij een op boverstaande wijze versleutelde string

het bijbehorende origineel te vinden, behalve door het systematisch afgaan van alle mogelijke sleutels en salts. Bij enkelvoudige DES-versleuteling is het overigens wèl mogelijk het origineel bij een versleutelde string te vinden. Voor meer informatie over enkelvoudige DES versleuteling zie [Til]. Als het niet mogelijk is het versleutelde password te ontcijferen, hoe kan een gebruiker dan inloggen? Dat gaat als volgt: de gebruiker typt zijn password dat als sleutel gebruikt wordt om 64 0-bits te versleutelen volgens de bovenstaande methode, met gebruikmaking van de salt zoals die in de passwordfile staat voor die gebruiker. Als de uitvoer overeenkomt met de elf bytes die het versleutelde password voorstellen in de passwordfile wordt het password goedgekeurd en krijgt de gebruiker toegang tot het systeem. Voor meer informatie over de precieze werking van deze versleutelingstechniek zie [Fel2].

Schematisch ziet het er als volgt uit:



Het is dus in praktijk niet mogelijk versleutelde passwords te decoderen. Maar het is wel mogelijk 64 0-bits te versleutelen met een aantal woorden en dan steeds te kijken of er 'toevallig' het gecrypte password uitkomt. Dan is het account gehacked. Men zou kunnen opmerken dat het zo mogelijk moet zijn om alle passwords te hacken door maar alle mogelijke lettercombinaties uit te proberen. Dit zou echter met de snelste computers nog langer duren dan de ouderdom van het heelal. Door zich te beperken tot alleen passwords met een lengte van zes tekens, die bestaan uit enkel kleine letters, kan men alle mogelijke combinaties echter wel weer in een redelijke tijd uitproberen, mits men de beschikking heeft over bijzonder snelle computers[2]. Passwords van voor een hacker aantrekkelijke accounts (de accounts met veel privileges) mogen dus nooit uit alleen kleine letters bestaan!

We zien dat het gebruikelijk is om passwords te vinden door aannemelijke passwords te proberen. Zorg er dus ook voor dat gebruikers geen voor de hand liggende passwords kiezen, dat wil zeggen passwords die in een file (woordenboek, encyclopedie, een file met astronomische termen, flora en fauna, etc.) staan. Via het Internet kan men zonder veel moeite aan zulke lijsten komen.

Stel dat een regel uit de password file er als volgt uitziet:

```
gigawalt:fURfuu4.4hY0U:129:129:Walter Belgers:/home/gigawalt:/bin/csh
```

Passwords die **niet** moeten worden gebruikt zijn dan onder andere:

- alle Nederlandse woorden (behalve 'lach' ook 'lachen', 'lachend', etc.)
- alle woorden uit een vreemde taal (men kan eenvoudig aan buitenlandse woordenboeken komen)
- woorden die in de passwordfile zelf voorkomen zoals Walter, Belgers, gigawalt, etc.
- toetscombinaties als 123456, qwerty, etc.

[2]Het laatste record voor passwords van zes kleine letters staat op één uur per gebruiker

- plaatsnamen
- woorden uit een encyclopedie ('Socrates')
- het nummerbord van een auto, het nummer van een kamer, een telefoonnummer of andere dingen die met de gebruiker te maken hebben
- eigennamen
- variaties hierop (walter, WALTER, retlaw, Walter, wAlter, walter0, walt3r, Retlaw4,...). Denk ook aan woordverdubbeling en het toevoegen van een willekeurig teken.

Een praktijkvoorbeeld

Om te laten zien hoe slecht passwords gekozen worden heb ik een programma om passwords te raden toegepast op een passwordfile van een operationeel systeem.

Het gebruikte programma was Crack v4.1 met ufcrypt (ultra-fast crypt, een snelle implementatie van het DES algoritme) op een netwerk van SUN ELC computers. De performance van deze computers (20 MIPS per machine) is vergelijkbaar met die van een hedendaagse PC. Het programma werd voortijdig afgebroken na bijna 60 uur. Alle gevonden passwords werden binnen de eerste 25 uur gevonden.

Resultaten:

Soort machines:	11x SUN ELC
Totaal aantal accounts:	521
Aantal gehacked:	58 (11,1%) (met interactieve shell 56 (10,7%))
Totale tijd	59:13 (dus echte tijd, geen CPU tijd)

1	woordenlijsten	42	(7,2%)
2	eigennamen	1	(0,2%)
3	user/account naam	5	(0,9%)
4	zinnen en patronen	3	(0,5%)
5	vrouwennamen	2	(0,3%)
6	mannennamen	4	(0,7%)
7	plaatsnamen	1	(0,2%)

Gevonden passwords:

1. cyclades, paardens, fiesta, regen, gnosis, police, fuselier, ballon, smaragd, marques, farao, kasteel, valent, adagio, clematis, gehannes, koeien, gnomen, onderkin, zeilboot, druppel, fietsen, testen, marathon, tamtam, global, vrijheid, wolf, kwiek, basket, stones, klomp9, fiets9, Zoutje, Biefstuk, neenee, tnbrg (dit is 'tonbrug' zonder klinkers).
2. fischer.
3. guest had password guest. Dit is natuurlijk geen fout van een user, maar van de systeembeheerders. Het is de vraag of inloggen op een account 'guest' met password 'guest' strafbaar is volgens de huidige Nederlandse wetgeving. Er is nog geen jurisprudentie over geweest. De wet spreekt over 'het doorbreken van enige beveiliging' (artikel 138a van het Wetboek van Strafrecht).

4. qwerty, unesco.
5. heather, joanne.
6. piet, atilla, Frans2, vatsug (dit is 'gustav' omgedraaid).
7. adelaide.

Er zijn mensen geweest die eerder al een studie hebben gedaan naar de hoeveelheid passwords die te kraken is zonder al te veel moeite. In [Kle] wordt door Daniel Klein 21% van 15.000 passwords gevonden na 1 week CPU-tijd. De eerste 2,7% werd gevonden binnen een kwartiertje (mensen die hun account ook als password gebruikten, bijv. account gigawalt, password gigawalt). De volgende categorieen leverden allemaal meer dan 1% van de 15.000 passwords op:

woordenlijsten	7,4%
eigennamen	4,0%
user/account naam	2,7%
zinnen en patronen	1,8%
vrouwennamen	1,2%
mannennamen	1,0%
machinenamen	1,0%

Deze resultaten vergelijken met de resultaten van hierboven heeft weinig zin vanwege de geringe omvang van mijn steekproef.

Een iets uitgebreidere steekproef (zie [Far]) had betrekking op password files van verschillende `.COM` systemen (computers van bedrijven in Amerika). Je zou verwachten dat commerciële bedrijven wat aan beveiliging doen, maar de passwords stroomden binnen, met een root-password(!) dat na iets meer dan een uur gevonden werd. (Het waren in totaal 1594 passwords, binnen een kwartier waren er al 50 van gebroken, na 35 minuten al 90).

Het kiezen van goede passwords

Het bovenstaande geeft het belang aan van een goed password voor elke gebruiker. Hier volgen enkele methodes om een goed password te kiezen. Een goed password bestaat uit acht tekens (een password kan in UNIX maximaal acht tekens lang zijn; alle extra tekens worden genegeerd, zodat de passwords 'Jantje zag eens pruimen hangen' en 'Jantje z' onderling uitwisselbaar zijn). Het moet moeilijk te raden, maar eenvoudig te onthouden zijn, omdat gebruikers anders geneigd zullen zijn het password op te schrijven waardoor de functie van het password volledig verloren gaat.

Kies een password dat niet alleen uit lowercase tekens bestaat, of alleen op één positie een hoofdletter heeft ('geHeim' is dus een slecht password). Gebruik liefst een niet-alfanumeriek teken in het password (%,=,*, etc.). Ook het gebruik van controletekens is mogelijk, maar niet alle controletekens kunnen gebruikt worden en het kan problemen geven met sommige netwerkprotocollen.

Een paar methodes:

– Neem twee woorden die samen zeven letters bevatten en die niets met elkaar te maken hebben. Zet die achter elkaar met een leesteken ertussen en vervang enige kleine letters door hoofdletters. Voorbeelden: 'Bak+laMp', 'baNK#hik'.

– Gebruik de eerste letters van woorden van een bepaalde zin. Als we als voorbeeld de zin 'Mijn twee goudvissen heten Justerini en Brooks!' nemen zou het password 'MtghJeB!' worden. (Zorg er ook hier voor dat het password acht tekens lang is en hoofdletters en/of leestekens bevat).

– Neem afwisselend een medeklinker en één of twee klinkers zodat er een uitspreekbaar (en dus eenvoudiger te onthouden) woord ontstaat. Voorbeelden: 'koEdupaN', 'eityPOop'.

Mogelijkheden

Het is van belang dat gebruikers een niet te raden en toch eenvoudig te onthouden password kiezen. Er zijn methodes om zulke passwords te genereren. Het is van belang dat systeembeheerders de gebruikers op de hoogte stellen van het belang van goede passwords.

Om het risico van een inbraak te verkleinen zijn er verschillende mogelijkheden:

– Zorg ervoor dat gebruikers weten waarom een goed password belangrijk is en hoe ze zo'n password kunnen kiezen.

– Installeer een nieuwe `/bin/passwd` (of `yppasswd`) die controleert of het password niet te voor de hand liggend is (door te controleren of er leestekens in zitten, of door te onderzoeken of het password voorkomt in standaard woordenlijsten).

– Installeer een shadow-password file (dit vergt aanpassingen van software).

– Laat passwords maar enige tijd meegaan, bijvoorbeeld drie maanden voor normale gebruikers en één maand voor gebruikers met extra privileges. De tijd dat een password meegaat moet niet te kort gekozen worden. Het gevaar blijft echter bestaan dat gebruikers rijtjes gaan gebruiken, zoals 'Geheim1', 'Geheim2',... zodat een hacker die eenmaal een password heeft gevonden kan voorspellen wat het volgende password zal zijn.

– Gebruik zelf een programma dat passwords hackt om te onderzoeken of er gebruikers zijn met een te raden password. Laat deze gebruikers *persoonlijk* langskomen om hen duidelijk te maken dat een goed password niet alleen in hun eigen belang, maar in het belang van alle gebruikers op het systeem is.

– Ga over op eenmalige passwords (dit is ingrijpend en er zijn kosten aan verbonden, zie [Ven]).

– Gebruik passwords van accounts met veel privileges zoals die van root alleen maar op de console om aftappen te voorkomen. Is dit niet haalbaar, vermijd dan in ieder geval het inloggen op zulke accounts vanaf computers of terminals die op een LAN segment zijn aangesloten waar mensen eenvoudig en/of anoniem het netwerk kunnen aftappen, zoals instructiezalen.

– Bedenkt dat de beveiliging van een systeem zo sterk is als de zwakste schakel. Bedenk ook dat een systeem met goede passwords alleen nog niet beveiligd is.

Referenties

[Bel] WALTER BELGERS, *Password Security – A Case Study*, TimeWasters Online Magazine #5, 9 maart 1993, op te vragen door email met Subject 'TOM5' naar `timewasters-request@win.tue.nl` te sturen.

[Cur] DAVID A. CURRY, *UNIX System Security*, Addison-Wesley 1992.

[Far] DAN FARMER, WIETSE VENEMA, *Improving the Security of Your Site by Breaking Into it*, USENET newsgroup `comp.security.unix`, op te vragen via anonymous ftp van `ftp.win.tue.nl` als `/pub/security/admin-guide-to-cracking.Z`, 1993.

[Fel1] DAVID C. FELDMEIER, *A High-Speed Software DES Implementation*, op te vragen via anonymous ftp van `thumper.bellcore.com` als `/pub/crypt/des.ps.Z`, 1989.

[Fel2] DAVID C. FELDMEIER, PHILIP R. KARN, *UNIX Password Security – Ten Years Later*, Proceedings of Advances in Cryptology – CRYPTO '89, 1989.

[Kle] DANIEL V. KLEIN, *"Foiling the Cracker": A survey of, and Improvements to, Password Security (revised paper)*, Proceedings of the USENIX Security Workshop, summer 1990.

[Muf] ALEC E. MUFFET, *Almost Everything You Ever Wanted To Know About Security (but were afraid to ask!)*, USENET newsgroup `alt.security`.

[Pet] R. PETHIA, S. CROCKER, B. FRASER, *RFC1281: Guidelines for the Secure Operation of the Internet*, november 1991.

[Til] HENK C.A. VAN TILBORG, *An Introduction to Cryptology*, Kluwer Academic Publishers, 1988.

[Ven] WIETSE VENEMA, *Using SecurID tokens in an open multi-host UNIX environment*, op te vragen via anonymous ftp van `ftp.nic.surfnet.nl` als `/surfnet/net-security/docs/securid.ps`, 1993.