OpenSCAP sigurnosno očvršćivanje i automatizacija sustava stvarnog vremena

Imširović, Leon

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:408302

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Diplomski sveučilišni studij Računarstvo

OPENSCAP SIGURNOSNO OČVRŠĆIVANJE I AUTOMATIZACIJA SUSTAVA STVARNOG VREMENA

Diplomski rad

Leon Imširović

Osijek, 2020.

Sadržaj

1. UVOD 1

1.1. Zadatak diplomskog rada 1

2. PRIMIJENJENE TEHNOLOGIJE I ALATI 2

2.1. Linux 2

2.1.1. Fedora 4

2.2. Bash 5

2.3. SCAP 6

2.3.1. OpenSCAP 7

3. REALIZACIJA SUSTAVA 8

3.1. Opis pravila 9

3.2. Prikaz rješenja 40

3.2.1. Sustav prije korištenja rješenja 45

3.2.2. Korištenje skripte 49

3.2.3. Sustav poslije korištenja rješenja 50

3.3. Testiranje rješenja 53

3.3.1. „SCAP Workbench“ rješenje 53

3.3.2. Usporedba rezultata 55

4. ZAKLJUČAK 58

LITERATURA 59

ABSTRACT 60

ŽIVOTOPIS 61

PRILOG A (Diplomski.sh) 62

1. UVOD

Danas na svakom koraku možemo vidjeti kako se sve oko nas mijenja. Brzina kojom se te promjene odvijaju od nas zahtjeva sve bolje i brže prilagođavanje novim uvjetima. Iz dana u dan može se primijetiti da sve više dolazi do sigurnosnih propusta i problema pretežno u smislu računalne sigurnosti. Iako svjesnost ljudi o informacijskoj sigurnosti postaje bolja, sama digitalizacija sve većeg broja sustava donosi razne nove izazove. Izazovi se mijenjaju od sustava do sustava s istim ciljem; a to je što bolja zaštita sustava od bilo kakvih neželjenih radnji. Informacije i brzina dobivanja istih smatraju se među najvažnijim resursima današnjice pa tako i sama zaštita informacija mora biti pravovremena. Zbog velikog broja sustava koji se iz dana u dan povećava i manjka kvalificiranog osoblja u području informacijske sigurnost gotovo je nemoguće napraviti kvalitetnu zaštitu bez neke vrste automatizacije koja će biti primjenjiva na veći broj sustava u isto vrijeme. Kako prijetnje postaju sve veće i kompleksnije, tako i sustavi obrane od tih napada moraju kombinirati i koristiti znanje iz više područja tehnologija kako bi se mogla održavati dovoljna razina zaštite.

U ovom diplomskom radu prikazana je kombinacija područja informacijske sigurnosti i automatizacije kako bi se sustav mogao osigurati od vanjskih prijetnji. Za samu provedbu odabrana je kombinacija s automatizacijom radi velike uštede vremena prilikom osiguravanja sustava i sigurnosti od neželjenih promjena prilikom promjene postavki na sustavu. Kako je preciznost i brzina prilikom osiguravanja sustava od velike važnosti, u radu se uz sustav većinom koriste već predefinirane mogućnosti i rješenja koja ne trebaju nikakve druge preinake za korištenje.

1.1. Zadatak diplomskog rada

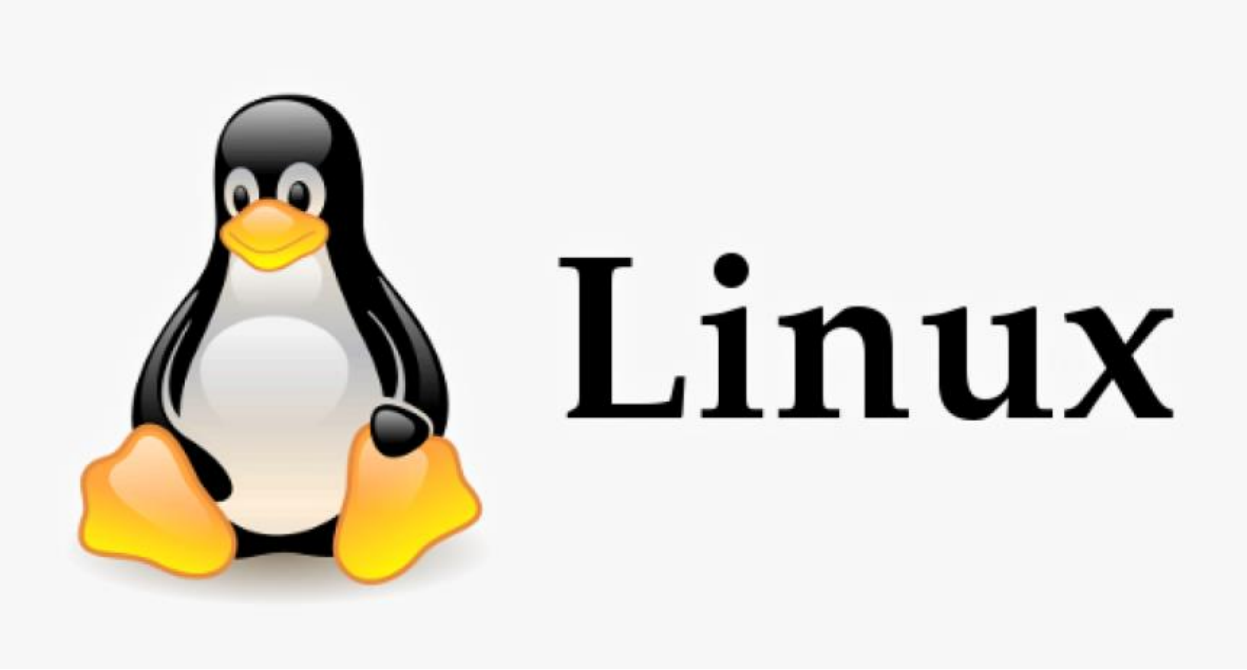
U radu je potrebno opisati i primijeniti sigurnosno očvršćivanje i automatizaciju jedne od Linux distribucija. Potrebno je prikazati razloge i načine korištenja OpenSCAP-a. Opisati najčešće sigurnosne prijetnje sustava te načine njihovog rješavanja uz pomoć automatizacije. Za rješavanje sigurnosnih prijetnji potrebno je koristiti shell programiranje i sve ostale metode potrebne za sigurnost sustava.

2. PRIMIJENJENE TEHNOLOGIJE I ALATI

U ovom poglavlju detaljnije su pojašnjeni i opisani alati i tehnologije koje su nužne za realizaciju diplomskog rada. Svaka od navedenih opisanih tehnologija koristi se tijekom cijelog rada pa su izdvojeni posebno kako bi se prvo detaljnije upoznali s često spomenutim pojmovima. Sve ostale tehnologije i alati koji su proizvoljno korišteni za rad pojašnjeni su u dijelovima rada u kojima su i korišteni.

2.1. Linux

Linux je jedan od najpopularnijih i najkorištenijih primjera softvera otvorenog koda što znači da se može modificirati i koristiti po volji. Jedna je od najvećih prednosti takvog softvera što se slobodno može dijeliti bez ikakvih restrikcija ako se koristi za vlastite svrhe i kasnije dijeli ostalima. Osim za sam operativni sustav ime Linux koristi se i za jezgru. Jezgra operacijskog sustava centralni je dio operativnog sustava koji služi za upravljanje sklopovljem na najnižoj i za povezivanje te razine s aplikacijama i ostalim uslugama na najvišoj razini sustava. Linux se ponekad naziva i GNU/Linux zato što većinu svojih biblioteka i sistemskih alata koristi iz GNU operativnog sistema.



Slika 2.1. Prikaz prepoznatljivog Linux loga[1]

[1] Preuzeto s https://www.pngitem.com/middle/xxwiww_linux-penguin-logo-png-linux-logo-png-transparent/

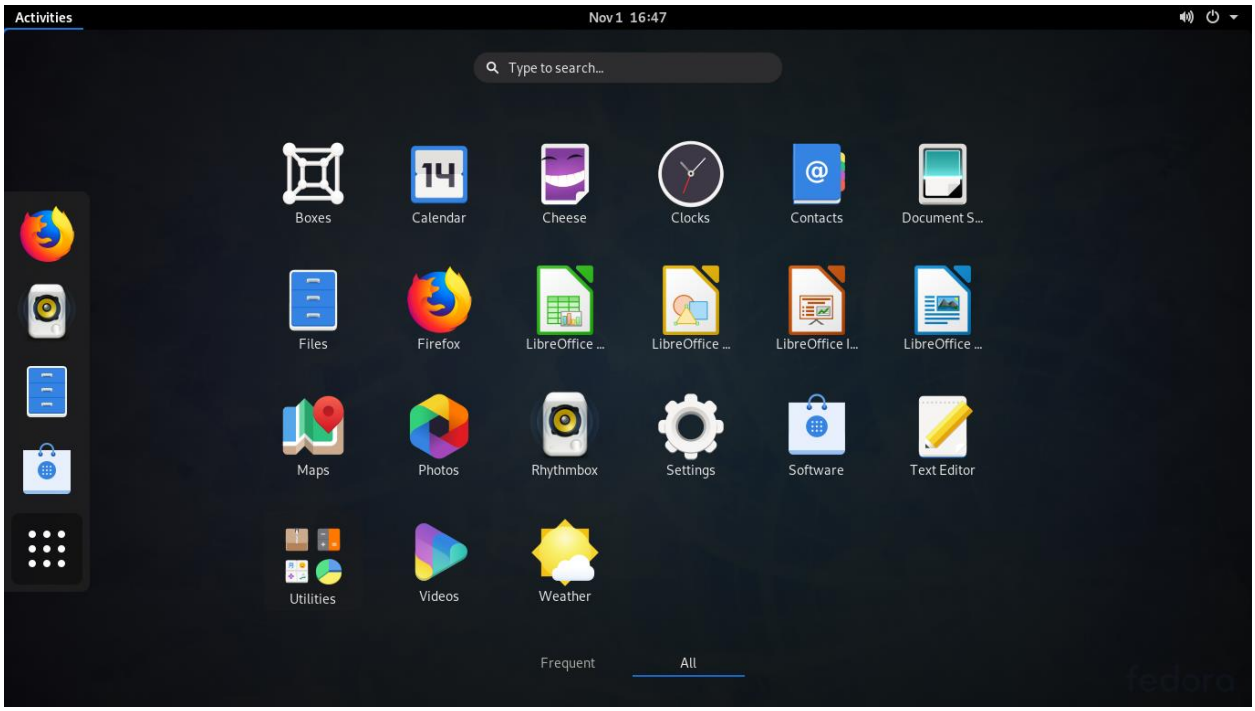
Linuxom se može upravljati na više načina. Dva su najkorištenija načina upravljanje uz pomoć grafičkog sučelja (GUI)[2] i upravljanje putem naredbenog retka (CLI)[3]. GUI se najviše koristi na stolnim računalima dok se CLI najviše koristi na serverima. Većina Linux komponenti koristi isključivo naredbeni redak. Posljedica toga korištenje je naredbenog retka i u slučaju kada je grafičko sučelje dostupno. Zato je najveća upotreba Linuxa je na poslužiteljskim stranama sustava kako bi se zadatci koji se periodički ponavljaju mogli automatizirati. Kako je Linux postajao sve popularniji, tako su nastajale nove verzije operacijskog sustava koje koriste Linux jezgru. Sve nove operacijske sustave nastale na Linux jezgri nazivamo distribucijama. Jedne od najpoznatijih distribucija su Fedora, Ubuntu i Debian. [1]

Slika 2.2. Prikaz najpoznatijih Linux distribucija[4]

[2] „GUI“ - graphical user interface
[3] „CLI“ - command line interface
[4] Preuzeto s http://nuttkracker.com/these-are-the-best-5-linux-distributions-for-beginners/

2.1.1. Fedora

Fedora je jedna od najpopularnijih distribucija Linuxa napravljena od strane „Red Hat“ multinacionalne softverske kompanije. „Red Hat“ ulaže u daljnji razvoj Fedore preko „Fedora Project“ projekta. Fedora dosta ulaže u nezavisne projekte kako bi ih u slučaju uspjeha mogla implementirati u operativni sustav. Tako Fedora uvelike pomaže i ostalim distribucijama. Zadano je korisničko sučelje na Fedora operativnom sustavu GNOME iako sva ostala mogu biti dodatno instalirana i dodana na sustav. Većina Fedora izdanja koristi RPM[5] sustav upravljanja paketa i DNF[6] alat prilikom upravljanja tim paketima. Uz Fedoru dolaze i neki predefinirani alati poput „LibreOffice“ i „Firefox“ alata koji olakšavaju korištenje Fedore svakodnevnim korisnicima. Svi ostali paketi i alati koji su dostupni na repozitorijima mogu se instalirati preko DNF alata ili preko GNOME softvera za instalaciju paketa. [2]

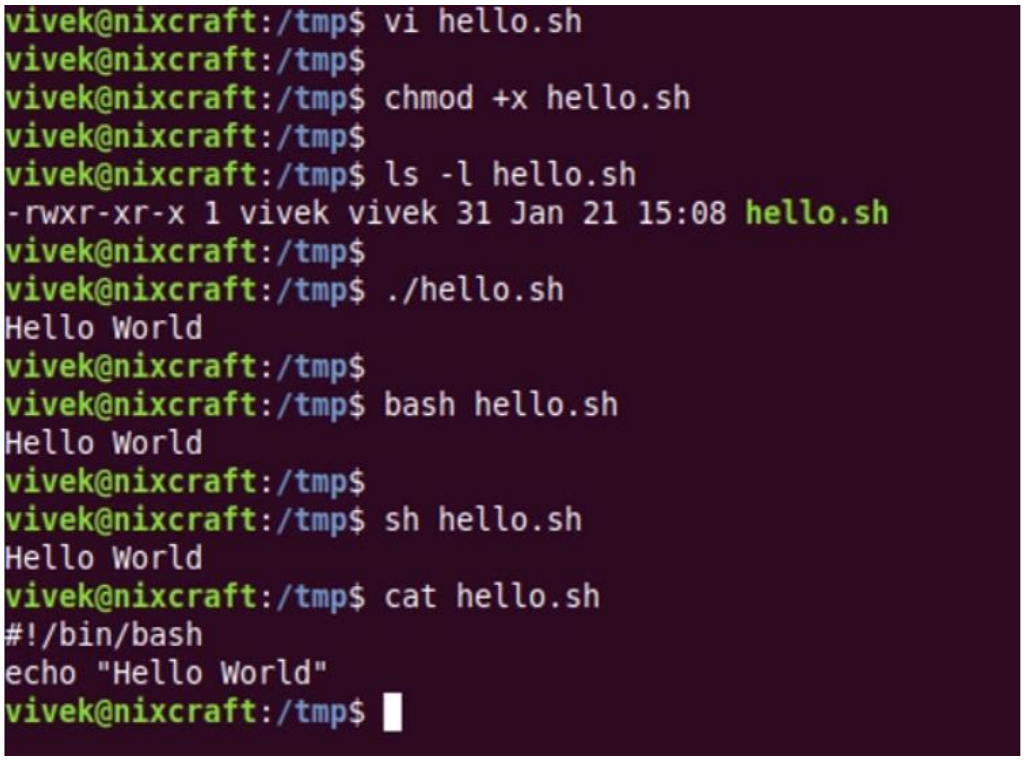


Slika 2.3. Prikaz „Fedora 31“ i korisničkog sučelja[7]

[5] „RPM“ - Red Hat Package Manager
[6] „DNF“ - Dandified YUM
[7] Preuzeto s https://en.wikipedia.org/wiki/Fedora_(operating_system)#/media/File:Fedora_31_(2019-10)_with_default_desktop_and_applications.png

2.2. Bash

Ljuska (engl. shell) je interaktivan alat kojim se služi za pretvaranje korisničkih naredbi u jezik razumljiv sustavu. Osim korištenja naredbi ljuska ima sve potrebno da bi se koristila kao i programski jezik. Bash[8] je najkorištenija te pretpostavljena ljuska na Linux operativnim sustavima. Izvedivi program kojim možemo pokrenuti i time aktivirati navedenu ljusku također se naziva „bash“. Jedan od najvećih razloga tolikoj popularnosti Bash ljuske mogućnost je editiranja naredbi direktno u naredbenom retku što uvelike pomaže prilikom ispravljanja određene naredbe ili njenog nadopunjavanja.[3]



Slika 2.4. Prikaz korištenja Bash ljuske[9]

[8] „Bash“ - Bourne Again SHell
[9] Preuzeto s https://www.cyberciti.biz/faq/hello-world-bash-shell-script/

2.3. SCAP

SCAP[10] je protokol automatizacije sigurnosnog sadržaja pokrenut od strane Nacionalnog instituta za standarde i tehnologiju (NIST)[11] i njihovih partnera. Koristi posebne standarde kako bi se osigurala što veća sigurnost sustava uz pomoć automatizacije. Svrha je SCAP-a primijeniti postojeće i zadane sigurnosne standarde na sustave koji nemaju pravu zaštitu. Aplikacije koje provode sigurnosno praćenje koriste standarde prilikom mjerenja sustava za pronalaženje ranjivosti i nude metode za ocjenu tih nalaza kako bi se procijenio mogući utjecaj. SCAP se sastoji od dva glavna dijela, a to su SCAP sadržaj i skener. Moduli SCAP sadržaja javno su dostupni sadržaji koji su napravljeni na temelju često viđenih sigurnosnih konfiguracija i povezanih znanja stručnjaka u području sigurnosti. Takvi moduli koriste već prije osigurane sustave kako bi mogli usporediti dobivene rezultate u odnosu na željeni sustav. Alati koji uspoređuju takve rezultate nazivaju se skenerima. Skener koristi takvu usporedbu kako bi zabilježio sve moguće nesigurnosti i sastavio izvještaj.

Slika 2.5. Prikaz NIST glavnih funkcija[12]

[10] „SCAP“ - Security Content Automation Protocol
[11] „NIST“ - National Institute of Standards and Technology
[12] Preuzeto s https://www.forescout.com/company/blog/how-to-comply-with-the-5-functions-of-the-nist-cybersecurity-framework/

2.3.1. OpenSCAP

Neke implementacije SCAP protokola sadržavaju i moguća rješenja koja mogu sam sustav uz pomoć automatizacije usmjeriti prema pravilnom standardiziranju. Jedna je od takvih implementacija i OpenSCAP koji je nastao kao rješenje „Red Hat“ kompanije. OpenSCAP je alat za reviziju sigurnosnih politika sustava te koristi razne određene standarde kako bi provjeravao mjeru sigurnosti na sustavima. Uz to što provjerava određene sigurnosne mane, postoje razni načini na koje OpenSCAP ih nakon skeniranja pokušava i riješiti. Velika je prednost OpenSCAP alata što može skenirati od raznih Linux distribucija (Fedora, Ubuntu, Centos...), pa sve do raznih usluga i alata (FireFox, Chromium…). Skeniranje se može odvijati preko naredbenog retka ili grafičkog sučelja pa tako postoji mogućnost skeniranja raznih sustava. [4] OpenSCAP se dijeli na svoje pod alate:

1. **OpenSCAP Base**: alat pomoću kojeg se odvija skeniranje konfiguracija i ranjivosti sustava. Prikazuje osnovne informacije sustava i moguće profile skeniranja sustava.

2. **OpenSCAP Daemon**: usluga koja se odvija u pozadini sustava. Provjerava da se određeni sustavi i uređaji skeniraju prema dogovorenom rasporedu. Funkcionalnost ove usluge može se podijeliti na dva dijela: jedno je kontinuirano praćenje i ocjenjivanje sustava, a drugo se odnosi na jednokratno ocjenjivanje sustava bez dodatnih naknadnih praćenja.

3. **SCAP Workbench**: alat s grafičkim sučeljem koji nudi jednostavan način za lokalno ili udaljeno skeniranje računala. Dodatna je prednost ovog alata to što može generirati izvješća sustava u više formata te ih prikazati na jednostavan način.

4. **SCAPtimony**: SCAP poslužitelj koji prikazuje određene dijelove skeniranog sustava uz pomoć baze podataka te nudi mogućnost manipuliranja nad tim podatcima.

5. **OSCAP Anaconda Add-on**: dodatak za instalaciju koji administratorima omogućuje unos sigurnosnih pravila u postupak instalacije i osigurava da su sustavi usklađeni od prvog pokretanja.[5]

3. REALIZACIJA SUSTAVA

Kako je tema ovog rada sigurnost i automatizacija sustava, u ovom djelu rada prikazana su i opisana pravila za zaštitu sustava te načini na koje se ta pravila trebaju riješiti uz pomoć automatizacije. Rješenje se temelji na Fedora Linux distribuciji pa je sukladno tome odabrano i 80 standardnih pravila za zaštitu tog sustava. Pravila su jedan od osnovnih profila OpenSCAP-a te se mogu dodatno uređivati po volji. Sva pravila definirana su dogovorenim sigurnosnim standardima. Svako je pravilo numerirano u kodu i tim su redom pravila detaljnije objašnjena.

Kôd je pisan u Bash-u koji je detaljnije objašnjen u prethodnom djelu rada (2.2.). Svako pravilo ima svoje opisno ime te izvorno ime koje se nalazi odmah uz opisno u zagradi. Izvorno je ime ostavljeno kako bi se u potpunosti moglo provjeriti samo pravilo. Osim numeriranja svakog pravila dodani su i brojevi početka i kraja tako da prvi broj u zagradi označava početnu liniju koda opisivanog pravila, a drugi je broj završna linija dijela koda koji se odnosi na to pravilo.

Primjer: (2-10) - Označava da kôd za opisivano pravilo počinje na drugoj liniji priloga, a završava na desetoj liniji priloga. Sam kôd na koji se referencira se nalazi pod nazivom Prilog A (Diplomski.sh) u zadnjem djelu rada.

Nakon opisa svih pravila nalazi se detaljno objašnjen način korištenja alata i skripte koja pomoću automatizacije rješava sva navedena pravila. Također, radi bolje usporedbe rezultata same skripte, rezultati su uspoređeni s trenutno službeno najboljim smatranim rješenjem koje se koristi za ovu skupinu pravila. Usporedba je odrađena u odnosu na sveukupnu ocjenu sigurnosti sustava te kompleksnosti i veličini napisanog rješenja.

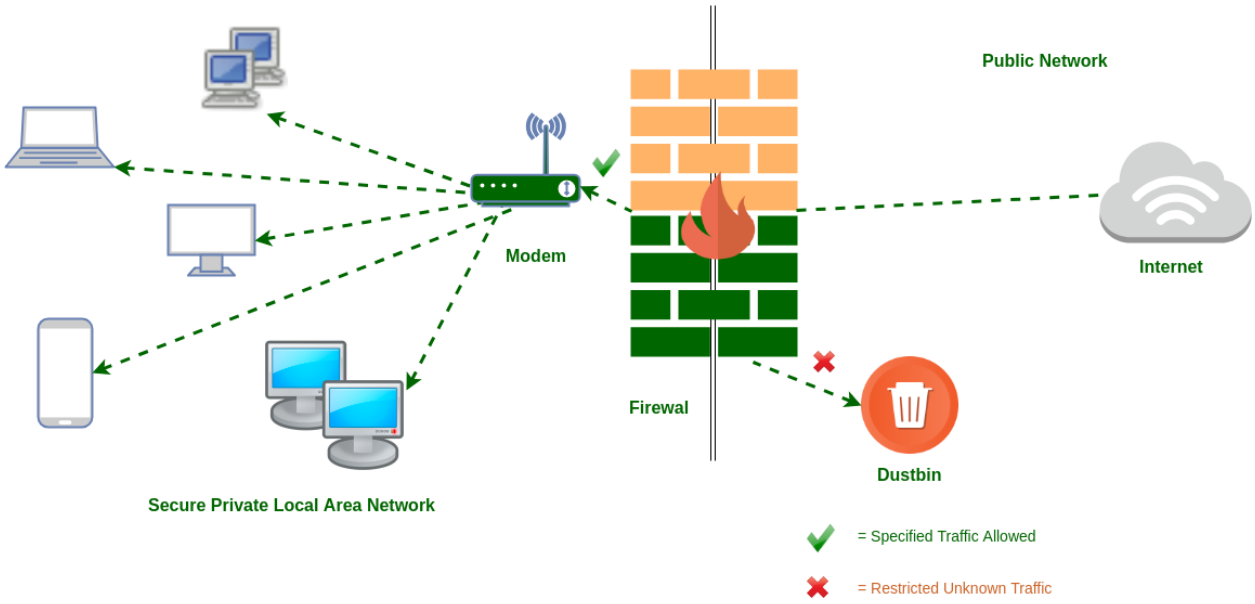
3.1. Opis pravila

3.1.1. Postavljanje zadane vatrozidne zone za dolazne pakete (3-10) (Set Default firewalld Zone for Incoming Packets)

Vatrozid je uređaj za provjeru prometa koji prolazi kroz njega, a na temelju zadanih ili proizvoljnih pravila taj promet propušta ili zaustavlja. Tim pravilima većinom upravlja mrežni administrator ili neka druga ovlaštena osoba. Paketi se mogu podijeliti na odlazne i dolazne. Kod odlaznih paketa većinom nema prevelikih zabrana pošto prilikom odlaznih paketa nema toliko sigurnosnih prepreka koje bi mogle utjecati na sam sustav dok su s druge strane dolazni paketi od velike važnosti radi toga što uvelike mogu utjecati na sustav i njegovu sigurnost. Firewalld je uslužni program za upravljanje vatrozidom kojim možemo definirati zone pomoću kojih možemo upravljati pravilima. Jedna je od velikih prednosti firewalld-a ta što se sve promjene mogu izvršavati u stvarnom vremenu bez potrebe za ponovnim pokretanjem uslužnog programa. Kako firewalld koristi zone koje definiraju razinu povjerenja, tako svaka zona ima svoja posebna pravila pomoću kojih propušta ili odbacuje pakete. Iz sigurnosnih razloga najbolje je zaštititi sustav tako da ne može na nikakav način biti oštećen pa se tako u ovom slučaju radi najveće sigurnosti postavlja zona „drop“ koja ima najnižu razinu povjerenja, a to znači da se svi dolazni paketi automatski odbacuju i omogućuju se samo odlazni paketi.[6]

3.1.2. Omogućavanje i pokretanje vatrozida (11-36) (Verify firewalld Enabled)

Za sigurnost bilo kakvog sustava povezanog na internet od velike je važnosti vatrozid pa je tako od samog pokretanja sustava jako bitno da je vatrozid pokrenut. Jedan je od najboljih načina da budemo sigurni da će vatrozid biti uključen da postavimo da se vatrozid automatski pokrene prilikom pokretanja samog sustava. Tako dobivamo sigurnost od samoga početka. Pošto se promjene uz pomoć firewalld uslužnog programa mogu primijeniti u stvarnom vremenu, rješenje se primjenjuje odmah nakon izvršavanja tog dijela koda, pa tako sustav u što kraćem vremenu može doći do bolje sigurnosti samog sustava.



Slika 3.1. Grafički prikaz vatrozida[13]

3.1.3. Zabrana pristupa priključenim USB uređajima (37-57) (Disable Kernel Support for USB via Bootloader Configuration)

U današnje vrijeme velike količine podataka prenose se na razne načine. Iako se većina podataka danas prenosi preko interneta, jedan od najkorištenijih uređaja današnjice za prijenos podataka je USB[14]. U toj velikoj količini podataka uvijek se može naći neka zaražena datoteka ili virus koji može naštetiti samom sustavu. Najsigurniji je način da se zaštiti od takvih mogućih napada onemogućiti sve USB-ove na razini jezgre. Kako takav način onemogućuje sve USB-ove, treba pripaziti da se u to računaju i uređaji koji se koriste svakodnevno poput tipkovnice i miša.

3.1.4. Postavljanje „root“ vlasnika za izvršne datoteke (58-71) (Verify that System Executables Have Root Ownership)

Svaka datoteka ili mapa ima svog vlasnika. Korisnik koji je stvorio neku datoteku ili mapu većinom je i njen vlasnik. Taj isti korisnik ako je vlasnik može mijenjati prava pristupa za ostale korisnike. Svaka od tih datoteka pripada nekoj skupini korisnika pa tako korisnici te skupine isto

[13] Preuzeto s https://www.geeksforgeeks.org/types-of-firewall-and-possible-attacks/
[14] USB – „Universal Serial Bus“

imaju određena prava na tu datoteku. Na Linuxu postoje 3 razine dodjeljivanja prava (u(user)-, g (group) i o (others)):

u - prava na datoteku ili mapu koja se odnosi na vlasnika.

g - prava na datoteku ili mapu koja se odnosi na skupinu.

o - prava na datoteku ili mapu koja se odnosi na ostale.

Izvršne datoteke na Linux-u odmah su spremne za korištenje, a sama radnja koju te datoteke mogu izvršavati svakodnevno može i naštetiti samom sustavu. Zbog toga je preporučeno da samo „root“ korisnik bude vlasnik tih datoteka. Tako se sprječava manipuliranje tim datotekama od strane drugih korisnika i smanjuje mogućnost same štete za sustav.

3.1.5. Postavljanje „root“ vlasnika za dijeljene bibliotečne datoteke (72-82) (Verify that Shared Library Files Have Root Ownership)

Dijeljene bibliotečne datoteke uvelike pomažu prilikom stvaranja neke aplikacije ili programa. One sadrže informacije koje može koristiti više programa kako svaki program ne bi morao te značajke imati zasebno. Na primjer jedna takva datoteka može sadržavati informacije i funkcije o tome kako pretražiti računalo u cijelosti. Tako više programa može pozvati tu datoteku i koristiti te značajke u vlastitim programima. Mijenjanje prava na takvim datotekama nije poželjno, pa se za vlasnika takvih datoteka treba postaviti „root“. Samim time omogućujemo da takve datoteke imaju što manje promjena.

3.1.6. Uklanjanje dozvola za dijeljene bibliotečne datoteke (83-99) (Verify that Shared Library Files Have Restrictive Permissions)

Korisnicima se mogu dodijeliti različita prava pristupa na datoteke ili mape:

r - pravo čitanja (read).

w - pravo izmjene sadržaja (write).

x - pravo izvršavanja (execute).

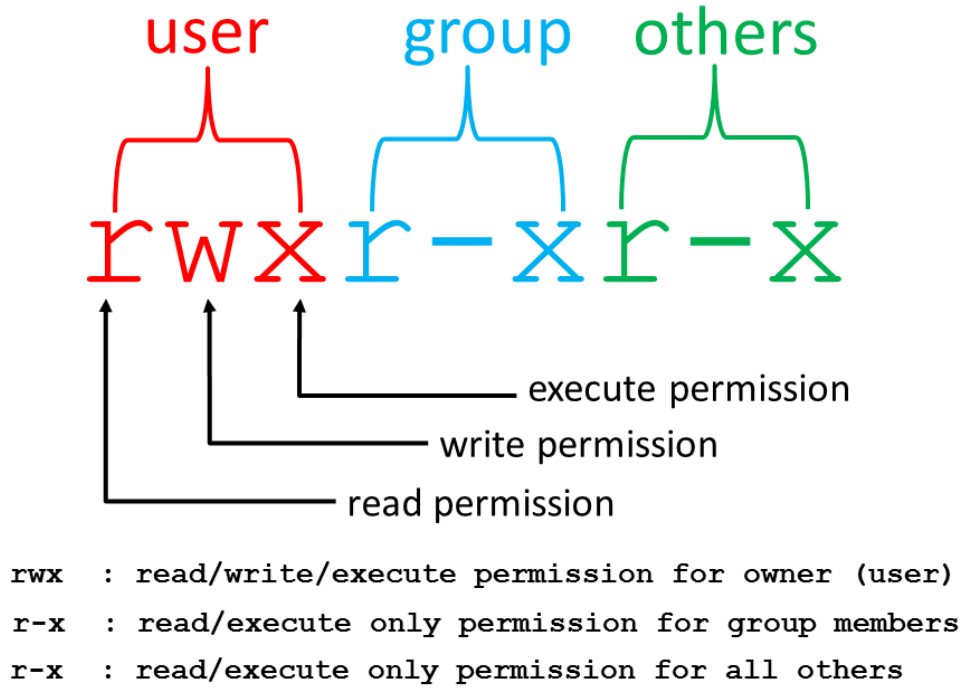
Svako pravo na datoteke daje više ovlasti za nekog korisnika pa ako korisnik ima više prava nego je preporučeno, dolazi do manje sigurnosti sustava. U slučaju dijeljenih bibliotečnih datoteka

pravo na čitanje i izvršavanje ne predstavljaju toliku prijetnju dok pravo na izmjenu sadržaja treba imati samo vlasnik same datoteke. To znači da se prava za izmjenu sadržaja moraju ukloniti za sve skupine i ostale korisnike.

3.1.7. Uklanjanje dozvola za izvršne datoteke (100-122)

(Verify that System Executables Have Restrictive Permissions)

Iako postavljanje „root“ korisnika kao vlasnika daje jednu vrstu sigurnosti, to nije dovoljno. „Root“ korisnik kao vlasnik jedini je koji može mijenjati prava pristupa izvršnim datotekama. Potrebno je provjeriti koja prava imaju korisnici skupina i ostali korisnici. U slučaju da ti korisnici imaju prava mijenjanja sadržaja, ta prava potrebno je odmah ukloniti. Tako štitimo izvršne datoteke da imaju samo svoju definiranu funkciju bez ikakvih problema i zloupotrebe.



Slika 3.2. Prikaz prava korisnika[15]

[15] Preuzeto s http://condor.liv.ac.uk/unix_guide/unix8.htm

3.1.8. Provjera autentičnosti za sve „dnf“ pakete (123-129)
(Ensure gpgcheck Enabled for All dnf Package Repositories)

„Dnf“ spada pod skupinu instalacijskih paketa. Većinom se koristi na Centos i Fedora Linux distribucijama. Specifičnost je takvih paketa da skraćuju i olakšavaju način potrebne instalacije. Radi jednostavnosti određenih instalacija potrebno je da ti paketi imaju i provjeru potpisa. Tako možemo biti sigurni da možemo sigurno koristiti te pakete. Provjera autentičnosti na ovakav način je jako važna radi kasnijih ažuriranja samih paketa i instalacija. Zato je potrebno postaviti ograničenja koja će omogućiti instalaciju samo provjerenih paketa na sustav.

3.1.9. Provjera autentičnosti paketa za glavnu “dnf” konfiguraciju (130-136)
(Ensure gpgcheck Enabled In Main dnf Configuration)

Provjera potpisa svakog paketa nužna je za sigurnost pa je tako bitna i pravovremena provjera tog potpisa. Osim što je potrebno da se onemogući instalacija paketa koji nemaju valjani potpis, tako je od velike sigurnosne važnosti da je taj potpis provjeren već prije instalacije. Tako možemo u potpunosti biti sigurni da možemo instalirati željeni paket.

3.1.10. Zaštita od nesigurnog ubrzavanja programa (137-143)
(Disable Prelinking)

Velika količina vremena troši se na sustavima prilikom pokretanja programa. Zbog toga sustavi s velikim opterećenjima većinom koriste programe koji omogućavaju smanjivanje vremena samog pokretanja tako da se pamte sve potrebne konfiguracije pokrenutih programa i spremaju radi uštede vremena. Tako svaki program ima svoju konfiguraciju spremljenu prije pokretanja i ne mora raditi novu svaki puta kada je sam program pokrenut. Iako takav način pokretanja donosi veliku uštedu vremena, sigurnosno nije prihvatljiv. Takvi programi ujedno mijenjaju i izvršne datoteke pa samim time cijeli sustav može biti kompromitiran i nesiguran. Najsigurniji način da ne dođe do toga je taj da se takav način ubrzavanja programa u potpunosti onemogući.

3.1.11. Stvaranje i testiranje AIDE baze podataka (144-155) (Build and Test AIDE Database)

AIDE[16] je program koji provjerava integritet mapa i datoteka na Linuxu. Radi tako da prvo stvori bazu podataka sustava u željenom i sigurnom stanju te nakon toga tu istu bazu podataka koristi za bilo kakve promjene na datotekama i moguće upade u sustav. „AIDE“ većinom nije instaliran na Fedori pa ga je potrebno instalirati uz pomoć „dnf“ paketa. Nakon same instalacije potrebno je napraviti inicijalnu bazu podataka koja će sustav spremiti u željenom stanju i trenutku. Takva baza podataka služi kao osnova za sve druge promjene i nadogradnje na sustavu. Prilikom stvaranja nove baze podataka ime je predefinirano s oznakom „.new“ koju je potrebno maknuti da sustav zna koju bazu podataka trenutno koristi i da se prilikom stvaranja nove baze podataka ne pregaze stare promjene. Nakon stvaranja baze podataka željenog stanja, sustav se može skenirati manualno ili postaviti skeniranje u određeno vrijeme svaki dan što osigurava pronalazak neočekivanih promjena u kratkom vremenu. [7]

3.1.12. Reguliranje definiranih dozvola za upravitelje softverskih paketa (156-177) (Verify and Correct File Permissions with RPM)

Na različitim distribucijama Linuxa koriste se različiti upravitelji softverskih paketa. Jedni su od najpoznatijih RPM i APT[17]. Upravitelji softverskih paketa upravljaju ažuriranjem i instalacijom softvera na samom sustavu. RPM upravitelj softverskih paketa može provjeriti koje se dozvole pristupa nalaze na svakom od njegovih instaliranih softverskih paketa uključujući i mnoge softvere bitne za informacijsku sigurnost. Svaki od tih paketa ima strogo definirane postavke koje omogućuju što sigurniju upotrebu instalacijskih paketa i njihovo ažuriranje. Samim time od velike je sigurnosne važnosti da se tih postavki pridržavaju svi korisnici sustava pa je tako potrebno sve pakete koji nemaju dobro definirane dozvole pristupa promijeniti u stroga pravila pristupa koja su predložena od strane proizvođača.

[16] „AIDE“ - Advanced Intrusion Detection Environment
[17] „APT“ - Advanced Package Tool

3.1.13. Provjera instaliranih softvera s RPM upraviteljem softverskih paketa (178-186) (Verify File Hashes with RPM)

Svaki puta kada je neki paket instaliran ili ažuriran uz pomoć RPM upravitelja softverskih paketa, promjene se bilježe u RPM sustav za praćenje promjena. Jako je bitno da RPM zapisuje sve promjene na svim instaliranim paketima kako bi se moglo provjeravati ako dođe do neželjenih grešaka. Osim što se takva provjera radi kako ne bi došlo do grešaka u sustavu, isto je tako potrebno provjeravati mogućnost drugih korisnika da ne mogu mijenjati pakete za svoje potrebe. Zato je potrebno da se sustav često provjerava, a u slučaju pronalaska paketa koji nisu sigurni za sustav potrebno je ponovno instalirati paket s definiranim i sigurnim postavkama.

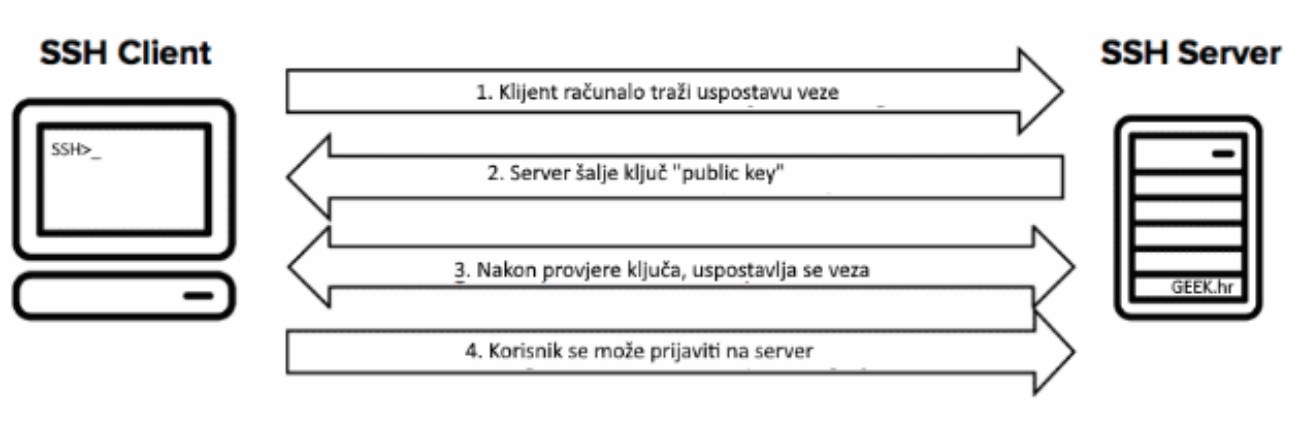
3.1.14. Konfiguracija kriptografske politike za korištenje SSH protokola (187-194) (Configure SSH to use System Crypto Policy)

SSH[18] je najpoznatiji protokol koji se koristi za povezivanje klijenta na server. Protokol koristi kriptiranu zaštitu za prijenos podataka te mogućnost udaljenog povezivanja. Način povezivanja s klijenta na udaljeni server prikazan je slikom (3.3.). SSH protokol toliko je poznat zato što ima jako dobre rezultate sigurnosti za razliku od prije korištenih (rcp, FTP, telnet…), ali to ne znači da ne može biti još sigurniji i da ne mora pratiti određena pravila pripadajuće politike sustava.

Kriptografska[19] politika omogućava centraliziranu kontrolu nad kriptografskim algoritmima koji se koriste na brojnim paketima i alatima. SSH je jedan od njih koji se već nalazi u određenoj kriptografskoj politici sustava, ali konfiguracija SSH protokola nekad zna biti postavljena da ignorira samu kriptografsku politiku što je loše za sam sustav. Zbog toga je potrebno namjestiti konfiguraciju SSH da ukloni sve moguće postavke zbog kojih se ignoriraju sigurnosne politike sustava kako bi sam sustav bio što sigurniji.

[18] „SSH“ - Secure Shell protocol

[19] Kriptografija je znanstvena disciplina koja se bavi proučavanjem metoda za slanje poruka u takvom obliku da ih samo onaj kome su namijenjene može pročitati



Slika 3.3. SSH povezivanje[20]

3.1.15. Konfiguracija kriptografske politike na sustavu (195-200) (Configure System Cryptography Policy)

Iako kriptografske politike imaju predefinirana pravila na sustavu, ona se mogu mijenjati ovisno o potrebi samog sustava. Zbog toga je potrebno pratiti sam rad sustava i moguće prijetnje na njemu kako bi se moglo ustanoviti koja politika najviše odgovara za određeni sustav. Jedini korisnik sustava koji može mijenjati predefinirana pravila kriptografske politike sam je administrator. Time dolazi do još veće sigurnosti i zaštićenosti sustava.

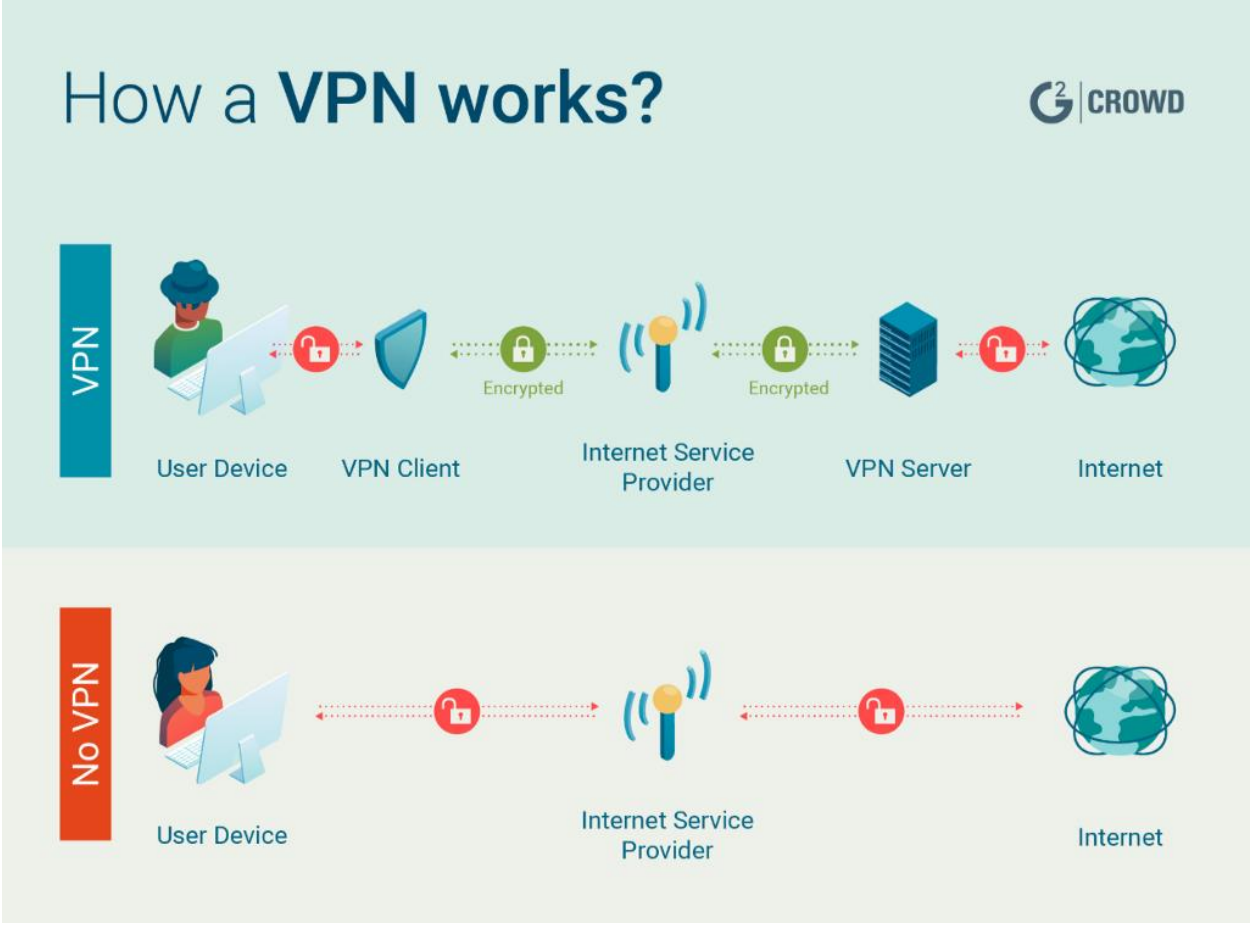
3.1.16. Konfiguracija kriptografske politike za korištenje Libreswan-a (201-211) (Configure Libreswan to use System Crypto Policy)

Korištenje Interneta bez neke zaštite može uvelike ugroziti sam sustav i njegovu konfiguraciju. Iako danas možemo vidjeti da gotovo svako veće mjesto ima besplatni Internet, to ne znači da je korištenje tih javnih Wi-Fi mreža sigurno. Uglavnom kad se radi o takvim mrežama, sigurnosti većinom ni nema, te ta veza nije kriptirana. Tako napadač može prisluškivati ostale korisnike na toj mreži i skupljati podatke koji su njemu potrebni za napad (lozinka, e-mail, slika...). Većinom takvi napadi nisu ni vidljivi samom korisniku što daje lažnu sigurnost samome korisniku, a to je napadaču i glavni cilj prilikom prikupljanja važnih informacija. Zbog takvih sigurnosnih nedostataka, najbolje je takve mreže ne koristiti ili pak koristiti uz neki oblik sigurnije veze poput virtualne privatne mreže (VPN[21]).

[20] Preuzeto s https://geek.hr/pojmovnik/sto-je-ssh/
[21] „VPN“ - Virtual Private Network

VPN je mrežna tehnologija kojoj je cilj stvoriti sigurnu vezu preko javne ili privatne mreže. Jedna je od temeljnih prednosti VPN-a to što šifrira vezu kojom se korisnik spaja na Internet pa ako napadač prisluškuje samu vezu, puno teže može doći do željenih informacija. Osim sigurnosti velika je prednost što se iz svog doma možemo spojiti na drugu mrežu koja nam je u tom trenutku potrebna pa ako imamo potrebu spojiti se na mrežu na poslu možemo preko VPN-a.

Libreswan je besplatna softverska implementacija najčešće podržanog i standardiziranog VPN-a predefinirana na Red Hat linux distribucijama. Kako vidimo, sama sigurnost veze je ključna prilikom spajanja na javnu ili privatnu mrežu. Iz sigurnosnih razloga također je potrebno da se i Libreswan pridržava kriptografske politike sustava. Libreswan koristi svoj način konfiguracije pa tako može zanemariti politiku sustava zbog čega je potrebno provjeriti i postaviti Libreswan da se pridržava svih potrebnih kriptografskih pravila sustava. [8]



Slika 3.4. Razlika korištenja mreže s i bez virtualne privatne mreže[22]

[22] Preuzeto s https://hackernoon.com/the-evolution-of-vpns-from-business-security-to-privacy-protection-b45a3bbf490f

3.1.17. Konfiguracija kriptografske politike za korištenje Kerberos-a (212-219) (Configure Kerberos to use System Crypto Policy)

Kerberos je jedan od najpoznatijih mrežnih protokola za autentifikaciju korisnika korišten na raznim operacijskim sustavima u arhitekturi klijent-poslužitelj pomoću enkripcije tajnog ključa. Osmišljen je tako da pruža jaku zaštitu koristeći kriptografiju. Tako se korisnik može povezati na server i preko nesigurne mrežne veze. Kerberos iz tog razloga mora biti pravilno postavljen kako ne bi došlo do ugrožavanja tajnosti i integriteta podataka. Kerberos također mora biti postavljen u okvire kriptografske politike sustava radi bolje zaštite. Postavljanje Kerberos-a u okvire kriptografske politike ne osigurava u potpunosti sustav, ali uvelike pomaže sigurnijem korištenju.[9]

3.1.18. Konfiguracija kriptografske politike za korištenje OpenSSL-a (220-241) (Configure OpenSSL library to use System Crypto Policy)

OpenSSL je skup alata koji se koristi za validaciju i kontrolu nad TLS[23] i SSL[24] protokolima. Većinom se koristi za dekodiranje i kodiranje određenog teksta, kriptografiju s javnim ključem, stvaranje i ukidanje certifikata te hashiranje[25]. Ovaj skup alata većinom je predefiniran na raznim Linux distribucijama. Pošto se radi o alatima koji su prisutni na većini servera na internetu, velika količina napada usmjerena je upravo prema njemu pa se često može pročitati da je OpenSSL pod nekom vrstom napada. Neovisno je li napad bio uspješan ili ne, sama količina napada govori o tome kako je od velike važnosti da se korištenje OpenSSL-a postavi pod kriptografsku politiku sustava.[10]

[23] „TLS“ - Transport Layer Security
[24] „SSL“ - Secure Sockets Layer
[25] Hashiranje - postupak koji omogućuje generiranje jedne ili više vrijednosti iz niza teksta pomoću matematičke formule. Rezultat je kriptirana vrijednost koja podatke čini sigurnima.

3.1.19. Prikazivanje zadnjeg pokušaja pristupa sustavu (242-251) (Ensure PAM Displays Last Logon/Access Notification)

PAM[26] je sustav koji omogućava priključivanje različitih metoda provjere autentičnosti na Linux sustav te ih zahtijeva od korisnika. Većina distribucija Red Hat Linuxa-a već predefinirano prilikom pokretanja imaju takav sustav, samo ga je potrebno postaviti na željene postavke. Prednost je takvog sustava da sam administrator može smanjiti ovlasti drugih korisnika i tako umanjiti neželjene i nesigurne radnje na sustavu. Jedna je od takvih mjera i mogućnost da se određenim korisnicima dozvoli da se logiraju samo u određeno vrijeme s određenog mjesta te da se prilikom svakog uspješnog pristupa u sustav prikaže vrijeme prethodnog pristupa. Tako možemo biti sigurni kada smo mi zadnji puta pristupili sustavu. U slučaju da se vrijeme ne poklapa s našom zadnjom prijavom znači da je netko neovlašteno s našeg računa pristupio sustavu te da dolazi do nesigurnosti sustava. Na ovakav način možemo i utvrditi tko se nije pridržavao zaštitnih mjera u sustavu te tako oštetio sam sustav.[11]

3.1.20. Zabrana pristupa sustavu bez lozinke (252-258) (Prevent Login to Accounts With Empty Password)

U današnje vrijeme gdje se koriste razne vrste sigurnosti sustava i dalje najkorišteniji način sigurnosti korištenje je lozinke. Ako korisnik ima račun koji je konfiguriran za provjeru autentičnosti zaporke, ali nema dodijeljenu lozinku, može doći do zloupotrebe samog sustava. Svaki drugi korisnik moći će se prijaviti na taj račun bez potrebnih zaštita koje su nužne radi zaštite sustava i pokretati sve naredbe s privilegijama tog računa. Zbog toga svi računi koji nemaju lozinku za pristup sustavu moraju biti obrisani ili im mora biti dodijeljena lozinka. Tako možemo biti sigurni da nitko neovlašten bez lozinke ne može pristupiti sustavu.

[26] „PAM“ - Pluggable Authentication Modules

3.1.21. Brisanje svih file-ova s nezaštićenim lozinkama (259-266) (Verify No netrc Files Exist)

Korištenje lozinki nužno je za prijavljivanje korisnika, ali i za pristup podataka s udaljenih servera. Prilikom svake prijave dolazi do korištenja korisničkog imena i lozinke koji mogu biti spremljeni u jednu od datoteka. Većina datoteka koje koriste takav način upisivanja podataka kriptiraju te podatke kako ne bi došlo do neovlaštenog korištenja, ali postoje i iznimke. Jedne od datoteka koje sadrže podatke o prijavi koji se koriste za automatsku prijavu na FTP[27] poslužitelju nazivaju se „.netrc“. Te datoteke mogu u sebi sadržavati nekriptirane lozinke što ih čini dostupnima za neovlaštene korisnike da pristupaju tim poslužiteljima. Sve takve datoteke treba pronaći na sustavu te ih obrisati.

3.1.22. Provjera svih spremljenih lozinki u sustavu (267-279) (Verify All Account Password Hashes are Shadowed)

Većina sustava ima određene datoteke koje sadrže popis svih korisnika, njihovih kriptiranih lozinki i određenih drugih karakteristika koje opisuju samog korisnika i razlikuju ga od drugih korisnika. Takve datoteke korisne su kada želimo provjeriti što sve pripada jednom korisniku sustava i koje su mu ovlasti postavljene. Iako sam administrator sustava može mijenjati određene karakteristike i mogućnosti svakog korisnika, lozinku nije moguće vidjeti. U slučaju da bilo koji korisnik, pa čak i administrator sustava, ima pristup lozinki bez da je kriptirana, dolazi do nesigurnosti sustava. U slučaju da se u datoteci uz sve karakteristike korisnika nalazi i lozinka u čitljivom obliku, mora doći do ponovnog postavljanja lozinke ili brisanja samog korisnika. Iako to daje jednu dozu sigurnosti, najbolji je način sigurnosti da ne postoje ni kriptirani oblici lozinki već samo oznaka „x“ uz ime koja ne može ni na koji način ugroziti čitljivost lozinke korisnika.

[27] „FTP“ - File Transfer Protocol mrežni je protokol koji služi za premještanje datoteka s jednog sustava na drugi (klijent – poslužitelj).

3.1.23. Provjera pripadnosti korisnika u valjanu grupu (280-292) (All GIDs referenced in /etc/passwd must be defined in /etc/group)

Popis svih korisnika na Linux-u nalazi se u datoteci /etc/passwd, dok se popis svih grupa i članova grupa nalazi u datoteci /etc/group/. Korisnici mogu biti definirani i u drugim datotekama također, ali u navedenim se datotekama nalaze u većini Linux sustava. Ime korisnika odnosno grupa samom Linux-u nisu od tolike važnosti. Sustavu su najvažniji korisnikov UID[28] i grupni GID[29]. Te dvije oznake u potpunosti prikazuju samom sustavu koje privilegije ima svaki korisnik svoje oznake i pripadajuće grupne oznake. Radi toga moramo biti u potpunosti sigurni da svaki korisnik ima ispravne oznake. U slučaju da neki od korisnika dobije oznaku grupe koja ne postoji na sustavu, grupa će se automatski stvoriti. Samim će time dobiti i privilegije grupe koje mogu premašiti potrebe tog korisnika, a to dovodi do velike nesigurnosti sustava. Zato je potrebno da svaku grupnu oznaku stvori grupa od strane administratora u kojoj će biti pravilno definirana pravila za svakog korisnika koji se nalazi u grupama.

3.1.24. Provjera jedinstvenosti korisničkih imena u sustavu (293-305) (Ensure All Accounts on the System Have Unique Names)

Prilikom stvaranja nekih korisnika može doći do poklapanja imena s drugim korisnikom zbog toga što korisnici mogu imati neke slične osobine ili karakteristike poput istog imena, sličnih poslova koje trebaju obavljati i raznih drugih razloga. Svaki bi od korisničkih računa trebao imati jedinstveno ime i lozinku. Poklapanje korisničkih imena i sustavu može poremetiti stabilnost i sigurnost tako da bi jedan od korisnika mogao dobiti ovlasti koje mu ne bi trebale biti dodijeljene i tako ugroziti sustav. Još jedan od loših primjera poklapanja imena je i lažno predstavljanje korisnika koje može naštetiti drugom korisniku s istim imenom. Tako ne možemo znati koji je korisnik napravio samu štetu ili još gore možemo optužiti drugog korisnika koji nije napravio ništa loše. Kako bi sustav mogao imati sva jedinstvena imena na sustavu potrebno je, ako se pronađe bilo koji od podijeljenih korisničkih imena, obrisati jedan od tih korisničkih računa ili promijeniti ime te uz posebnu pozornost dodatno provjeriti privilegije tih korisnika.

[28] „UID“ - User IDentificator
[29] „GID“ - Group IDentificator

3.1.25. Postavljanje posebnih privilegija samo za „root“ korisnika (306-318) (Verify Only Root Has UID 0)

Na Linux sustavu korisnik koji ima najviše ovlasti naziva se „root“. Kako sam sustav većinom gleda UID, a ne korisničko ime, tako je „root“ korisniku dodijeljen UID = 0. To znači da korisnik bilo kojeg imena koji ima UID = 0 ima iste ovlasti sustava kao i „root“. Tu dolazi do velikih sigurnosnih problema koji omogućuju drugom korisniku da mijenja sve u sustavu uključujući i promjene privilegija „root“ korisnika. Zato je potrebno često provjeravati sustav na ovakvu vrstu prijetnje te svakog korisnika koji ima UID = 0 obrisati ili mu postaviti novi UID.

3.1.26. Zabrana pristupa „root“ korisnika na povezane sustave (319-331) (Restrict Serial Port Root Login)

Kako je do sada objašnjeno, privilegije „root“ korisnika jako su velike i mogu ugroziti sustav na razne načine. Prema tome, poželjno je da se na „root“ korisnika spaja što je moguće manje te samo kada je to potrebno. Prilikom korištenja takvih privilegija najbolje je rješenje prestanak korištenja „root“ korisnika čim to više nije potrebno za daljnje akcije samog korisnika sustava. Jedna je od najboljih rješenja sigurnog korištenja „root“ korisnika i zabrana pristupa „root“ korisnika svim mogućim ostalim uređajima koji su spojeni na sam sustav. Tako se sustav štiti od prevelikih ovlasti korisnika koje mu nisu namijenjene prethodno.

3.1.27. Zabrana izravnih pristupa „root“ korisnika (319-331) (Direct root Logins Not Allowed)

Prema zadanim postavkama Red Hat Linux distribucija /etc/securetty je datoteka koja opisuje na koje se sustave „root“ korisnik može spojiti sa svim svojim privilegijama. U toj je datoteci većinom definirano da se „root“ korisnik može spojiti izravno samo na sustave koji su fizički spojeni na naš sustav. U slučaju da ta datoteka ne postoji, s „root“ privilegijama možemo se spojiti preko bilo koje komunikacije na neki od drugih sustava. Pa tako u slučaju korištenja „root“ korisnika preko nesigurnog protokola možemo otkriti svoju lozinku i ugroziti svoj sustav. Kako smo u prethodnom pravilu takav način pristupa ostalim spojenim sustavima označili kao nesigurnim, potrebno je u navedenoj datoteci obrisati sav sadržaj te ju održavati praznom što bi značilo da se s „root“ privilegijama ne možemo spojiti ni na jedan drugi sustav. Tek u slučaju neke nužnosti potrebno je mijenjati prava kako bi mogli pristupiti samo nužnim udaljenim sustavima.

3.1.28. Zabrana pristupa virtualnim konzolama s „root“ privilegijama (319-331) (Restrict Virtual Console Root Logins)

Linux je višekorisnički sustav što znači da ga može koristiti više korisnika. No, što ako dođe do potrebe da više korisnika koristi sustav istodobno? Zbog toga došlo je do potrebe za korištenjem virtualnih konzola. Virtualne konzole slične su konzolama koje se koriste izravno na samom sustavu, a jedna je od glavnih razlika što se sve u virtualnoj konzoli radi samo uz pomoću tipkovnice. Iako korištenje virtualnih konzola uvelike pomaže pri raspoređivanju određenih poslova u isto vrijeme, korištenje „root“ korisnika u njima treba biti zabranjeno. Samim time štitimo sustav od mogućih nesigurnih radnji od strane zlonamjernih korisnika.

3.1.29. Postavljanje upozorenja prije isteka lozinke (332-346) (Set Password Warning Age)

Lozinke su primarni način potvrđivanja korisnika na Linux distribucijama. To je jedan od razloga zašto je njihova sigurnost toliko bitna za sigurnost korisnika te cjelokupnog sustava. Iako je sama važnost čuvanja lozinke na sigurnom mjestu jedna od najboljih zaštita koje korisnik može napraviti, postoje razni drugi načini koji mogu ugroziti lozinku i samog korisnika. Zbog toga je potrebno da se postave posebne postavke sustava koje će se morati poštivati kako bi korisnik mogao upravljati svojim lozinkama. Tako dajemo dodatnu sigurnost sustavu od mogućih nesigurnih napada na sustav. Neke su mjere postavljene tako da se korisnik može lakše prilagoditi upravljanju lozinki. Jedna je od takvih mjera i postavljanje upozorenja prije isteka same lozinke. U slučaju da korisnik ima lozinku koja zbog sigurnosti mora biti promijenjena u određeno vrijeme, korisnik će dobiti poruku koliko dana može proći najviše prije nego korisnik sam mora promijeniti lozinku. Na takav se način daje pravo korisniku da si može prilagoditi obveze na sustavu i u sebi prikladnom vremenu promijeniti lozinku. U slučaju da na sustavu nije postavljen broj dana znači da upozorenja ni nema što bi moglo ugroziti politiku cijelog sustava i njegovu sigurnost. Najbolji je broj dana za održavanje ovog pravila 7.

3.1.30. Postavljanje minimalne duljine lozinke (347-363)
(Set Password Minimum Length in login.defs)

Prilikom promjene lozinke većina korisnika ima jednu lozinku koju su već prije koristili te samo na tu istu naprave malu promjenu. Takav način promjene lozinki stvara prijetnju da će lozinka biti otkrivena u neko dosljedno vrijeme zbog sličnosti sa starim lozinkama koje se više ne koriste. Svaka lozinka na sustavu mora imati određenu „jačinu“ koja nam pokazuje koliko je ta lozinka zapravo sigurna u odnosu na nekog drugog zlonamjernog korisnika koji bi mogao doći do nje. Jedan je od načina koji uvelike utječe na tu jačinu lozinke i njena duljina. Što je lozinka dulja to je manja vjerojatnost da će zlonamjerni napadač pogoditi lozinku. Najbolja je praksa da se lozinka nigdje ne zapisuje nego da je korisnik pamti kako netko ne bi mogao doći do tih zapisa. Kako bi korisnik mogao zapamtiti lozinku, te da bude sigurnosno ispravna politika sustava treba postaviti da je minimalna duljina lozinke 12.

```
root@wks01:~# chage -l vivek
Last password change                                    : May 05, 2012
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

Slika 3.5. Prikaz loše politike promjene lozinke na sustavu[30]

3.1.31. Postavljanje minimalne dobi za lozinku (364-379)
(Set Password Minimum Age)

Svaka lozinka, bez obzira koliko složena, na kraju može biti probijena od strane zlonamjernog korisnika. Stoga, lozinke moraju imati svoju dob trajanja i moraju se povremeno mijenjati. Prilikom stvaranja korisničkog računa potrebno je postaviti minimalnu dob trajanja lozinke kako sam korisnik ne bi morao stalno mijenjati lozinku. Jedan je od razloga zašto se postavlja minimalna dob lozinke taj što korisnici često žele promijeniti svoju lozinku nazad na staru koju su zapamtili, a to stvara velike sigurnosne probleme sustava. Kako bismo bili sigurni da ne dođe do prečestog mijenjanja lozinke, potrebno je postaviti politiku sustava da je minimalna dob lozinke 7 dana.

[30] Preuzeto s https://www.cyberciti.biz/faq/changing-password-of-specific-account-in-linux-commandline/

3.1.32. Postavljanje maksimalne dobi za lozinku (380-395) (Set Password Maximum Age)

Korisnici znaju često koristiti istu lozinku na više korisničkih računa. Ako sustav ne ograniči vijek trajanja lozinki i dopusti da se jedna lozinka koristi duže vrijeme, može dovesti sustav u opasnost. Tako zlonamjerni korisnik, ako dođe do lozinke, može koristiti sustav duže vrijeme bez ikakvih smetnji. Postavljanje maksimalne dobi lozinke osigurava da korisnik mora povremeno mijenjati svoje lozinke i tako osigura da taj korisnički račun mogu koristiti samo oni kojima je to i prethodno namijenjeno. Zato je potrebno postaviti politiku sustava da se lozinke moraju promijeniti svakih 90 dana.

3.1.33. Postavljanje odgovarajućih pravila za datoteke unutar „root“ putanja (396-403) (Ensure that Root's Path Does Not Include World or Group-Writable Directories)

Sve datoteke koje se nalaze unutar „root“ putanje mogu biti izvršne pa tako i rukovanje takvim datotekama mora biti s velikim oprezom. Putanja takvih datoteka postavljena je da se one mogu izvršavati od strane bilo kojeg korisnika, na bilo kojoj putanji u sustavu.. Neke od najčešćih korištenih naredbi u Linux distribucijama tako su postavljene. Zato promjenu takvih datoteka i stvaranje novih koje bi cijeli sustav mogao koristiti treba imati samo „root“ korisnik. Ako bilo koji drugi korisnik osim „root“ korisnika ima te privilegije, trebaju mu se ukloniti.

3.1.34. Omogućavanje usluge revizije za praćenje radnji na sustavu (404-427) (Enable auditd Service)

Najveću nesigurnost većinom za sam sustav stvaraju njegovi korisnici. Razlog tome je što ti korisnici već imaju pristup sustavu i tako je veliki dio posla već napravljen. Korisnici sustava mogu i nenamjerno napraviti veliku štetu sustavu. Stoga, svaki sustav mora imati neki način zapisa ili revizije svih radnji na sustavu kako bismo mogli biti sigurni što je dovelo do sigurnosnih pogrešaka na sustavu. Takva se usluga većinom već nalazi predefinirana na Linux distribucijama, stoga je potrebno urediti željene postavke zapisa i omogućiti da se usluga nesmetano koristi tijekom cijelog rada sustava. Bez takvog utvrđivanja događaja na sustavu bilo bi teško utvrditi što je zapravo uzrokovalo da dođe do pada sustava ili napada na njemu. Također, ako je usluga tako postavljena, možemo pojedinačno zapisivati postupke svakog korisnika te tako svaki korisnik

mora odgovarati za svoje postupke. U slučaju da usluga nije pokrenuta, potrebno ju je postaviti na željene postavke i omogućiti da bude pokrenuta odmah nakon podizanja sustava.[12]

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 success=no
exit=-13 a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1 ppid=2686 pid=3538 auid=500
uid=500 gid=500 euid=500 suid=500 fsuid=500 egid=500 sgid=500 fsgid=500 tty=pts0 ses=1
comm="cat" exe="/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
key="sshd_config"
type=CWD msg=audit(1364481363.243:24287):  cwd="/home/shadowman"
type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd_config"
inode=409248 dev=fd:00 mode=0100600 ouid=0 ogid=0 rdev=00:00
obj=system_u:object_r:etc_t:s0
```

Slika 3.6. Prikaz revizije radnji na sustavu[31]

3.1.35. Revizija procesa odmah nakon pokretanja sustava (428-435) (Enable Auditing for Processes Which Start Prior to the Audit Daemon)

Postavke revizije procesa mogu se mijenjati ovisno o samom sustavu korištenja. Jedno je od najboljih rješenja revizije sustava da se prati cijeli sustav i radnje na njemu za sve procese koje mogu koristiti takav način rada. Usluga praćenja sustava bilježi reviziju samo onih procesa koji su pokrenuti nakon same usluge. Tako postoji vrijeme korištenja sustava za koje se ne bilježi revizija što stvara mogućnost zloupotrebe sustava. Zbog toga je potrebno da se postavke usluge zapišu i u samu jezgru sustava kako bi bila pokrenuta odmah nakon samog pokretanja.

3.1.36. Veličina revizije podataka za sustav (436-447) (Configure auditd Number of Logs Retained)

Ukupna pohrana revizije procesa mora biti dovoljno velika da zadrži potrebne podatke određeno razdoblje. Pošto se nakon određenog vremena revizije brišu kako bi se mogle zapisati druge, od velike je važnosti da nam se ne obrišu revizije koje su nam i dalje potrebne. Zato je potrebno odrediti koliko će zadnjih revizija ostati zapisano na sustavu kada dođe do ažuriranja zapisa revizija. Ovisno o svakom sustavu taj broj nije isti, ali je preporučljivo da bude minimalno 5 zadnjih revizija procesa.

[31] Preuzeto s https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/security_guide/sec-understanding_audit_log_files

3.1.37. Upozorenja nastala revizijom sustava (448-459)
(Configure auditd space_left Action on Low Disk Space)

Neki sustavi koji su od velike važnosti koriste reviziju za svaku akciju na sustavu. Takav način zapisivanja procesa u sustavu daje veliku sigurnost, ali dovodi do nekih drugih problema. Jedan je od tih problema i količina podatkovnog prostora koji iz dana u dan postaje prevelika za održavanje i pohranu. U slučaju da dođe do toga, administrator sustava mora biti obaviješten kako bi mogao obavljati akcije koje mogu spriječiti zagušenost sustava. Zato je potrebno postavke revizije podesiti da se pošalje e-mail administratoru sustava o problemu koji je nastao tijekom revizije. Iako to ne rješava sam problem, uvelike osigurava da se problem primijeti na vrijeme.

3.1.38. Ažuriranje revizije procesa (460-472)
(Configure auditd max_log_file_action Upon Reaching Maximum Log Size)

Nakon nekog vremena praćenja određenih procesa na sustavu i njihovog zapisa stare nam zabilješke nisu više potrebne. U slučaju da administrator sustava odredi da stare revizije sustava nisu potrebne, on ne mora dobivati obavijest da ponestaje prostora na sustavu. Sustav će sam obrisati stare revizije i umjesto njih zapisati nove, koje su daleko važnije za sigurnost sustava. Takvo rješenje daje sigurnost da će sustav i dalje pratiti procese iako ponestane prostora za njihovo zapisivanje. Kako je i prije opisano postoji mogućnost da se odredi koliko će zadnjih revizija procesa ostati na sustavu.

3.1.39. Organizacija sustava prilikom manjka prostora za reviziju (473-484)
(Configure auditd admin_space_left Action on Low Disk Space)

Iako je već u prethodnim pravilima pojašnjeno koje su najbolje opcije prilikom nedostatka prostora za reviziju, neki ključni sustavi ne smiju imati obrisan nijedan zapis na sustavu. Pa tako brisanje prilikom dostignuća maksimalnog prostora nije opcija. Prilikom takvih sustava potrebno je da pri nedostatku memorije sustav prebaci u jednokorisnički sustav u kojemu će jedini korisnik moći biti administrator. Tako se smanjuje revizija i brzina kojom se ta revizija popunjava. Za to vrijeme taj korisnik mora osposobiti sustav da dalje može nesmetano zapisivati procese i provedene radnje na sustavu. Jedna je od najboljih opcija spremanje starijih revizija na neki drugi sustav kojem se može pristupiti s trenutnog sustava.

3.1.40. Postavljanje zapisa aktivnosti na jedno mjesto (485-498)
(Configure auditd to use audispd's syslog plugin)

Briga za svim zapisima koji se vode u reviziji aktivnosti s vremenom postane preveliko opterećenje za sustav i njegovo praćenje. Na nekim je sustavima bitno da su svi zapisi zabilježeni te da se čuvaju određeno vrijeme radi sigurnosti sustava. Radi toga je potrebno postaviti postavke zapisa da sve revizije sa zapisima budu posebno spremljene na lokalni server. Tamo se mogu dodatno pregledati u slučaju nekih nesigurnosti na sustavu. Tako sve zapise možemo imati na jednom mjestu bez brige za gubitkom nekih od zapisa.

3.1.41. Određivanje maksimalne veličine revizije zapisa (499-510)
(Configure auditd Max Log File Size)

Postoje razne mogućnosti na koji će način revizija biti formirana. Jedna je od tih i veličina datoteke u kojoj se nalazi revizija. Veličina je određena u megabajtima. Ukupna veličina datoteka mora biti dovoljno velika da zadrži potrebne podatke tijekom određenog razdoblja. Određivanje prevelike mogućnosti pohrane u datoteku donosi nesigurnost prilikom pokušavanja prijenosa podataka u slučaju neke nesigurnosti. Svaki sustav ima različit broj korisnika pa je tako i količina zapisa drugačija. Samim time potrebno je sam sustav ispitiati prije donošenja ovakve odluke.

3.1.42. Određivanje odgovarajuće osobe tijekom nesigurnosti sustava (511-522)
(Configure auditd mail_acct Action on Low Disk Space)

Prije spomenuta situacija u kojoj ponestaje mjesta na samom sustavu samo je jedna od nesigurnih mogućnosti za sustav. U današnje vrijeme raznih mogućnosti prilikom samog napada na sustav potrebno je odrediti odgovornu osobu koja će dobiti obavijest prilikom neželjenog ponašanja sustava. Većinom je ta osoba administrator sustava koji ima sve ovlasti kako bi mogao pokušati riješiti taj problem nesigurnosti. Zato je potrebno da administrator sustava postavi koje se sve situacije smatraju nesigurnim za sustav te što se smatra neuobičajenim korištenjem sustava. Tako će administrator biti odmah obaviješten preko e-maila kako je došlo do nesigurnosti sustava. U slučaju da se neki drugi korisnik treba obavijestiti, potrebno ga je postaviti kao kontakt osobu prilikom mijenjanja postavki revizije sustava.

3.1.43. Kontrola postupaka administratora (523-553)
(Ensure auditd Collects System Administrator Actions)

Svi postupci moraju biti zabilježeni na sustavu, pa tako i oni od strane administratora. Administrator ima najveće privilegije na sustavu, pa tako postoji i najveća šansa da iskoristi te privilegije za nesigurnost sustava, bilo to namjerno ili slučajno. Praćenje tih postupaka ne odnosi se samo na administratora, već i na sve ostale administratorske radnje koje mogu biti i od strane drugih korisnika ako im je to dopušteno. Ovakvom revizijom procesa možemo biti sigurni da ni korisnici s najviše privilegija ne mogu iskoristiti sustav za neku svoju korist bez da odgovaraju za to kasnije.

3.1.44. Provjera promjene korisničkih podataka (554-571)
(Record Events that Modify User/Group Information)

Postavke zapisa i vođenje revizije procesa na sustavu može koristiti razne načine prikupljanja podataka na sustavu. Jedan je od važnijih procesa koji treba biti zabilježen u reviziji i promjena korisničkih podataka na sustavu ili sam pokušaj promjene. Omogućavanje takvih postavki dodatno će dodatno pridonijeti sigurnosti sustava. Osim što će svaki takav proces biti zabilježen, administrator će dobiti obavijest za sve izmjene korisničkih ili grupnih postavki. Tako ostali korisnici mogu biti upozoreni za postupke koje ne bi trebali raditi bez znanja i odobrenja ovlaštene osobe.

3.1.45. Mogućnosti pregleda revizije procesa (572-580)
(System Audit Logs Must Be Owned By Root)

Svaki od zapisa i revizija na Fedora sustavu sprema se u /var/log/audit. Ta mogućnost pregledavanja i svih drugih datoteka koji se nalaze pod direktorijem stvara veliku odgovornost na sustavu. Stoga, jedina osoba koja može biti vlasnik i kao korisnik i kao grupa može biti „root“. Loše obavljanje revizijskih zapisa može napadačima otkriti podatke o sustavu i njegovoj konfiguraciji čime je ugrožena i njegova povjerljivost. U slučaju da je neki drugi korisnik vlasnik revizija, vlasništvo mora biti promijenjeno u „root“ korisnika.

3.1.46. Kontrola izvoza podataka sa sustava (581-592)

(Ensure auditd Collects Information on Exporting to Media (successful))

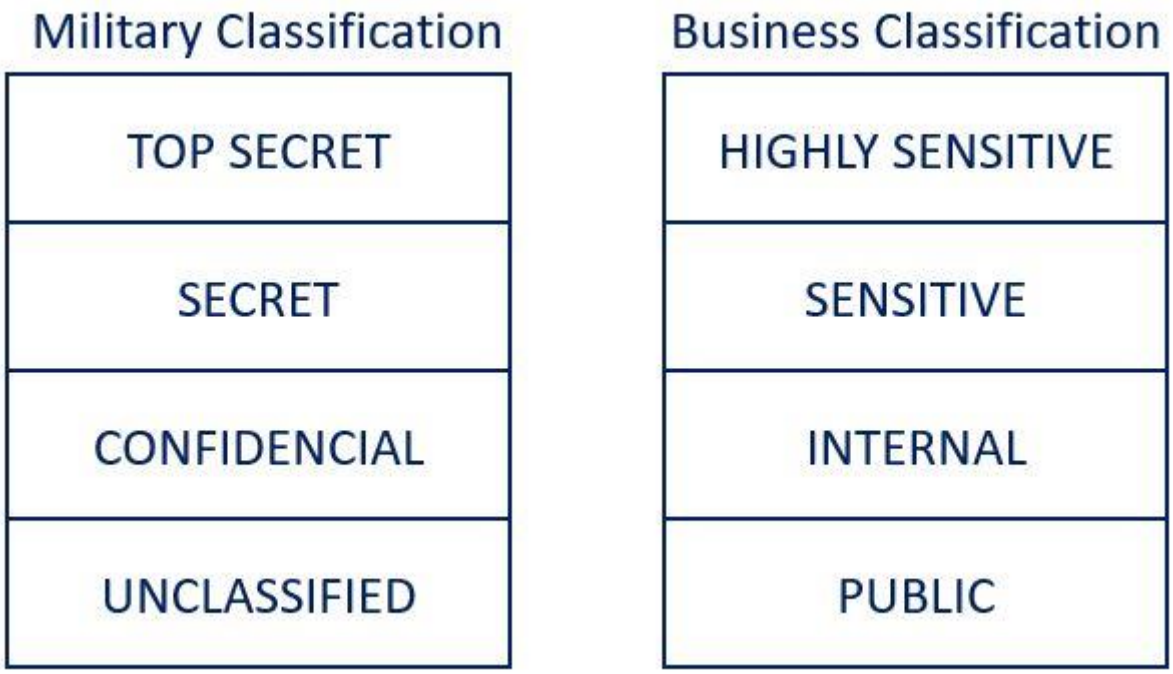
Neovlašteni izvoz podataka na vanjske medije mogao bi rezultirati curenjem informacija gdje bi se mogli izgubiti važni osjetljivi podaci. Trag revizije mora se stvoriti svaki put prilikom instalacije nekog novog datotečnog sustava i pokušaja prebacivanja na njega. Tako se mogu prepoznati neovlaštene radnje i pokušaji prebacivanja povjerljivih informacija na neki drugi sustav bilo to slučajno ili namjerno. U slučaju da su podaci već prebačeni, možemo biti sigurni koji su podaci prebačeni i koji je korisnik to napravio kako bismo lakše mogli ispitati situaciju i tako brže riješiti problem.

3.1.47. Provjera zapisa mandatornog modela pristupa (593-602)

(Record Events that Modify the System's Mandatory Access Controls)

U gotovo svakom sustavu dio sigurnosne politike čini i kontrola pristupa. Kontrola pristupa na sustavu služi za ograničavanje korisnika i procesa kako ne bi došlo do neželjenih radnji, a ujedno i ograničava štetu koju neoprezni korisnici ili napadači mogu prouzročiti. Pristupom nekom resursu korisnik ima određene akcije s kojima može raspolagati te se takve akcije moraju dovesti do samo nužnih za svakog korisnika. Mandatorni model pristupa (MAC)[32] kontrola je pristupa u kojoj sam sustav ograničava korištenje određenih resursa radi bolje sigurnosti. Svaki korisnik i određeni resurs posebno se klasificira kako bi se znalo koji korisnik ima pravo i točno koje akcije odraditi na nekom resursu. Jedino administrator ima prava na postavljanje sigurnosne politike za cijeli sustav dok korisnici nemaju nikakva prava za promjenom. Tako možemo biti sigurni da cijeli sustav mora poštivati određena pravila sigurnosne politike na sustavu. Takav model pristupa većinom se koristi na sustavima koji imaju određene podatke od velike važnosti te samim time ti podatci nikako ne smiju doći neovlaštenom korisniku sustava. Samo korištenje nekog resursa moguće je od strane korisnika samo ako je taj korisnik u klasifikaciji koja ima jednaku ili višu važnost od samog resursa, u slučaju da nema dovoljnu klasifikaciju neće moći pristupiti tom resursu. Radi same osjetljivosti podataka, svaka akcija ili pokušaj akcije promjene kontrole pristupa mora biti zabilježena u reviziju kako bi se detaljnije mogla pregledati.[13]

[32] „MAC" - Mandatory Access Control



Slika 3.7. Primjer klasifikacije u mandatornom modelu pristupa[33]

3.1.48. Provjera individualnih izmjena revizije procesa (603-616) (Record Attempts to Alter Process and Session Initiation Information)

Sustav revizije procesa već skuplja podatke od svih korisnika i administratora. To nam daje sigurnost da će svaki proces biti zapisan na sustavu u određenu datoteku zapisa. Većina napadača upoznata je s tim pravilima sustava te postoji mogućnost da prilikom napada ili poslije njega pokušaju doći do tih zapisa kako bi ih obrisali ili još gore promijenili da se ne bi primijetilo sumnjivo ponašanje na sustavu. Individualno mijenjanje revizije podataka mora biti također zapisano pošto tako možemo biti sigurni koji je izvorni zapis bio mijenjan.

3.1.49. Provjera promjena mrežnih postavki sustava (617-637) (Record Events that Modify the System's Network Environment)

Najveće opasnosti sustava u današnje vrijeme proizlaze iz nesigurnih mrežnih postavki. Korištenje interneta uvelike pomaže napadačima provjeravati podatke koji mogu pomoći pri neovlaštenom pristupu sustavu. Iako su korisnici ti koji najviše uzrokuju izloženost sustava, mrežni programi koji se koriste svakodnevno također mogu naštetiti sustavu. Korištenje bilo koje od aplikacija znači

[33] Preuzeto s https://thorteaches.com/cissp-certification-mac/

da se ista ta aplikacija može koristiti i u zlonamjerne svrhe. Stoga je bitno da svaka promjena u mrežnim postavkama može doći samo od ovlaštenih korisnika za mrežu na sustavu. Bilo kakve promjene mrežnih postavki moraju biti zapisane u reviziji.

3.1.50. Nemogućnost promjene revizije procesa (638-647)
(Make the auditd Configuration Immutable)

Razne postavke revizije sustava mogu pomoći optimizirati određeni sustav prema željenim specifikacijama. Jedna je od tih postavki i nemogućnost promjene revizije bez da se sustav mora ponovno pokrenuti. U nepromjenjivom načinu korištenja revizije neovlašteni korisnici ne mogu izvršavati promjene u revizorskom sustavu kako bi potencijalno sakrili zlonamjerne aktivnosti i zatim vratili pravila revizije. Korisnici će najvjerojatnije primijetiti ponovno pokretanje sustava i to bi moglo upozoriti administratore na pokušaj neovlaštene revizije. Povodom ponovnog pokretanja sustava administrator može naći druge mjere zaštite i dobiti dodatno vrijeme za rješavanje napada ako je to nužno.

3.1.51. Prikupljanje zapisa procesa za jezgru sustava (648-659)
(Ensure auditd Collects Information on Kernel Module Loading and Unloading)

Jezgra (engl. kernel) moduli dijelovi su koda koji dodaju dodatne funkcionalnosti osnovnoj jezgri sustava. Određeni moduli mogu se dodavati direktno na jezgru ili se mogu ukloniti od nje. Tako se povećava funkcionalnost jezgre bez da se sustav mora ponovno pokrenuti. Dodavanje i uklanjanje modula može se koristiti za promjenu ponašanja jezgre i potencijalno zlonamjerno ponašanje na sustavu. Zbog toga je potrebno da svaka promjena koja se odnosi na jezgru bude zapisana u reviziji sustava i tako svaka promjena može biti pravodobno provjerena.

3.1.52. Prikupljanje zapisa za sve obrisane datoteke (660-671)
(Ensure auditd Collects File Deletion Events by User)

Datoteka je logička cjelina koja služi za pohranjivanje nekih informacija. Te informacije mogu imati jako malo utjecaja na sigurnost sustava dok druge mogu biti od velikog značaja. Stoga, bilo koja promjena nad datotekama na sustavu može naštetiti njegovoj sigurnosti. Mijenjanje određenih datoteka treba biti dopušteno samo od strane ovlaštene osobe. Promjene mogu biti razne, od brisanja određene datoteke sve do malog uređivanja sadržaja koji može proći nezapaženo od strane

ovlaštenog korisnika. Svaka promjena nad datotekama neovisno od kojeg je značaja može doprinijeti nesigurnosti sustava i zbog toga takva promjena mora biti zapisana u reviziju sustava radi daljnje provjere opasnosti same promjene.

3.1.53. Prikupljanje zapisa od privilegiranih naredbi (672-686) (Ensure auditd Collects Information on the Use of Privileged Commands)

Iako postavljanje ograničenja korisnicima na određene akcije daje jednu vrstu sigurnosti, te iste akcije mogu biti iskorištene za zlonamjerne radnje. Prilikom postavljanja prava za nekog korisnika većinom se korisniku daju i mogućnosti korištenja i nekih privilegiranih naredbi. Privilegirane naredbe specifične su po tome što imaju mogućnost kao da su pokrenute s „root“ korisnika što znači da ima sve moguće privilegije za određenu promjenu na sustavu. Zloupotreba privilegiranih naredbi, nenamjerno ili namjerno od strane ovlaštenog ili neovlaštenog korisnika može imati značajan štetni utjecaj na sam sustav. Revizija zapisa privilegiranih naredbi jedan je od načina zaštite od zloupotrebe korištenja takvih naredbi. Mogućnost kasnije provjere revizije daje mogućnost ispravljanja nesigurnosti na sustavu.

3.1.54. Zabrana mijenjanja zapisa za prijavu i odjavu sa sustava (687-700) (Record Attempts to Alter Logon and Logout Events)

Revizija sustava prati sve prijave i odjave sa sustava. Tako se osigurava tko je sve bio na sustavu u određeno vrijeme. U slučaju neovlaštenog pristupa na sustavu taj zapis može pomoći pri pronalasku štete na sustavu. Određeni zlonamjerni korisnici mogu pokušati obrisati dokaze koji prikazuju prijavu i odjavu sa sustava kako ne bi bili otkriveni prilikom nekog napada na sustav. Zbog toga je potrebno da sve takve moguće promjene također budu zabilježene u posebnu datoteku koja će biti dostupna samo administratoru sustava. Tako možemo biti sigurni da dokazi nisu obrisani od strane napadača.

(3.1.55.– 3.1.67.) Prikupljanje podataka zapisa uz diskrecijski model kontrole pristupa (701-861)

Sljedeći skup pravila opisan je pod jednom kategorijom (Prikupljanje podataka zapisa uz diskrecijski model kontrole pristupa) zato što ima jako sličnu važnost za sigurnost sustava. Popis je napravljen od izvornog imena samog pravila te poslije pravila slijedi raspon redova iz priloga A koji rješavaju to pravilo.

Popis pravila:

3.1.55. (Record Events that Modify the System's Discretionary Access Controls - fsetxattr) (701-712)

3.1.56. (Record Events that Modify the System's Discretionary Access Controls - lremovexattr) (713-724)

3.1.57. (Record Events that Modify the System's Discretionary Access Controls - fremovexattr) (725-736)

3.1.58. (Record Events that Modify the System's Discretionary Access Controls - chmod) (737-749)

3.1.59. (Record Events that Modify the System's Discretionary Access Controls – lchown) (750-761)

3.1.60. (Record Events that Modify the System's Discretionary Access Controls - lsetxattr) (762-773)

3.1.61. (Record Events that Modify the System's Discretionary Access Controls - fchowna) (774-785)

3.1.62. (Record Events that Modify the System's Discretionary Access Controls - fchown) (786-798)

3.1.63. (Record Events that Modify the System's Discretionary Access Controls -removexattr) (799-811)

3.1.64. (Record Events that Modify the System's Discretionary Access Controls - chown) (812-824)

3.1.65. (Record Events that Modify the System's Discretionary Access Controls - fchmod) (825-836)

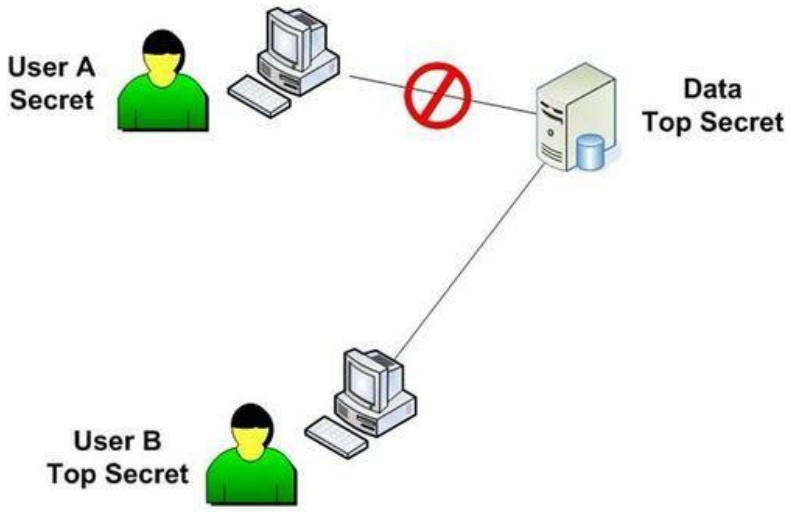
3.1.66. (Record Events that Modify the System's Discretionary Access Controls - setxattr) (837-849)

3.1.67. (Record Events that Modify the System's Discretionary Access Controls - fchmodat) (850-861)

Diskrecijski model kontrole pristupa (DAC)[34] model je pristupa u kojem se upravlja računalnim resursima sustava. U ovakvom modelu pristupa pojedini korisnici dobivaju jedinstvene ovlasti. Ovlasti određuje sam vlasnik nekog resursa pa tako jedan korisnik može imati prava čitanja i

[34] „DAC" - Discretionary access control

pisanja na resursu dok drugi može imati samo prava čitanja. Vlasnik nekog resursa većinom je ona osoba koje je taj resurs i stvorila, pa samim time može imati najviše ovlasti nad njim. Prava pristupa trebaju biti dodijeljena samo onim korisnicima kojima je to nužno i potrebno i samo određeno vrijeme za koje je to potrebno. U distribucijskom modelu pristupa ne ograničava se kopiranje resursa što znači ako korisnik ima ovlasti čitanja, tada korisnik ima pravo i kopiranja. Tako korisnik može stvoriti vlastiti resurs u kojemu će on sam biti vlasnik te će za svoj resurs moći dodjeljivati prava ostalim korisnicima. Zbog toga može biti narušen integritet i sigurnost podataka na sustavu. Stoga, sve takve akcije i naredbe koje su povezane s bilo kojom mogućnošću za narušavanjem sigurnosti diskrecijskog modela moraju biti zapisane u reviziju procesa na sustavu. Prikupljanjem takvih zapisa možemo ustanoviti koji od korisnika namjerno ili nenamjerno nanosi štetu samom sustavu.[13]



Slika 3.8. Skica diskrecijskog modela kontrole pristupa[35]

[35] Preuzeto s https://www.indiamart.com/proddetail/discretionary-access-control-10426347662.html

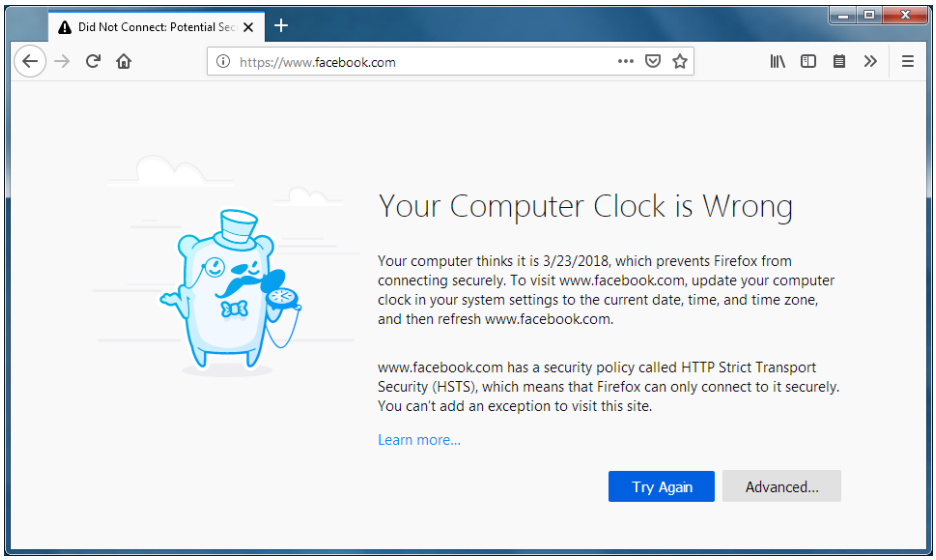
(3.1.68.-3.1.72.) Provjera zapisa revizije za promjenu vremena (862-917)

Sljedeći skup pravila opisan je pod jednom kategorijom (Provjera zapisa revizije za promjenu vremena) zato što ima jako sličnu važnost za sigurnost sustava. Popis je napravljen od izvornog imena samog pravila te poslije pravila slijedi raspon redova iz priloga A koji rješavaju to pravilo.

Popis pravila:

3.1.68. (Record Attempts to Alter the localtime File) (862-871)
3.1.69. (Record attempts to alter time through adjtimex) (872-883)
3.1.70. (Record attempts to alter time through settimeofday) (884-895)
3.1.71. (Record attempts to alter time through stime) (896-905)
3.1.72. (Record Attempts to Alter Time Through clock_settime) (906-917)

Svaki sustav koji ima akcije u trenutnom vremenu mora imati dobre podatke o vremenu. Samovoljne promjene vremena mogu se iskoristiti za nesigurne radnje na sustavu. Vrijeme na sustavu koristi se za puno više procesa nego samo za trenutno vrijeme prikazano za svakog korisnika. Na sustavu se nalaze brojni procesi i usluge koje uvelike ovise o točnom vremenu. Mijenjanjem vremena na sustavu svi zapisi za koje nam je potrebno vrijeme imat će grešku u vremenu te tako nećemo moći biti sigurni u ispravnost revizije podataka. Zbog toga je potrebno da sve promjene na sustavu koje utječu na vrijeme budu zapisane u reviziji podataka da tako administratori sustava mogu dobiti prave zapise u reviziji podataka.



Slika 3.9. Prikaz usluge koja ovisi o vremenu[36]

[36] Preuzeto s https://support.mozilla.org/en-US/kb/troubleshoot-time-errors-secure-websites

3.1.73. Prikupljanje zapisa o neuspješnim pokušajima pristupa (918-934)
(Ensure auditd Collects Unauthorized Access Attempts to Files (unsuccessful))

Pristup informacijama na sustavu mora biti reguliran prema postavljenoj politici zaštite podataka na sustavu. Prilikom postavljanja politike na sustav svi korisnici moraju biti upoznati sa svojom ovlastima i mogućim zabranama na sustavu. Mnogi korisnici na sustavu namjerno ili nenamjerno pokušavaju koristiti informacije i resurse koji im nisu namijenjeni. Svi takvi pokušaji koji ne mogu biti svrstani jesu namjerni ili nenamjerni i moraju biti zabilježeni u reviziji zapisa kako bi se kasnije moglo doći do razloga pokušaja dolaska to tih resursa.

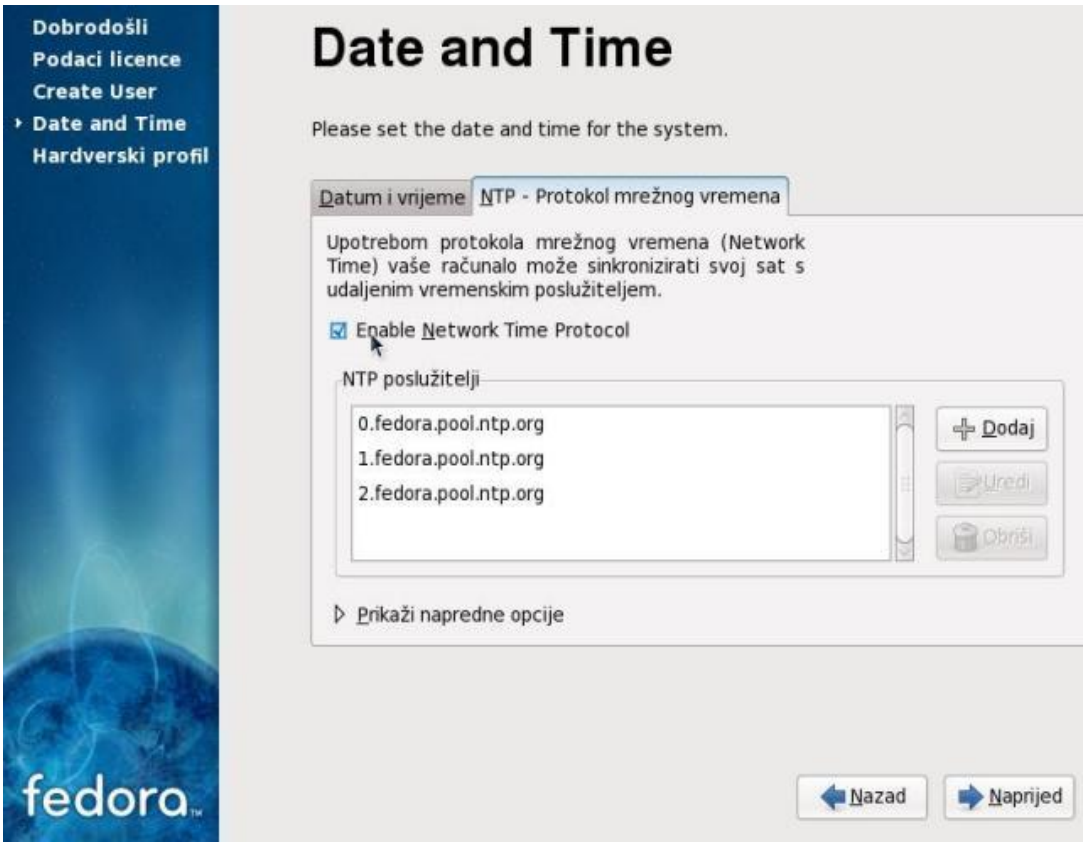
3.1.74. Postavljanje NTP protokola (935-960)
(Enable the NTP Daemon)

NTP[37] je najkorišteniji i najpoznatiji protokol za sinkronizaciju vremena na sustavu. Radi na principu server-klijent u kojemu je sustav spojen na više servera u isto vrijeme kako ne bi došlo do pogreške u vremenu zbog prevelikog prometa sustava. Jedan od osnovnih uvjeta za uspostavu NTP protokola na server internetska je veza, ali se može koristiti i lokalni server sustava. Takav oblik usluge jako je bitan za usluge provjere autentičnosti te usluge provjere revizije podataka prilikom provjere sigurnosti sustava. Kako bi vrijeme na sustavu bilo korektno postavljeno, sustav mora omogućiti stalno korištenje NTP usluge.[14]

3.1.75. Određivanje udaljenih servera za NTP (961-976)
(Specify a Remote NTP Server)

Svaki od servera ima svoju specifikaciju u odnosu na klijenta. U slučaju NTP servera preporučljivo je da ti serveri budu na što bližoj lokaciji u odnosu na klijenta kako ne bi došlo do grešaka u latenciji. Serveri rade u grupama kako ne bi došlo do poteškoća ako dođe do pada od jednih od servera. Prilikom postavljanja servera bitno je da su serveri dobro organizirani te na što bližoj lokaciji jedan od drugog. Ako serveri nisu korektno postavljeni, moraju se postaviti odgovarajući serveri za sigurnu sinkronizaciju vremena na sustavu.

[37] „NTP“ - Network Time Protocol



Slika 3.10. Prikaz grafičkog dodavanja NTP servera[38]

3.1.76. Postavljanje SSH intervala za neaktivnost (977-982)

(Set SSH Idle Timeout Interval)

SSH omogućava administratorima sustava da postave interval neaktivnosti korisnika. Nakon što taj interval prođe korisnik će se automatski odjaviti sa sustava. Prilikom spajanja na sustav postoji mogućnost da neki od korisnika ostane spojen na sustavu preko SSH duži period iako nije aktivan. Samim time ugrožava sustav i postoji mogućnost prilikom nepažnje korisnika da se ta konekcija iskoristi za ugrožavanje sustava. Zato je potrebno da se posebno za sustav odredi maksimalni interval koji će odrediti koliko korisnik može biti spojen na sustav bez da obavlja ikakve aktivnosti.

[38] Preuzeto s https://wiki.open.hr/wiki/Datoteka:Fedora10live-21.jpeg

3.1.77. Onemogućavanje spajanja na „root“ korisnika preko SSH (983-990) (Disable SSH Root Login)

Iako je SSH protokol šifriran, dodatnim slojem zaštite postiže se proširivanje politike sustava neprijavljivanja na sustav direktno kao „root“ korisnik. Tako možemo biti sigurni da napadač ne može doći do svih ovlasti i tako naštetiti sustavu. Dodatna je prednost što se tako svaki korisnik mora autentificirati i odgovarati za svoje postupke na sustavu. Djelovanje „root“ korisnika na sustavu mora biti ograničeno na što manje korištenje kako ne bi došlo do sigurnosnih problema na sustavu, dok prilikom spajanja preko SSH protokola na „root“ korisnika mora biti zabranjeno.

3.1.78. Postavljanje održavanje klijenta spojenim (991-999) (Set SSH Client Alive Max Count)

Prilikom postavljanja SSH intervala za neaktivnost ujedno se postavlja i održavanje klijenta spojenim na sustav. Do toga dolazi kako sam korisnik ne bi bio odmah odjavljen ako nenamjerno nije imao nikakvu aktivnost na sustavu. Sustav i administrator koji upravlja politikom sustava ne mogu znati koji su korisnici namjerno, a koji nenamjerno ostali duže vrijeme neaktivni na sustavu. Zbog toga održavanje klijenta mora biti ukinuto tako da svaki puta kada korisnik bude neaktivan za postavljeni interval vremena mora biti odmah odjavljen.

3.1.79. Onemogućavanje spajanja na sustav bez lozinke preko SSH protokola (1000-1006) (Disable SSH Access via Empty Passwords)

Korisnici koji se nalaze na udaljenim lokacijama od sustava znaju često koristiti sustav u raznim intervalima i tako imaju obvezu spajanja i pisanje lozinke svaki puta ispočetka. Zbog toga SSH ima mogućnost politike u kojoj lozinku nije potrebno upisivati prilikom spajanja na sustav. Takav način rada skraćuje samo vrijeme spajanja na sustav, ali radi i veliku nesigurnost u sustavu. Ako sustav ima takvu politiku, potrebno ju je odmah maknuti te postaviti politiku u kojoj je potrebno koristiti lozinku prilikom spajanja na sustav.

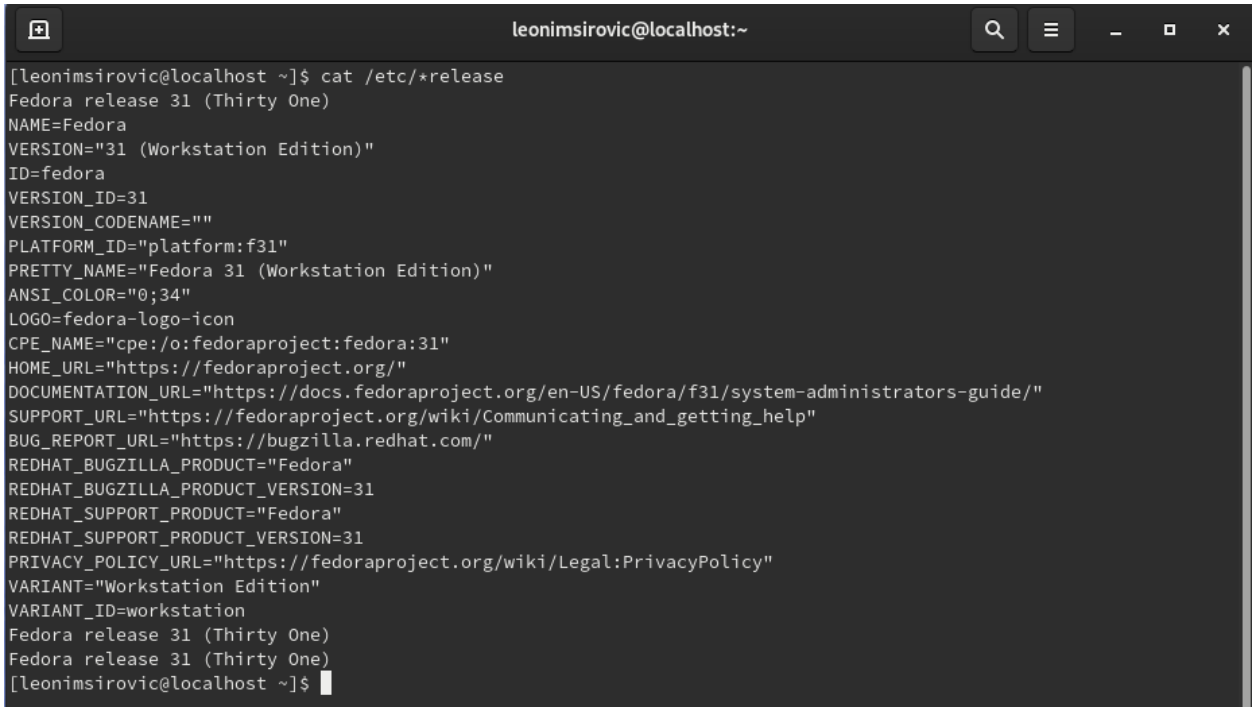
3.1.80. Konfiguracija BIND poslužitelja za korištenje kriptografske politike sustava (Configure BIND to use System Crypto Policy)

BIND[39] je jedan od najpopularnijih i najkorišteniji DNS[40] poslužitelj. BIND translatira domene u IP adrese sustava te se većinom koristi na Linux distribucijama. Učestalost korištenja interneta povećava važnost korištenja BIND poslužitelja. Zbog toga korištenje BIND poslužitelja mora biti uvršteno u kriptografsku politiku sustava. Tako ako dođe do neke nesigurnosti u sustavu, možemo biti sigurni da podaci nisu zapisani bez ikakve zaštite.

Zadnje je pravilo samo opisano bez njegovog rješenja u Prilogu A zato što OpenSCAP ne provjerava točnost rješavanja ovog problema. [15]

3.2. Prikaz rješenja

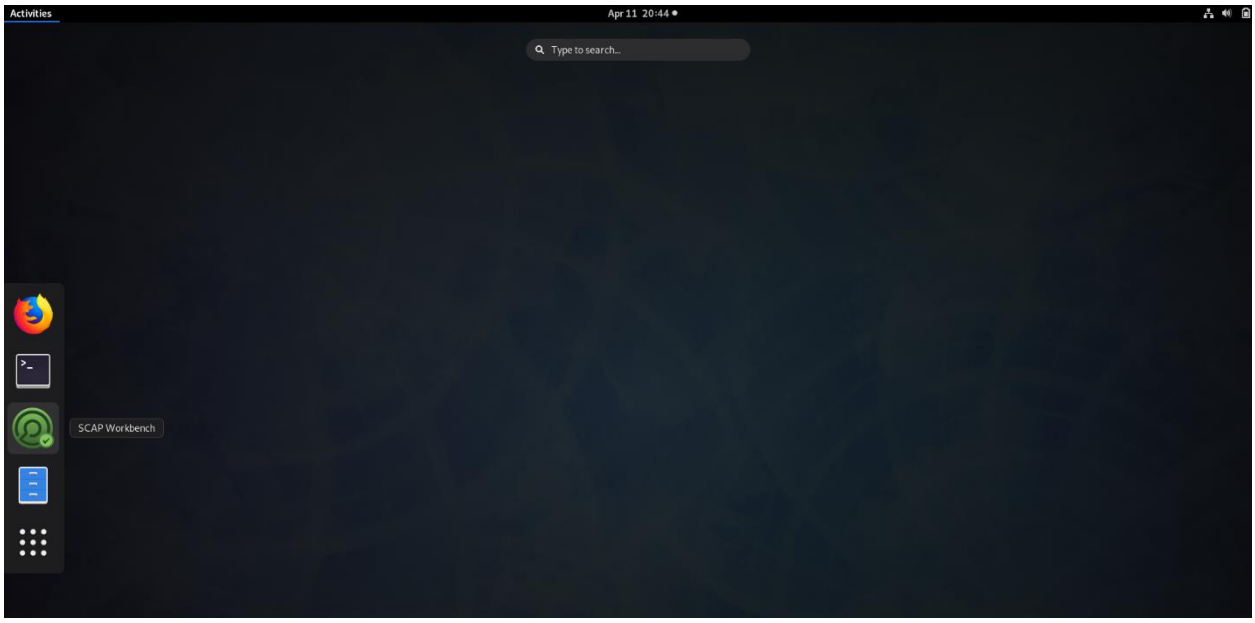
Sustav na kojemu se odvijala sigurnosna provjera te pokušavalo uz pomoć automatizacije riješiti sigurnosne propuste je Fedora 31. Verzija testiranog sustava prikazana je prema slici 3.11.



Slika 3.11. Prikaz verzije testiranog sustava

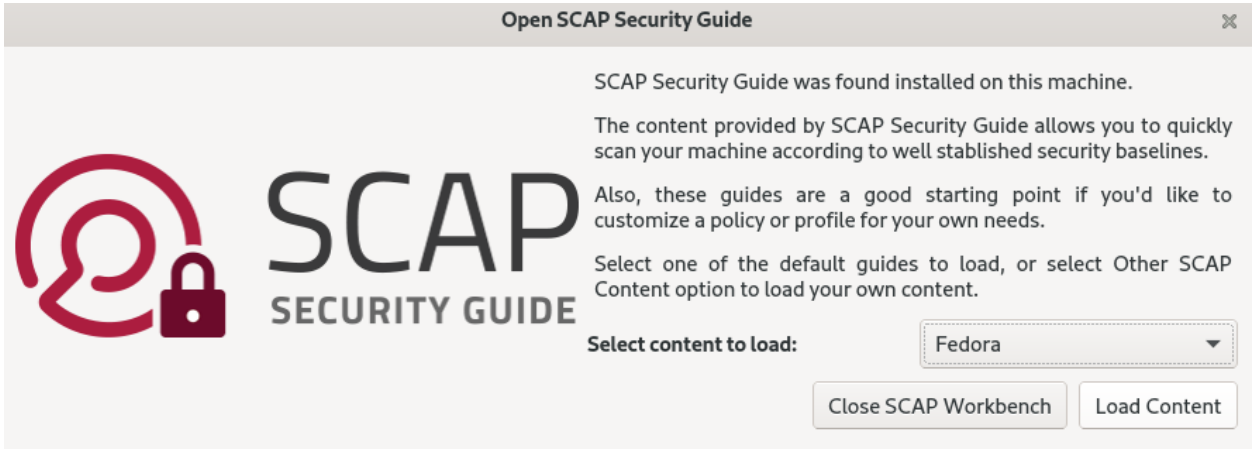
[39] „BIND“ - Berkeley Internet Name Domain
[40] „DNS“ - Domain Name System

Radi jednostavnijeg skeniranja sustava i korištenja OpenSCAP-a, prilikom skeniranja korišten je grafički alat „SCAP Workbench“. Korištenje „SCAP Workbench“ alata moguće je s Linux operativnog sistema (lokalno i daljinski) te s Windows operativnog sistema (daljinski). Prilikom skeniranja bit će objašnjen lokalni način korištenja spomenutog alata.



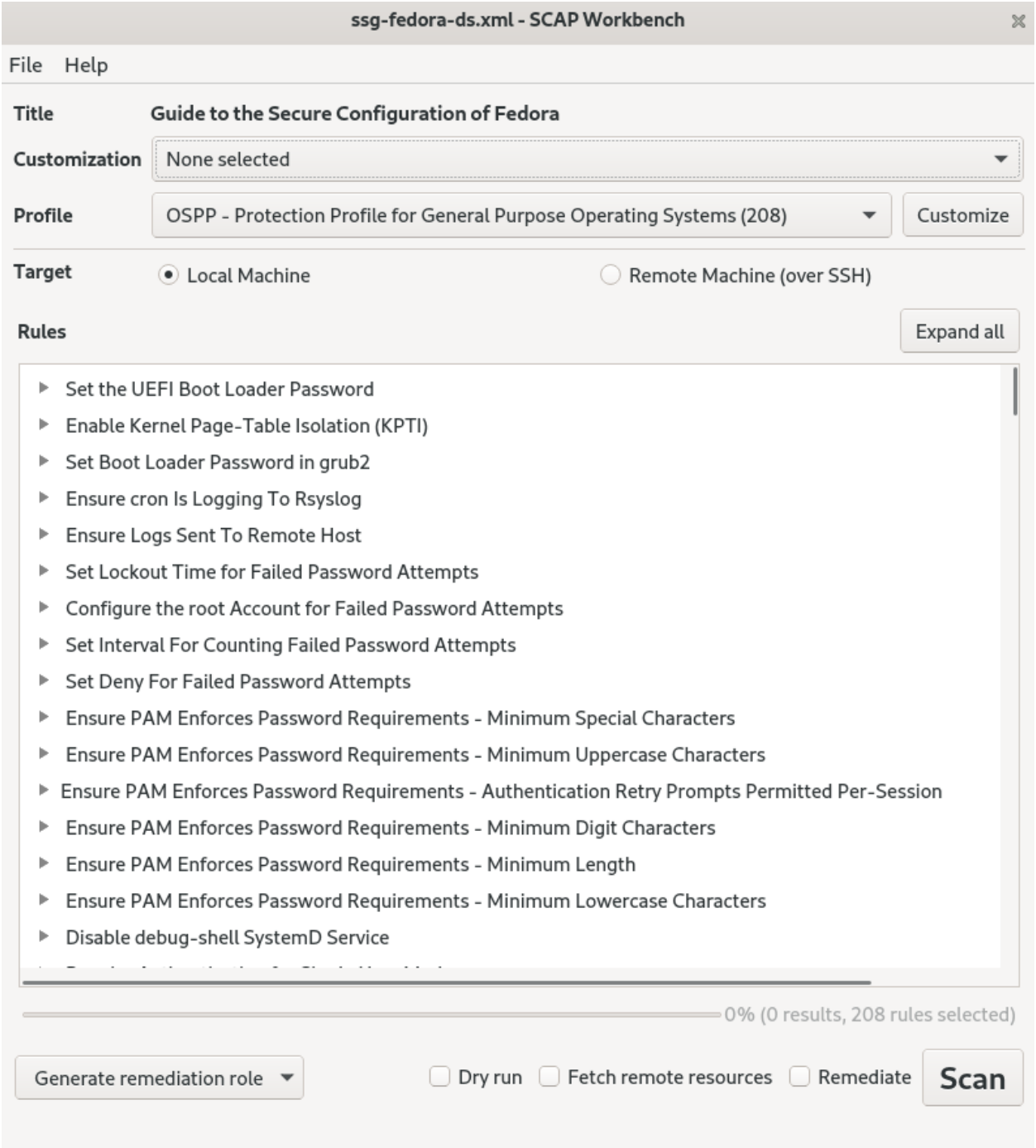
Slika 3.12. Prikaz instaliranog „SCAP Workbench“ alata na Fedora 31 sustavu

Prilikom otvaranja „SCAP Workbench“ alata pojavljuje nam se jedan od pomoćnih alata „SCAP Security Guide“ prikazani prema slici 3.13. SCAP Security Guide pomaže pri odabiranju ispravne sigurnosne politike za sustav i sadrži sigurnosna pravila koja su dostupna za brojne operativne sustave i druge softvere.



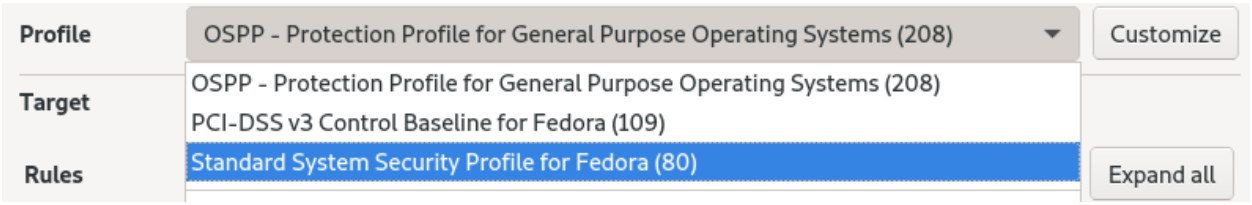
Slika 3.13. Prikaz „SCAP Security Guide“ alata

Nakon odabira željenog sustava ili softvera za skeniranje otvara nam se grafičko sučelje „SCAP Workbench“ alata prikazano prema slici 3.14.



Slika 3.14. Prikaz „SCAP Workbench“ alata

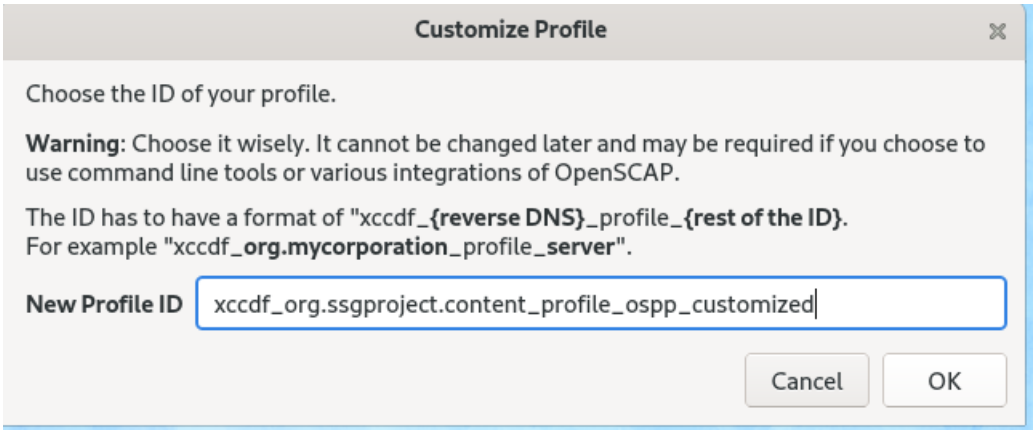
Nakon otvaranja pretpostavljenih postavki alata, moramo postaviti alat kako bi skenirao naš sustav u odnosu na definirana pravila. Skupine pravila možemo odabrati na „Profile“ izborniku.



Slika 3.15. Odabir zadanog profila za sustav

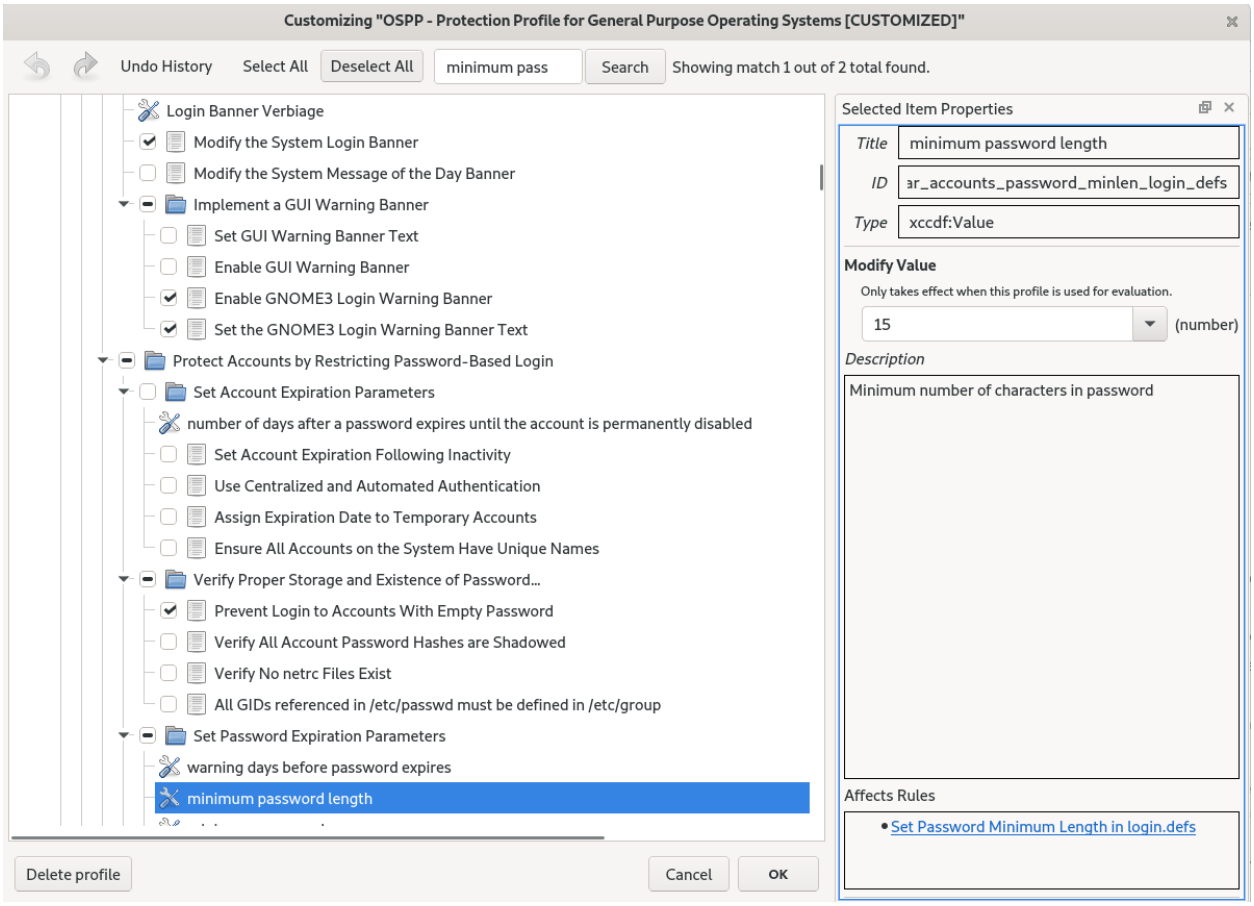
U slučaju da odabrani profil želimo prilagoditi našem sustavu tako da dodamo ili uklonimo pravila, to možemo odabirom na „Customize“.

Nakon odabira otvorit će nam se prozor koji traži ime ID profila u slučaju da budemo mijenjali nešto. Prilikom odabira moramo biti sigurni u odabrano ime pošto ga kasnije nije moguće mijenjati kao prema slici 3.16.



Slika 3.16. Prikaz prozora za određivanje imena profila

Nakon odabiranja imena za novi modificirani profil, pojavljuje nam se novi prozor u kojemu možemo odabrati koja ćemo pravila dodati to jest ukloniti iz našeg profila. U slučaju da ne želimo zadržati modificirani profil, uvijek možemo vratiti profil u prvobitno stanje tako da izađemo iz novootvorenog prozora ili odabirom na „Delete profile“. Neka određena pravila moguće je i modificirati prilikom njihovog odabira kao što je prikazano prema slici 3.17. Nakon završetka uređivanja profila, možemo spremiti profil i koristiti ga u slučaju potrebe.



Slika 3.17. Prikaz modificiranja određenog profila

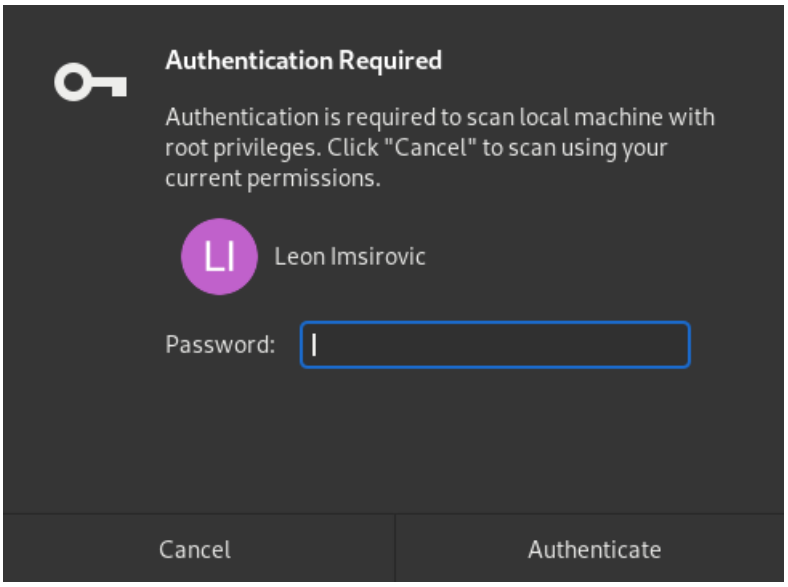
Pošto ovaj diplomski rad prikazuje slučaj sa 80 standardnih pravila za Fedora operativni sistem, pravila se nalaze već u pretpostavljenom profilu te nije potrebno ništa uređivati. Skeniranje sustava odvija se na lokalnom sustavu. Prilikom skeniranja možemo odabrati razne mogućnosti sustava. Ako želimo skenirati sustav s odabranim postavaka preko naredbenog retka, možemo označiti „Dry Run“ te ćemo tako prilikom pokušaja skeniranja dobiti naredbu kao prema slici 3.18.

Slika 3.18. Naredba za skeniranje sustava pomoću naredbenog retka

U slučaju da nam neki od sadržaja za skeniranje nedostaje, možemo označiti „Fetch remote resources“ te će tako sam alat preuzeti sve resurse koji su potrebni za skeniranje sustava ako nedostaju. Zadnja je mogućnost prije skeniranja sustava „Remediate“. Ako odaberemo tu mogućnost, sustav će prvo biti skeniran i nakon toga sva pravila koja nisu zadovoljila pokušat će se ispraviti prema određenim standardima. Korištenje ove mogućnosti prikazano je prilikom testiranja rezultata.

3.2.1. Sustav prije korištenja rješenja

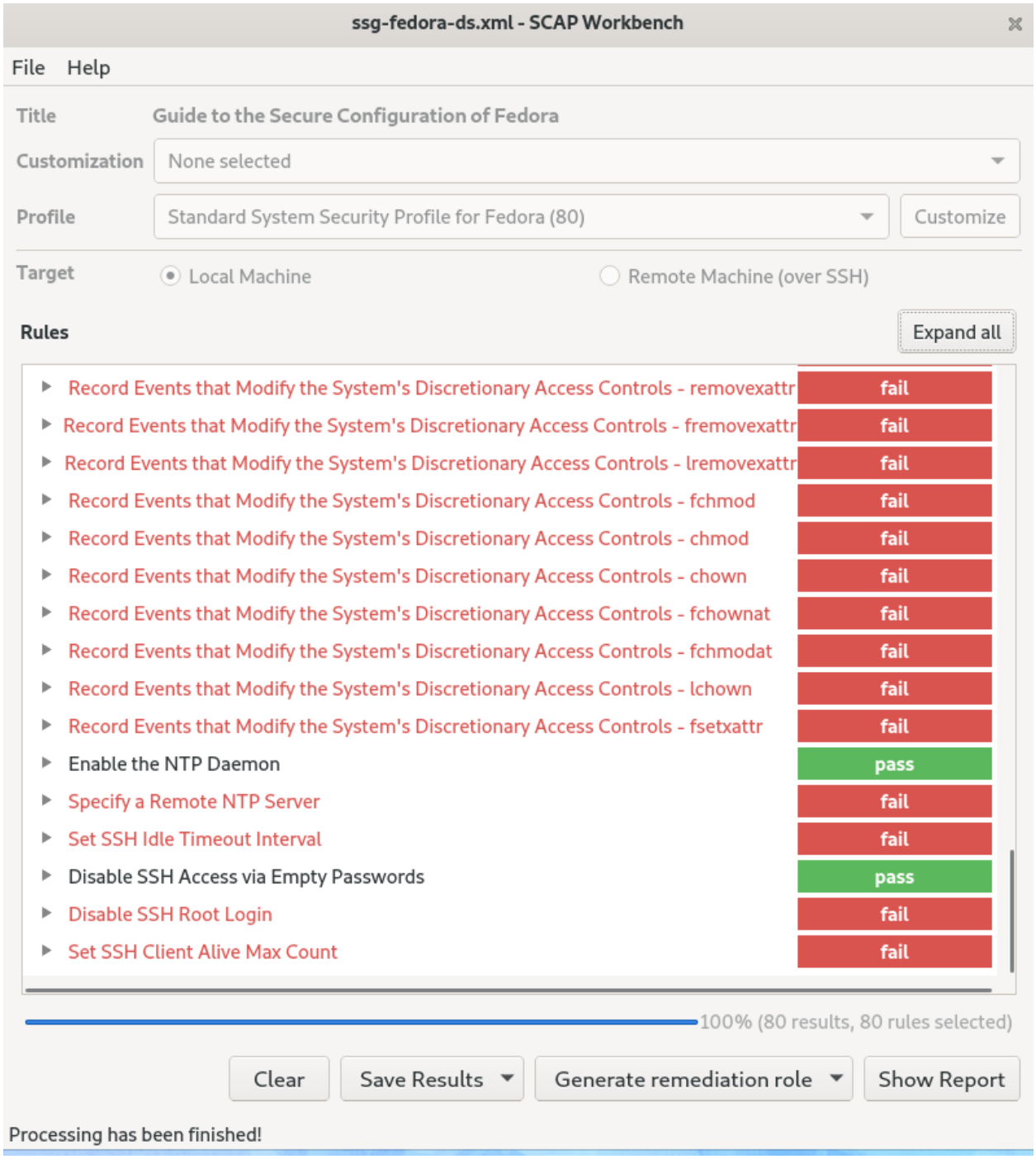
Nakon pojašnjavanja sadržaja samog skenera možemo skenirati sustav. Nakon pokretanja skeniranja sustav će nas tražiti lozinku za administratora. Razlog je tomu taj što određene konfiguracije na sustavu trebaju imati ovlasti administratora kako bi se moglo pristupiti kao prema slici 3.19. Ako želimo nastaviti skeniranje bez administratorskih ovlasti, možemo samo izaći iz traženog prozora te će sva pravila koja zahtijevaju te ovlasti biti prikazana kao da nisu zadovoljila pravila standarda.



Slika 3.19. Prikaz potrebne autentifikacije

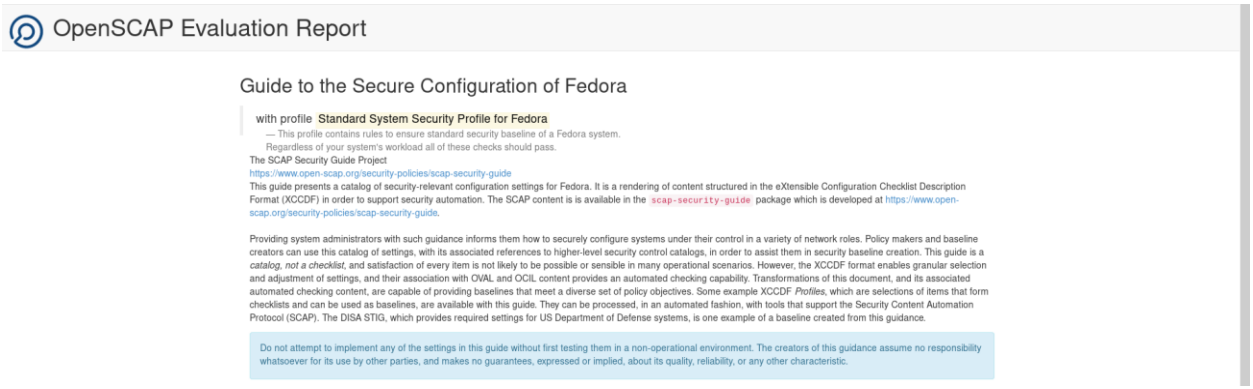
Nakon skeniranja sustava imamo prikaz svih pravila koja su zadovoljila „pass“ i ona koja nisu zadovoljila „fail“ definirane standarde kao prema slici 3.20. Ako ne želimo više pregledavati rezultate, možemo odabrati „Clear“ i nakon toga se prozor vraća u prvobitno stanje kao prema slici

3.14. Ako želimo spremiti rezultate u određenom formatu, možemo odabrati „Save Results“ i odgovarajući format. Isto tako možemo odabrati „Generate remediation role“ te odabrati jednu od ponuđenih opcija i generirati skriptu koja će pomoći riješiti pravila koja nisu zadovoljila standarde.



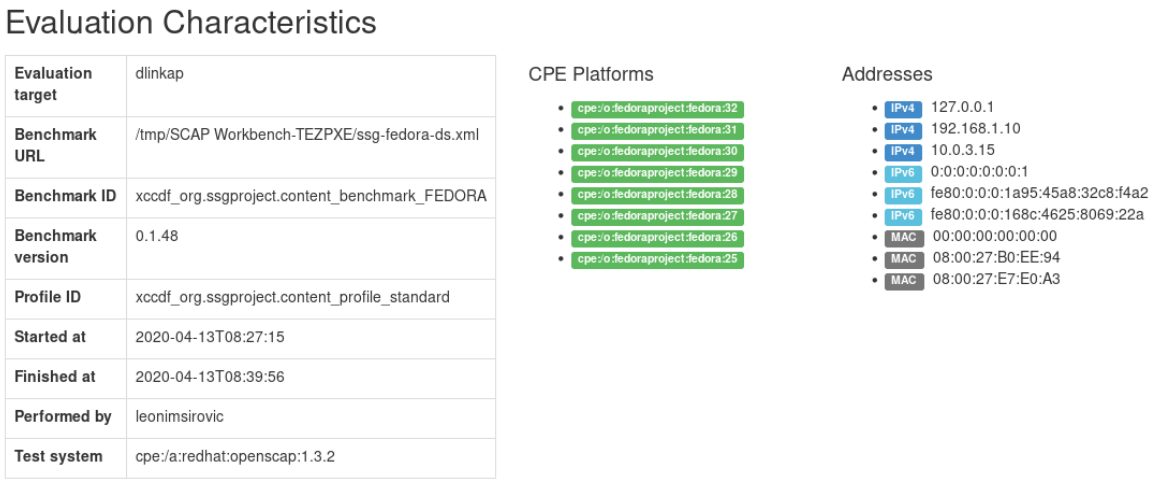
Slika 3.20. Prikaz „SCAP Workbench“ alata nakon skeniranja

Ako želimo vidjeti detaljnije rezultate skeniranja, možemo odabrati „Show Report“ te nam se rezultati otvaraju u obliku web stranice kao prema slici 3.21.



Slika 3.21. Prikaz uputa o skeneru

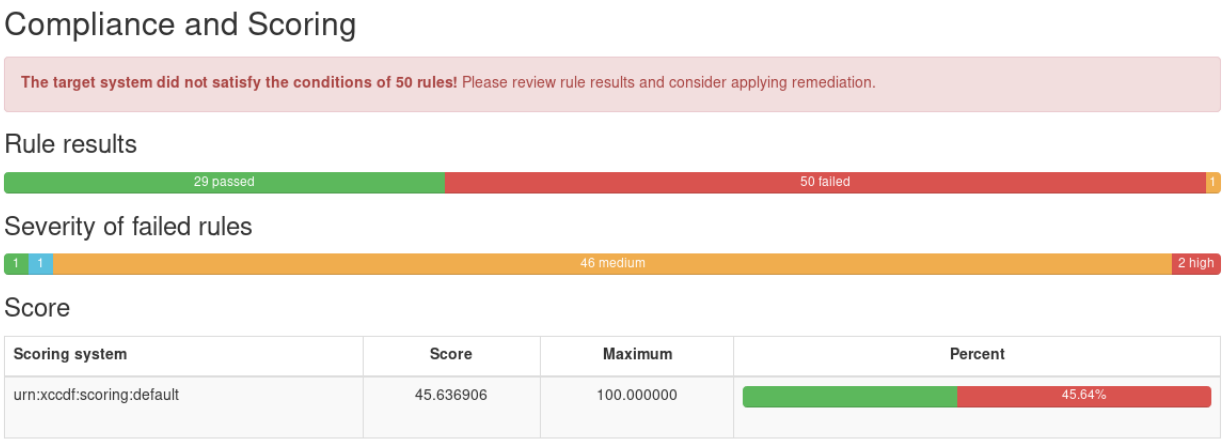
Osim o uputama o skeneru na stranici se nalaze i detalji samog skeniranja kao prema slici 3.22. Ti detalji pružaju dodatnu sigurnost prilikom provjeravanja korisnika koji su koristili skener.



Slika 3.22. Prikaz osnovnih informacija skeniranja

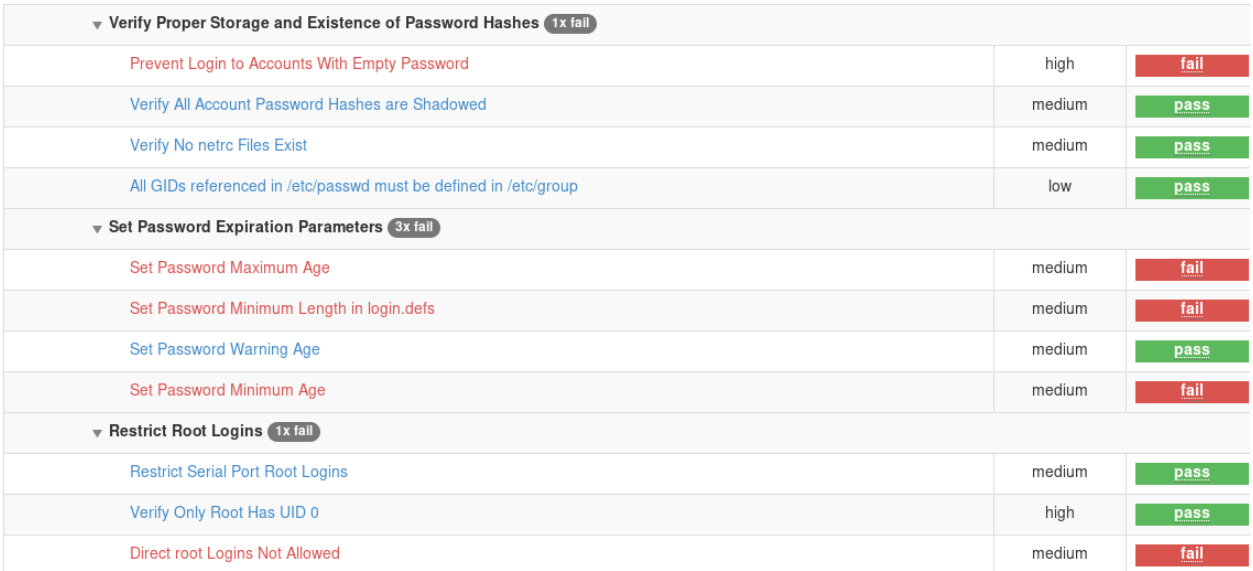
Najvažniji dio samih rezultata ocjena je skeniranog sustava. Prilikom svakog skeniranja svaki od pravila ima svoju ozbiljnost te o njegovoj prolaznosti ovisi cjelokupna ocjena sustava. Prema slici 3.23. vidi se broj pravila koja su zadovoljila standarde (zeleno), broj pravila koja nisu zadovoljila standarde (crveno) te pravila koja nisu dobro skenirana (žuto). Osim samog broja odmah ispod nalazi se i ozbiljnost samih rezultata. Prikazana ozbiljnost rezultata prikazana je samo za pravila koja nisu zadovoljila standarde. Velika ozbiljnost pravila prikazana je crvenom bojom, srednja ozbiljnost žutom, te mala ozbiljnost plavom. Zelenom su označena ona pravila koja nisu dobro

skenirana. Kako bi administratoru sustava bilo jednostavnije prikazati, ukupan rezultat skeniranja sustava također je prikazan u postocima.



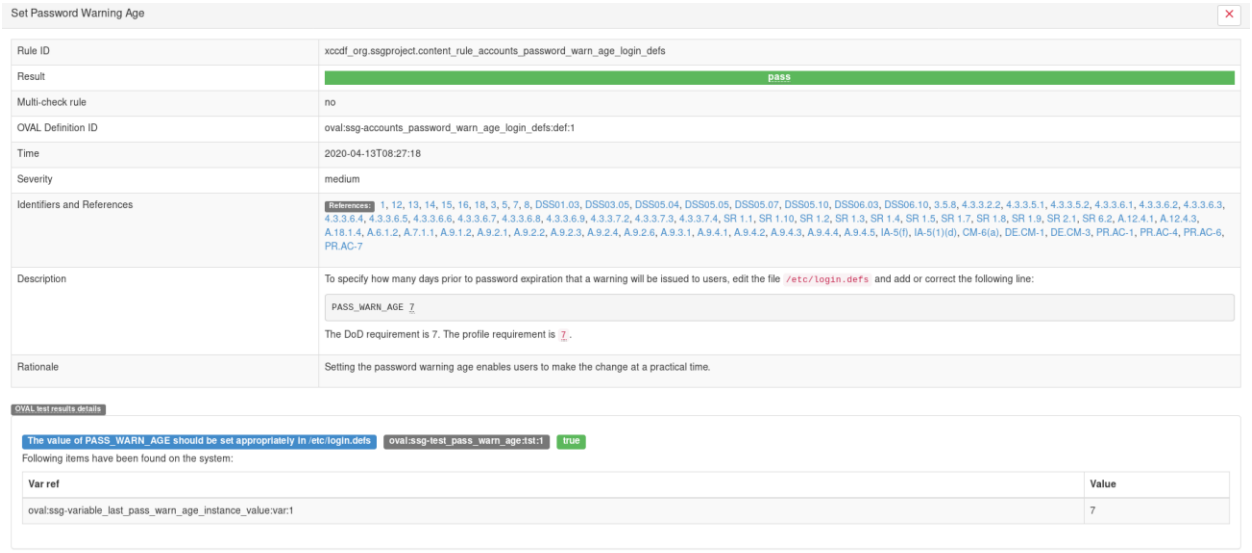
Slika 3.23. Prikaz rezultata skeniranja

Osim ukupnih rezultata imamo i svako pravilo posebno prikazano prema slici 2.24.



Slika 3.24. Prikaz rezultata za pojedine rezultate

Ako odaberemo jedno od pravila, možemo dobiti detaljan opis odabranog pravila, razloge zašto pravilo treba riješiti i na koji način. (Slika 3.25.)

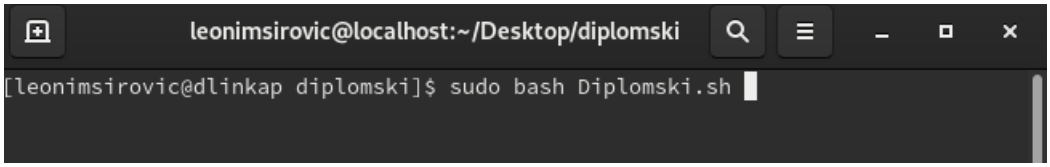


Slika 3.25. Detaljniji prikaz odabranog pravila

3.2.2. Korištenje skripte

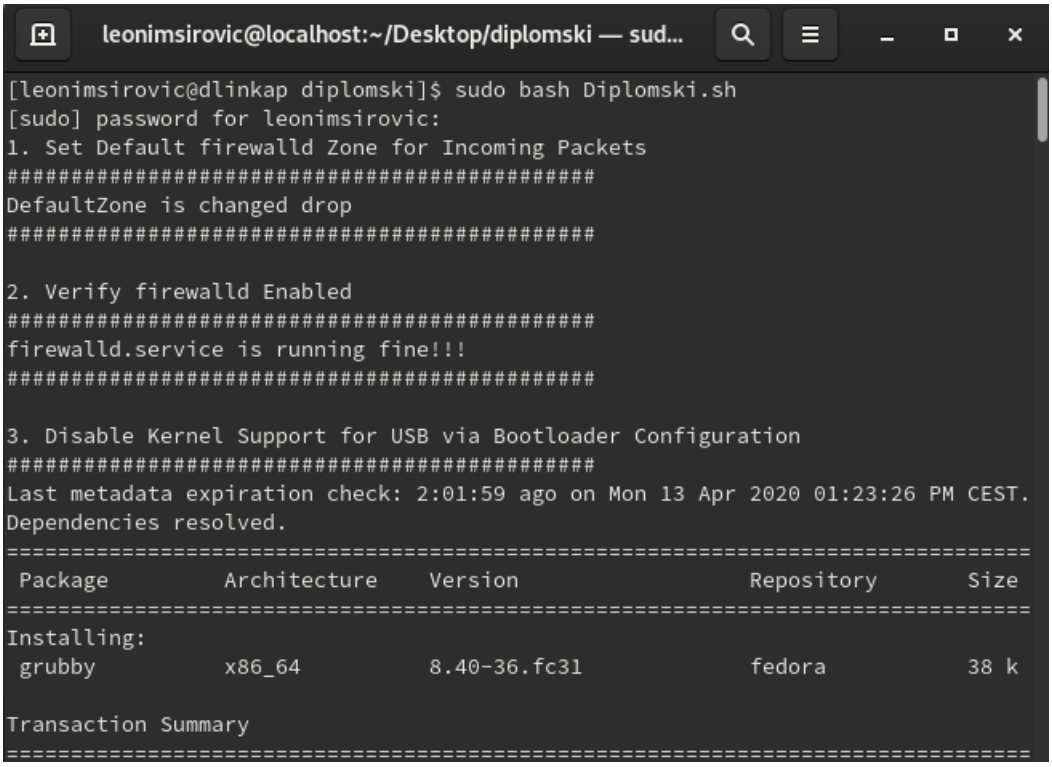
Nakon skeniranja sustava možemo vidjeti da ovisno o broju pravila koja nisu zadovoljila standarde, sav posao oko pokušavanja osiguravanja sustava traje dugo vremena. Mogućnost automatizacije bilo kojeg dijela procesa osiguravanja sustava uvelike pomaže. Pa je tako cilj korištenja rješenja (Prilog A), osim osiguravanja sustava, korištenje automatizacije kako bi bilo koji korisnik rješenja mogao doći do rezultata bez puno posla.

Rješenje se temelji na velikom broju naredbi koje umjesto korisnika obavljaju posao. Za korištenje rješenja potrebno je imati administratorske ovlasti kako bi se promjene mogle odvijati na za to potrebnim konfiguracijama na sustavu. Samo korištenje rješenja (skripte) dosta je jednostavno. Korisnik skripte treba samo pokrenuti skriptu kao prema slici 3.26., a sve će ostalo same naredbe unutar skripte obaviti.



Slika 3.26. Prikaz naredbe za pokrivanje rješenja (skripte)

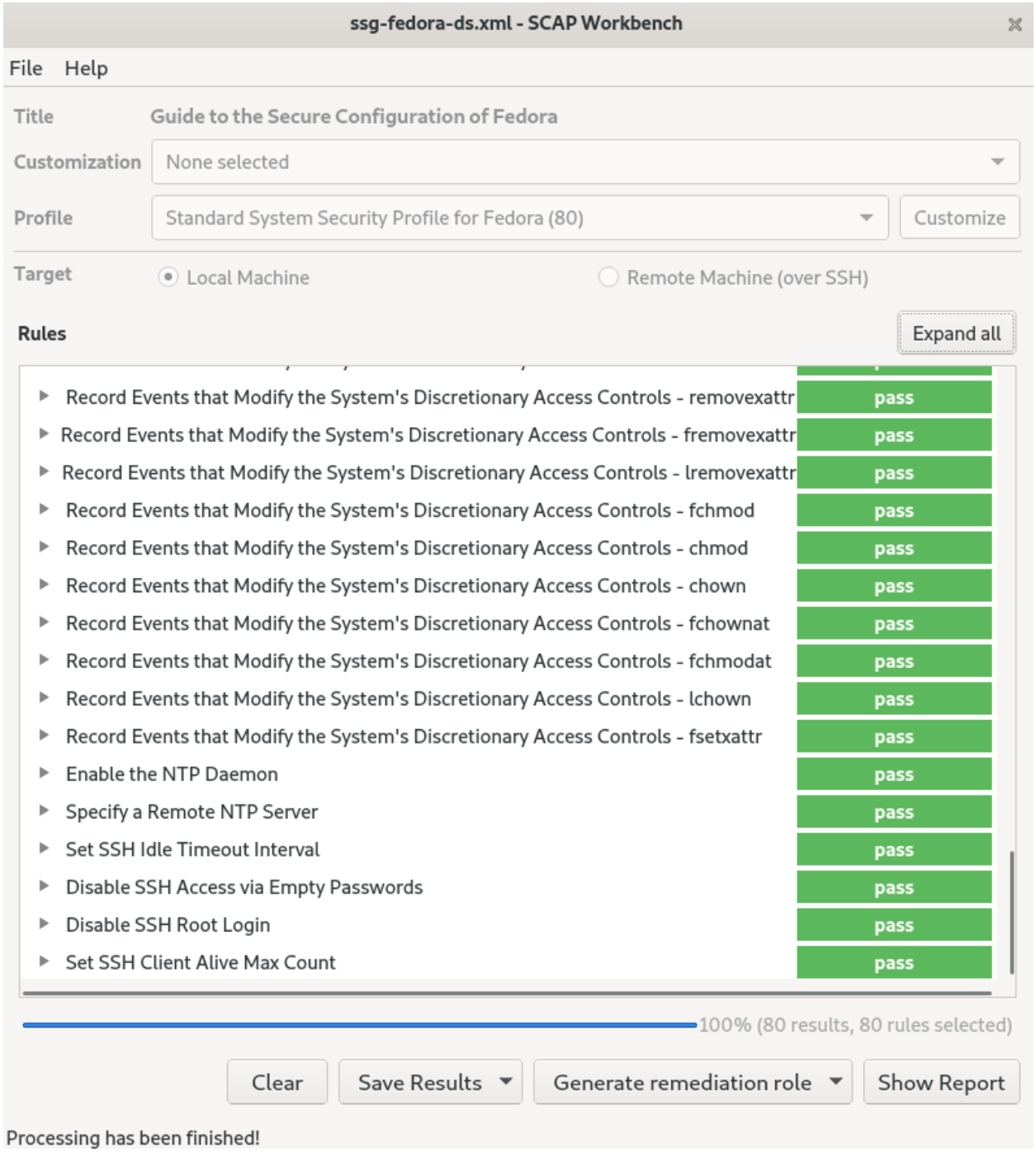
Nakon pokretanja skripte redoslijed rješavanja pravila formiran je prema prilogu A. Pa tako nakon pokušavanja rješavanja potrebnih pravila ispisuje se status u naredbenom retku kao prema slici 3.27.



Slika 3.27. Prikaz ispisa prilikom korištenja skripte

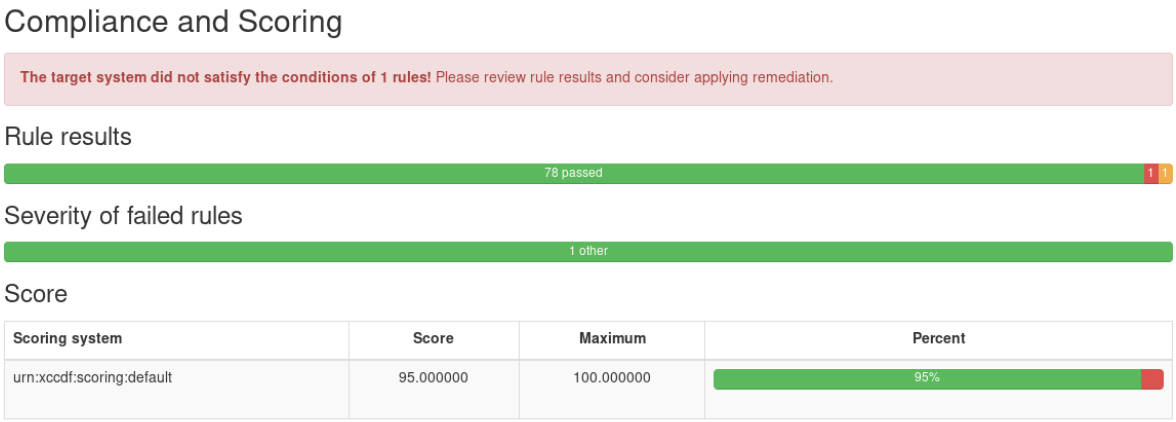
3.2.3. Sustav poslije korištenja rješenja

Poslije korištenje rješenja potrebno je ponovno skenirati sustav kako bismo mogli vidjeti kakvi su rezultati sigurnosti u odnosu na traženo prema standardu kao prema slici 3.28. Nakon dobivanja rezultata sustava, kako bismo detaljnije vidjeli dobivene rezultate, potrebno je odabrati „Show Report“.



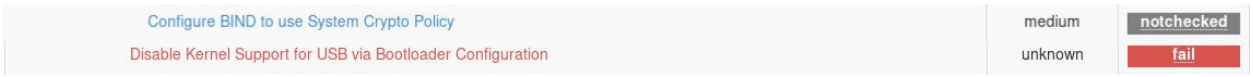
Slika 3.28. Prikaz „SCAP Workbench“ alata nakon skeniranja

Nakon detaljnijeg promatranja sadržaja izvještaja nakon skeniranja prema slici 3.29. možemo zaključiti kako su se rezultati nakon korištenja skripte uvelike poboljšali.



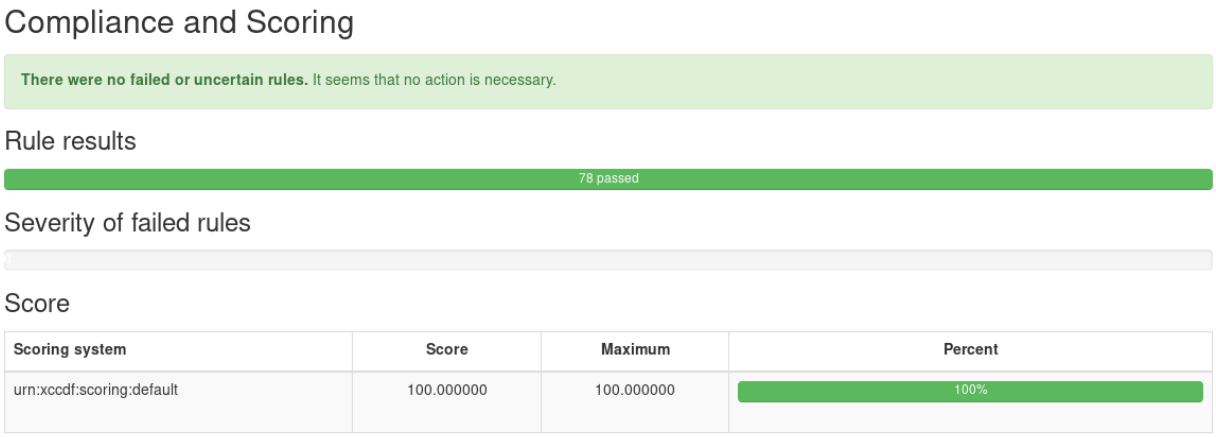
Slika 3.29. Prikaz rezultata nakon korištenja skripte

Iako skener nakon korištenja skripte pokazuje odlične rezultate od 95%, skener prikazuje i dva pravila koja nisu u potpunosti u pravilu sa sigurnosnim standardima. Nakon detaljnije provjere tih pravila došlo se do zaključka da skener ima grešku prilikom skeniranja tih dviju pravila, na što ukazuje i prikaz njihovih rezultata i „ozbiljnosti“ u skeneru. Prema slici 3.30. može se navedeno i zaključiti. Pravilo „Configure BIND to use System Crypto Policy“ nema mogućnost provjere, a pravilu „Disable Kernel Support for USB via Bootloader Configuration“ skener ne može odrediti ozbiljnost pravila te ga stavlja u skupinu onih koje ne zadovoljavaju standard.



Slika 3.30. Prikaz loše skeniranih pravila

Ako uklonimo pravila koja nije moguće pravilno prikazati pomoću skenera, dobijemo rezultate koji daju ocjenu sustava 100% kao prema slici 3.31.

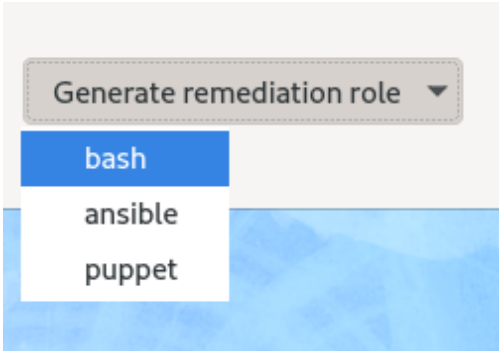


Slika 3.31. Prikaz rezultata nakon uklanjanja pravila koja nisu dobro skenirana

3.3. Testiranje rješenja

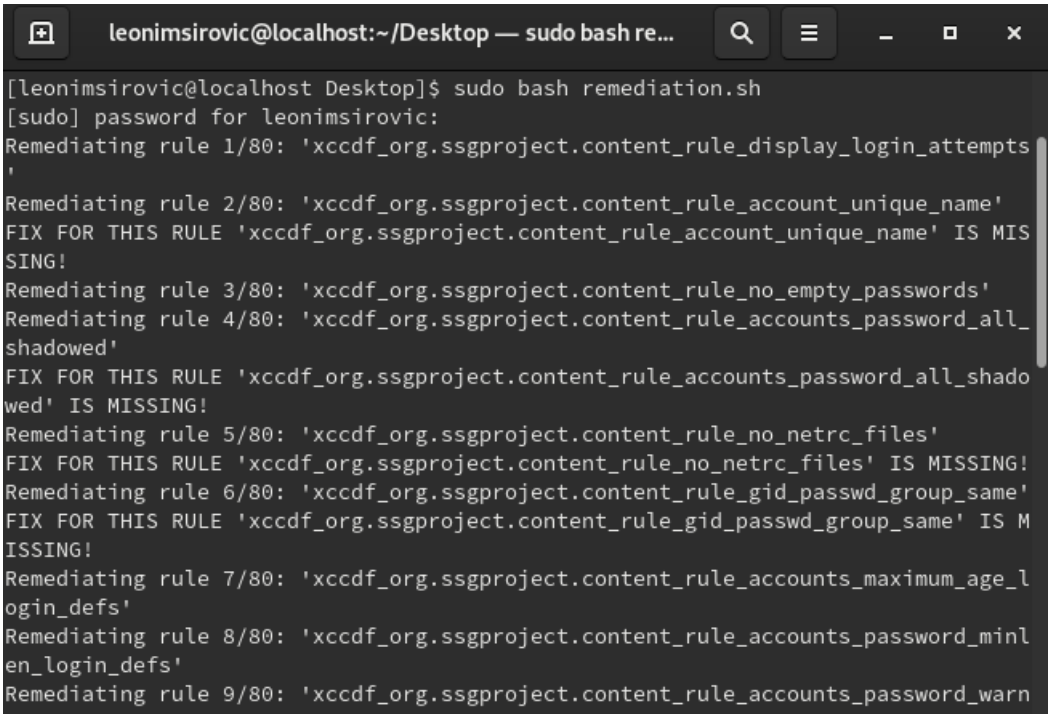
3.3.1. „SCAP Workbench“ rješenje

Za testiranje dobivenih rezultata prikazat će se način korištenja skenera „SCAP Workbench“ koji daje mogućnost rješavanja pravila koja nisu zadovolja standarde. Radi istih uvjeta testiranja rezultata uzet je identični sustav kao prema slici 3.11. Prilikom skeniranja sustava možemo odabrati mogućnost skenera „Generate remeditiation role“ te odabrati u kojem obliku nam treba skripta kao prema slici 3.32.



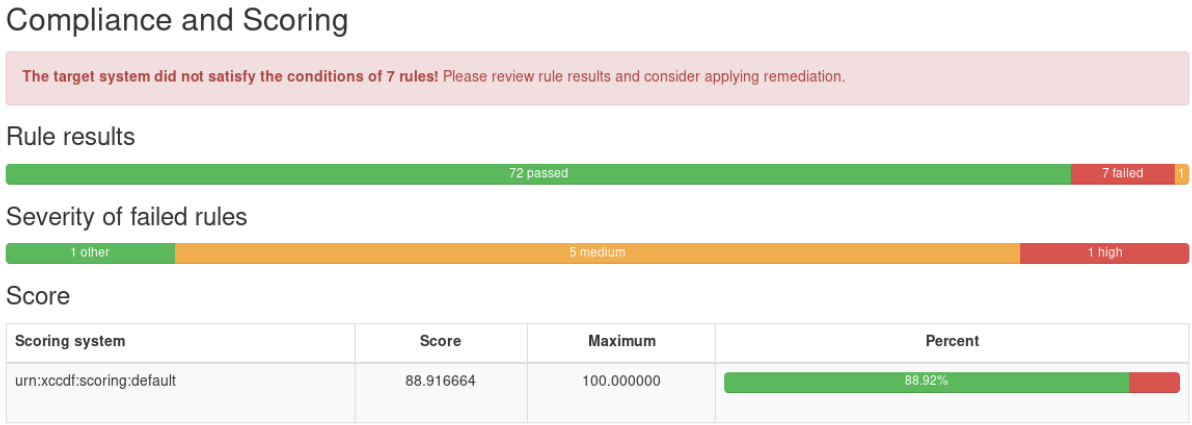
Slika 3.32. Prikaz odabira oblika skripte

Radi uspoređivanja rezultata odabrana je bash skripta. Generirana skripta naziva se „remediation.sh“. Skripta je pokrenuta na isti način kao prema slici 3.26. te daje sljedeće rezultate prilikom izvođenja (Slika 3.33.)



Slika 3.33. Prikaz ispisa skripte „remediation.sh“

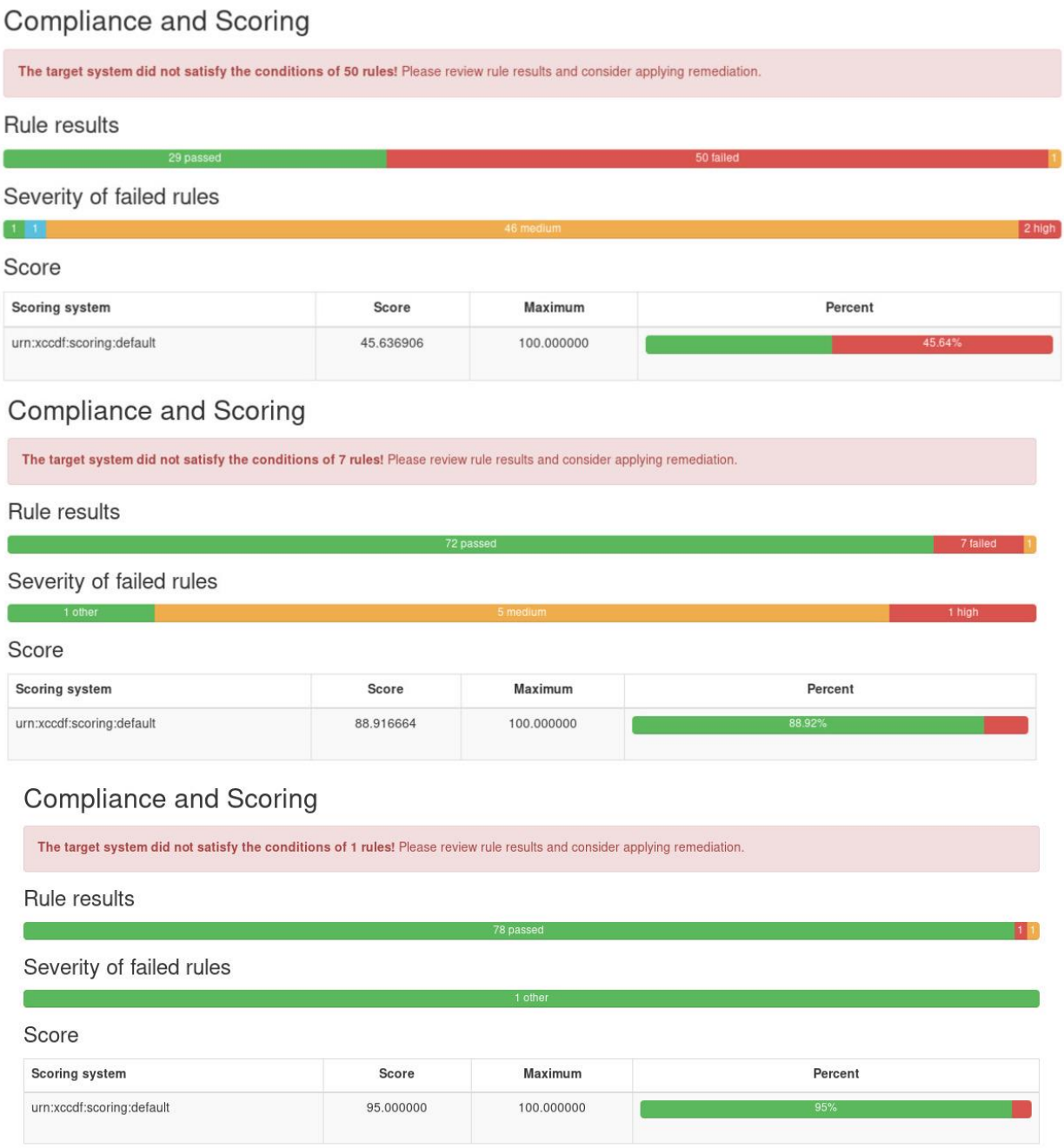
Nakon izvođenja skripte sustav je skeniran te u izvješću daje rezultate kao prema slici 3.34.



Slika 3.34. Prikaz najboljeg rješenja uz pomoć SCAP Workbench alata

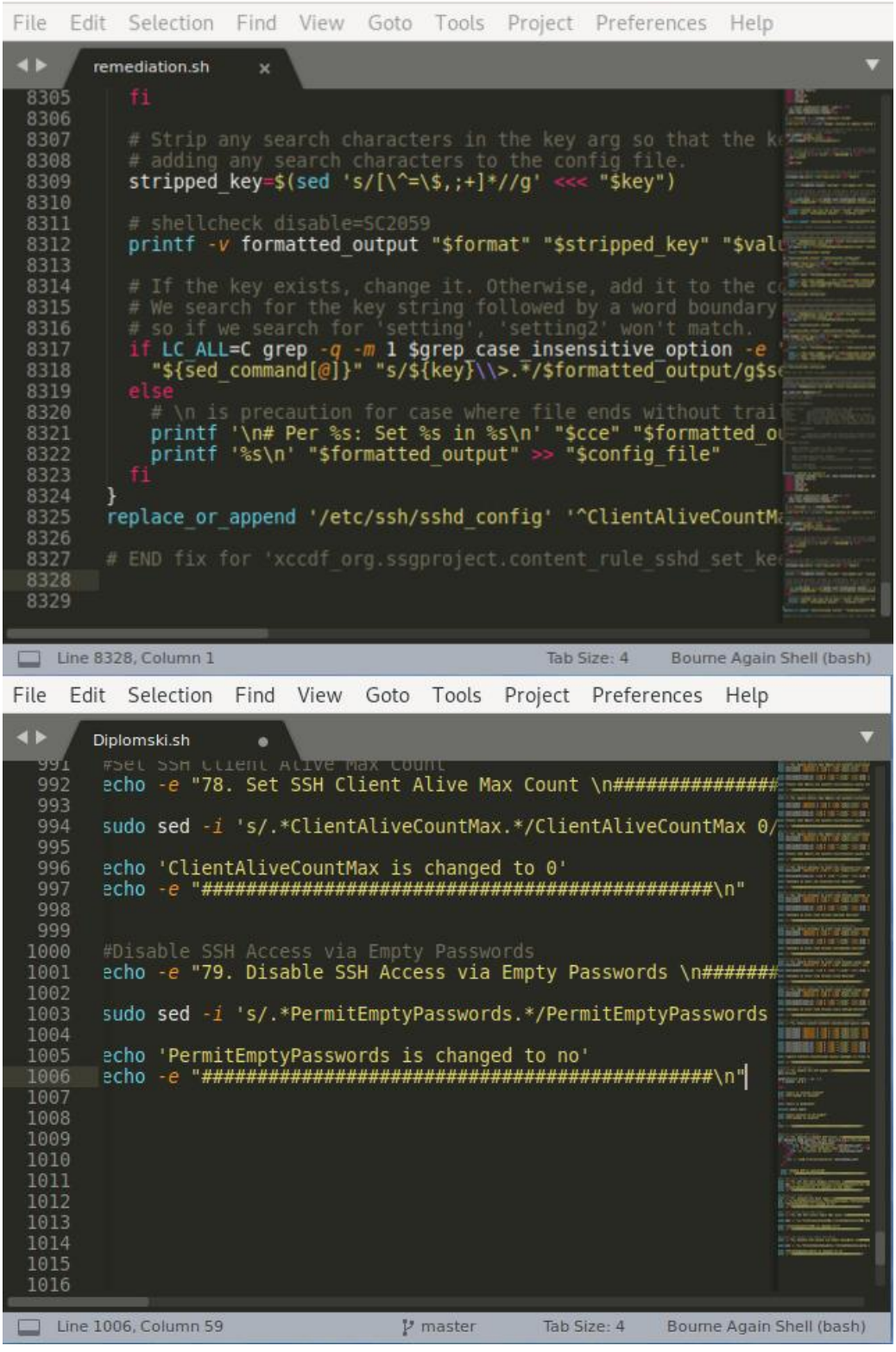
3.3.2. Usporedba rezultata

Prema slici 3.35. prikazani su: rezultati za tek instalirani sustav, sustav nakon najboljeg mogućeg rješenja od strane SCAP Workbench i sustav nakon korištenja skripte „Diplomski.sh“ (Prilog A), tim redom. Da se zaključiti kako su rezultati nakon skripte „Diplomski.sh“ (Prilog A) bolji nego nakon skripte „remediation.sh“ (SCAP Workbench). Osim ocjene rezultata koji predstavlja glavnu razliku u rezultatima, može se vidjeti i razlika u broju pravila koji ne zadovoljavaju standard.



Slika 3.35. Prikaz dobivenih rezultata nakon skeniranja

Iako je ocjena SCAP Workbench alata najbolji pokazatelj valjanosti skripte „Diplomski.sh“ (prilog A), velika prednost ranije spomenute skripte kompleksnost je i veličina te skripte. Na slici 3.36. uspoređene su skripte „remediation.sh“ i „Diplomski.sh“ (Prilog A).



Slika 3.36. Usporedba skripti „remediation.sh“ (SCAP Workbench) i „Diplomski.sh“ (Prilog A).

Skripta „remediation.sh“ (SCAP Workbench) sadrži preko osam tisuća linija koda, dok skripta „Diplomski.sh“ (Prilog A) ima približno tisuću linija koda. Pri tome je prikazano da je uz manje koda koji je lakši za održavanje moguće dobiti bolje rezultate.

4. ZAKLJUČAK

U diplomskom radu prikazana je važnost održavanja sigurnosti sustava pomoću automatizacije. Iz dana u dan količina prijetnji koje mogu naštetiti sustavu i njegovim elementima raste pa je zato potrebno koristiti pravila i smjernice koje pomažu prilikom zaštite sustava. SCAP postaje sve popularniji prilikom zaštite sustava. OpenSCAP je jedan od najpopularnijih alata koji koristi standarde SCAP protokola. Učestalo unaprjeđenje OpenSCAP alata pokazuje koliko je sigurnost sustava bitna te koliko brzina rješavanja problema utječe na sigurnost sustava. Osim brzine rješavanja problema jako je bitno da se sustav učestalo provjerava kako bi se sigurnosni problemi mogli uočiti na vrijeme. Iako sam alat može dati određene rezultate koji poboljšavaju sigurnost sustava, najbolji je način zaštite posebno rješenje koje će najviše odgovarati sustavu, što pokazuju i dobiveni rezultati ovoga rada. Svaki sustav ima određene elemente koji imaju veću važnost za sustav pa je prilikom osiguravanja sustava potrebno posvetiti veću pozornost na te elemente. Kako se napadi na sustav mijenjaju, tako je potrebno da se i rješenja prilagođavaju prilikom zaštite sustava. Iako sustav neće nikada biti u potpunosti siguran, potrebno je da se odradi sve kako bi sustav bio što sigurniji te da se otežа svaki mogući napad na njega. (Travanj 2020.)

LITERATURA

[1] HrOpenWiki: „Linux“ https://wiki.open.hr/wiki/Linux (siječanj, 2020.)

[2] Fedora: „Fedora“ https://getfedora.org/ (siječanj, 2020.)

[3] GNU Operating System: „Bash Reference Manual“

https://www.gnu.org/software/bash/manual/bash.html#What-is-Bash_003f (siječanj, 2020.)

[4] OpenSCAP: „OpenSCAP“ https://www.open-scap.org/ (siječanj, 2020.)

[5] DAVOUD TEIMOURI - VIRTUALIZATION AND DATA CENTER BLOG: „What is OpenSCAP?“ https://www.teimouri.net/what-is-openscap/ (veljača, 2020.)

[6] Centar informacijske sigurnosti(CIS): „Zaštita mreže – vatrozid“ https://www.cis.hr/sigurnosni-alati/zastita-mreze-vatrozid.html (veljača, 2020.)

[7] AIDE: „About AIDE“ https://aide.github.io/ (veljača, 2020.)

[8] Libreswan: „Libreswan VPN software“ https://libreswan.org/ (veljača, 2020.)

[9] Massachusetts Institute of Technology: „Kerberos: The NetworkAuthentication Protocol“ http://web.mit.edu/Kerberos/ (ožujak, 2020.)

[10] OpenSSL: „OpenSSL Blog“ https://www.openssl.org/blog/ (ožujak, 2020.)

[11] Archlinux: „PAM“ https://wiki.archlinux.org/index.php/PAM (ožujak, 2020.)

[12] THE GEEK DIARY: „Linux OS service ‘auditd’“ https://www.thegeekdiary.com/linux-os-service-auditd/ (ožujak, 2020.)

[13] CERT: „Modeli kontrole pristupa“ https://www.cert.hr/wp-content/uploads/2019/04/CCERT-PUBDOC-2008-01-218.pdf (travanj, 2020.)

[14] HrOpenWiki: „NTP“ https://wiki.open.hr/wiki/Dokumentacija_NTP (travanj, 2020.)

[15] LinuxConfig: „Linux DNS server BIND configuration“ https://linuxconfig.org/linux-dns-server-bind-configuration (travanj, 2020.)

SAŽETAK

Naslov: OpenSCAP sigurnosno očvršćivanje i automatizacija sustava stvarnog vremena

U ovome radu prikazani su načini osiguravanja sustava uz pomoć automatizacije. Detaljno se pojašnjava praktično korištenje alata OpenSCAP i njemu pripadajućeg alata SCAP Workbench. Sva su pravila detaljnije objašnjena kako bi se prikazala njihova važnost za sigurnost sustava. Prikazuju se načini osiguravanja sustava pomoću skripte napravljene na temelju zadanih pravila i sigurnosnih standarda. Na kraju rada uspoređeni su rezultati između skripte napravljene u svrhu diplomskog rada i trenutno najboljeg rezultata od strane OpenSCAP alata. Rezultati su uspoređeni na temelju ukupne ocjene sigurnosti sustava koju daje sam alat te složenosti napisanog koda u oba slučaja.

Ključne riječi: OpenSCAP, informacijska sigurnost, automatizacija, Linux, Bash

ABSTRACT

Title: OpenSCAP security hardening and real time system automation

This paper describes how to provide secure system with automation. The practical use of OpenSCAP and its associated SCAP Workbench is explained in detail. All the rules are explained in more detail to demonstrate their importance for system security. Ways to secure your system using a script created based on default rules and security standards are presented. At the end of the paper, the results were compared between a script made for the purpose of the graduate thesis and the currently best result using the OpenSCAP tool. The results were compared based on the overall system security rating given by the tool itself and the complexity of the code written in both cases.

Keywords: OpenSCAP, information security, automation, Linux, Bash

ŽIVOTOPIS

Leon Imširović rođen je 27.6.1996. u Osijeku. Pohađao je Isusovačku klasičnu gimnaziju s pravom javnosti u Osijeku. Sudjelovao je na brojnim županijskim natjecanjima iz informatike tijekom srednjoškolskog obrazovanja. Zbog zalaganja i odličnog uspjeha, 2015. godine ostvaruje izravan upis na Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Akademske 2015./2016. upisuje Sveučilišni preddiplomski studij, a 2018./2019. upisuje Sveučilišni diplomski studij računarstva te je trenutno 2. godina diplomskog studija računarstva smjer Informacijske i podatkovne znanosti. Stipendist je u ATOS-u te je sudjelovao i organizirao brojne konferencije vezane za razne tehnologije. U slobodno vrijeme bavi se sportom i proučavanjem raznih tehnologija povezanih za sigurnost računalnih sustava.

Leon Imširović

PRILOG A (Diplomski.sh)

```
#!/bin/bash

#Set Default firewalld Zone for Incoming Packets
echo -e "1. Set Default firewalld Zone for Incoming Packets
\n##############################################"

sudo sed -i 's/.*DefaultZone.*/DefaultZone=drop/g' /etc/firewalld/firewalld.conf

echo 'DefaultZone is changed drop'
echo -e "##############################################\n"

# Verify firewalld Enabled

echo -e "2. Verify firewalld Enabled \n##############################################"

-------
serv=firewalld.service

STATUS=$(systemctl show -p ActiveState --value firewalld.service )
if [ $STATUS = 'active' ]

then

echo "$serv is running fine!!!"

else

echo "$serv is down/dead"

service $serv start

echo "$serv service is UP now!!!"

fi

echo -e "##############################################\n"

#Disable Kernel Support for USB via Bootloader Configuration
echo -e "3. Disable Kernel Support for USB via Bootloader Configuration
\n##############################################"
if ! rpm -q --quiet "grubby" ; then
    dnf install -y "grubby"
fi
# Correct the form of default kernel command line in /etc/default/grub
if ! grep -q ^GRUB_CMDLINE_LINUX=\".*nousb.*\" /etc/default/grub;
then
  # Edit configuration setting
  # Append 'nousb' argument to /etc/default/grub (if not present yet)
  sed -i "s/\(GRUB_CMDLINE_LINUX=\)\"\(.*\)\"/\1\"\2 nousb\"/" /etc/default/grub

  # Edit runtime setting
  # Correct the form of kernel command line for each installed kernel in the bootloader
  /sbin/grubby --update-kernel=ALL --args="nousb"
fi
echo 'Kernel Support for USB Disabled'
echo -e "##############################################\n"



echo -e "4. Verify that System Executables Have Root Ownership
\n##############################################"
find -L /bin \! -user root -exec chown root {} \;
find -L /sbin \! -user root -exec chown root {} \;
find -L /usr/bin \! -user root -exec chown root {} \;
find -L /usr/libexec/ \! -user root -exec chown root {} \;
find -L /usr/local/bin \! -user root -exec chown root {} \;
find -L /usr/local/sbin \! -user root -exec chown root {} \;
find -L /usr/sbin \! -user root -exec chown root {} \;


echo 'System Executables Have Root Ownership'

echo -e "##############################################\n"

#Verify that Shared Library Files Have Root Ownership
echo -e "5. Verify that Shared Library Files Have Root Ownership
\n##############################################"
find -L /lib \! -user root -exec chown root {} \;
find -L /lib64 \! -user root -exec chown root {} \;
find -L /usr/lib \! -user root -exec chown root {} \;
find -L /usr/lib64 \! -user root -exec chown root {} \;

```

```
79  echo 'Shared Library Files Have Root Ownership'
80
81  echo -e "############################################\n"
82
83  #Verify that Shared Library Files Have Restrictive Permissions
84  echo -e "6. Verify that Shared Library Files Have Restrictive Permissions
\n############################################"
85
86  find -L /lib -perm /022 -type f -exec chmod go-w {} \;
87  find -L /lib64 -perm /022 -type f -exec chmod go-w {} \;
88  find -L /usr/lib -perm /022 -type f -exec chmod go-w {} \;
89  find -L /usr/lib64 -perm /022 -type f -exec chmod go-w {} \;
90
91  find -L /lib -perm /022 -type d -exec chmod go-w {} \;
92  find -L /lib64 -perm /022 -type d -exec chmod go-w {} \;
93  find -L /usr/lib -perm /022 -type d -exec chmod go-w {} \;
94  find -L /usr/lib64 -perm /022 -type d -exec chmod go-w {} \;
95
96  echo 'Shared Library Files Have Restrictive Permissions'
97
98  echo -e "############################################\n"
99
100 #Verify that System Executables Have Restrictive Permissions
101 echo -e "7. Verify that System Executables Have Restrictive Permissions
\n############################################"
102 find -L /bin -perm /022 -type f -exec chmod go-w {} \;
103 find -L /sbin -perm /022 -type f -exec chmod go-w {} \;
104 find -L /usr/bin -perm /022 -type f -exec chmod go-w {} \;
105 find -L /usr/libexec -perm /022 -type f -exec chmod go-w {} \;
106 find -L /usr/local/bin -perm /022 -type f -exec chmod go-w {} \;
107 find -L /usr/local/sbin -perm /022 -type f -exec chmod go-w {} \;
108 find -L /usr/sbin -perm /022 -type f -exec chmod go-w {} \;
109
110
111 find -L /bin -perm /022 -type d -exec chmod go-w {} \;
112 find -L /sbin -perm /022 -type d -exec chmod go-w {} \;
113 find -L /usr/bin -perm /022 -type d -exec chmod go-w {} \;
114 find -L /usr/libexec -perm /022 -type d -exec chmod go-w {} \;
115 find -L /usr/local/bin -perm /022 -type d -exec chmod go-w {} \;
116 find -L /usr/local/sbin -perm /022 -type d -exec chmod go-w {} \;
117 find -L /usr/sbin -perm /022 -type d -exec chmod go-w {} \;
118
119 echo 'System Executables Have Restrictive Permissions'
120
121 echo -e "############################################\n"
122
123 #Ensure gpgcheck Enabled for All dnf Package Repositories
124 echo -e "8. Ensure gpgcheck Enabled for All dnf Package Repositories
\n############################################"
125 find /etc/yum.repos.d/ -type f -exec sed 's/^gpgcheck=0$//g' {} \;
126 echo 'gpgcheck Enabled for All dnf Package Repositories'
127
128 echo -e "############################################\n"
129
130 #Ensure gpgcheck Enabled In Main dnf Configuration
131 echo -e "9. Ensure gpgcheck Enabled In Main dnf Configuration
\n############################################"
132 sudo sed -i 's/^.*gpgcheck=.*/gpgcheck=1/g' /etc/dnf/dnf.conf
133 echo 'gpgcheck Enabled In Main dnf Configuration'
134
135 echo -e "############################################\n"
136
137 #Disable Prelinking
138 echo -e "10. Disable Prelinking \n############################################"
139 sed -i 's/PRELINKING=yes/PRELINKING=no/' /etc/sysconfig/prelink
140 echo 'Prelinking Disabled'
141
142 echo -e "############################################\n"
143
144 #Build and Test AIDE Database
145 echo -e "11. Build and Test AIDE Database \n############################################"
146 if ! rpm -q --quiet "aide" ; then
147     dnf install -y "aide"
148 fi
149
150 /usr/sbin/aide --init
151 /bin/cp -p /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
152 echo 'AIDE Database is Tested'
153
154 echo -e "############################################\n"
155
156 #Verify and Correct File Permissions with RPM
157 echo -e "12. Verify and Correct File Permissions with RPM \n############################################"
158
159 declare -A NEEDTOCORRECT
160
161
```

```
readarray -t DIFFERENTFROMEXPECTED < <(rpm -Va --nofiledigest | awk '{ if (substr($0,2,1)=="M") print $NF }')

for i in "${DIFFERENTFROMEXPECTED[@]}"
do
    PACKRPM=$(rpm -qf "$i")
    NEEDTOCORRECT["$PACKRPM"]=1
done

for PACKRPM in "${!NEEDTOCORRECT[@]}"
do
    rpm --setperms "${PACKRPM}"
done
echo 'File Permissions with RPM are Corrected'

echo -e "############################################################\n"

#Verify File Hashes with RPM
echo -e "13. Verify File Hashes with RPM \n##############################################"
rpm -Va | grep '^..5'

echo -e '-list which files on the system have hashes that \ndiffer from what is expected by the RPM'
echo -e '-if scan result is pass dont do nothing!! \nelse you need to reinstall packages which package owns the file'

echo -e "############################################################\n"

#Configure SSH to use System Crypto Policy
echo -e "14. Configure SSH to use System Crypto Policy \n##############################################"

sudo sed -i 's/CRYPTO_POLICY=/#CRYPTO_POLICY=/g' /etc/sysconfig/sshd

echo 'Crypto Policies settings are configured correctly'
echo -e "############################################################\n"

#Configure System Cryptography Policy
echo -e "15. Configure System Cryptography Policy\n##############################################"
sudo update-crypto-policies --set DEFAULT

echo -e "############################################################\n"

#Configure Libreswan to use System Crypto Policy
echo -e "16. Configure Libreswan to use System Crypto Policy \n##############################################"
if ! rpm -q --quiet "libreswan" ; then
    dnf install -y "libreswan"
fi

sudo sed -i 's|.*include /etc/crypto-policies/back-ends/libreswan.config*|include /etc/crypto-policies/back-ends/libreswan.config|g' /etc/ipsec.conf
echo 'Libreswan Configured to use System Crypto Policy'

echo -e "############################################################\n"

#Configure Kerberos to use System Crypto Policy
echo -e "17. Configure Kerberos to use System Crypto Policy \n##############################################"
ln -s /etc/crypto-policies/back-ends/krb5.config /etc/krb5.conf/

echo 'Kerberos Configured'

echo -e "############################################################\n"

#Configure OpenSSL library to use System Crypto Policy
echo -e "18. Configure OpenSSL library to use System Crypto Policy \n##############################################"
SECTION='[ crypto_policy ]'
SECTION_REGEX='\[\s*crypto_policy\s*\]'
INCLUSION='.include /etc/crypto-policies/back-ends/openssl.config'
INCLUSION_REGEX='^\s*\.include\s*/etc/crypto-policies/back-ends/openssl.config$'

    Path="/etc/pki/tls/openssl.cnf"
    if test -f "$Path"; then
        if ! grep -q "^\\s*$SECTION_REGEX" "$Path"; then
            printf '\n%s\n\n%s' "$SECTION" "$INCLUSION" >> "$Path"
        elif ! grep -q "^\\s*$INCLUSION_REGEX" "$Path"; then
            sed -i "s|$SECTION_REGEX|&\\n\\n$INCLUSION\\n|" "$Path"
        fi
    else
        echo "'$Path' in not found." >&2
    fi

echo 'OpenSSL library Configured'

echo -e "############################################################\n"

#Ensure PAM Displays Last Logon/Access Notification
```

```
echo -e "19. Ensure PAM Displays Last Logon/Access Notification
\n##############################################"
if $(grep -q "^session.*pam_lastlog.so" /etc/pam.d/postlogin) ; then
    sed -i --follow-symlinks "/pam_lastlog.so/d" /etc/pam.d/postlogin
fi
echo "session     [default=1]   pam_lastlog.so nowtmp showfailed" >> /etc/pam.d/postlogin
echo "session     optional      pam_lastlog.so silent noupdate showfailed" >> /etc/pam.d/postlogin
echo "PAM Displays Last Logon"
echo -e "################################################\n"

#Prevent Login to Accounts With Empty Password
echo -e "20. Prevent Login to Accounts With Empty Password
\n################################################"
sudo sed -i 's/nullok//g' /etc/pam.d/system-auth
echo 'Login Prevented'

echo -e "################################################\n"

#Verify No netrc Files Exist
echo -e "21. Verify No netrc Files Exist \n################################################"
find /root /home/ -type f -name ".netrc" -exec rm -f {} \;

echo 'No netrc Files'

echo -e "################################################\n"

#Verify All Account Password Hashes are Shadowed
echo -e "22. Verify All Account Password Hashes are Shadowed
\n################################################"
variable=$(awk -F: '{if ($2 != "x") print$2}' /etc/passwd)

echo "$variable"

for i in `echo "$variable"` ; do
sed -i "s/$i/x/g" /etc/passwd
done
echo 'Verified All Account Password Hashes and Shadowed'

echo -e "################################################\n"

#All GIDs referenced in /etc/passwd must be defined in /etc/group
echo -e "23. All GIDs referenced in /etc/passwd must be defined in /etc/group
\n################################################"

if   pwck -r | grep 'no group'
then
  echo 'You must define All this GIDs'
else
  echo 'All GIDs are fine'

fi

echo -e "################################################\n"

#Ensure All Accounts on the System Have Unique Names
echo -e "24. Ensure All Accounts on the System Have Unique
Names\n################################################"
variable=$(awk -F: '{count[$1]++; users[$1] = $1 " " users[$1]} END {for (i in count) {if (count[i] > 1)
{ print users[i] } } }' /etc/passwd)
if ! [ -z "$variable" ]
then
  echo 'These names are not Unique change them!!!'
  echo "$variable"
else
  echo 'All Accounts on the System Have Unique Names'

fi
echo -e "################################################\n"

#Verify Only Root Has UID 0
echo -e "25. Verify Only Root Has UID 0 \n################################################"
STATUS=$(cat /etc/passwd | awk -F: '($3 == 0) { print $1 }')
if [ "$STATUS" = 'root' ]; then
    echo 'Only root have UID 0'

else
    echo 'you need to Remove any users other than root with UID 0 or assign them a new UID if appropriate.'
    echo "$STATUS"
fi

echo -e "################################################\n"

#Restrict Serial Port Root Logins
#Direct root Logins Not Allowed
#Restrict Virtual Console Root Logins
echo -e "26. Restrict Serial Port Root Logins \n"
echo -e "27. Direct root Logins Not Allowed \n"
```

```
echo -e "28. Restrict Virtual Console Root Logins \n############################################"
echo > /etc/securetty
echo "Serial Port Root Logins Restricted"
echo "Direct root Logins Not Allowed"
echo "Virtual Console Root Logins Restricted"

echo -e "############################################\n"

#Set Password Warning Age
echo -e "29. Set Password Warning Age \n############################################"
declare Pass_War_Age
Pass_War_Age="7"

grep -q ^PASS_WARN_AGE /etc/login.defs && \
sed -i "s/PASS_WARN_AGE.*/PASS_WARN_AGE\t$Pass_War_Age/g" /etc/login.defs
if ! [ $? -eq 0 ]
then
  echo -e "PASS_WARN_AGE\t$Pass_War_Age" >> /etc/login.defs
fi
echo "Password Warning Age is now good"

echo -e "############################################\n"

#Set Password Minimum Length in login.defs
echo -e "30. Set Password Minimum Length in login.defs \n############################################"

declare Pass_Min_len
Pass_Min_len="12"

grep -q ^PASS_MIN_LEN /etc/login.defs && \
sed -i "s/PASS_MIN_LEN.*/PASS_MIN_LEN\t$Pass_Min_len/g" /etc/login.defs
if ! [ $? -eq 0 ]
then
  echo -e "PASS_MIN_LEN\t$Pass_Min_len" >> /etc/login.defs
fi

echo "Password Minimum Length in login.defs is now good"

echo -e "############################################\n"

#Set Password Minimum Age
echo -e "31. Set Password Minimum Age \n############################################"
declare Pas_Min_Days
Pas_Min_Days="7"

grep -q ^PASS_MIN_DAYS /etc/login.defs && \
sed -i "s/PASS_MIN_DAYS.*/PASS_MIN_DAYS\t$Pas_Min_Days/g" /etc/login.defs
if ! [ $? -eq 0 ]
then
  echo -e "PASS_MIN_DAYS\t$Pas_Min_Days" >> /etc/login.defs
fi

echo "Password Minimum Age is now good"

echo -e "############################################\n"

#Set Password Maximum Age
echo -e "32. Set Password Maximum Age \n############################################"
declare Pass_Max_Days
Pass_Max_Days="90"

grep -q ^PASS_MAX_DAYS /etc/login.defs && \
sed -i "s/PASS_MAX_DAYS.*/PASS_MAX_DAYS\t$Pass_Max_Days/g" /etc/login.defs
if ! [ $? -eq 0 ]
then
  echo -e "PASS_MAX_DAYS\t$Pass_Max_Days" >> /etc/login.defs
fi

echo "Password Maximum Age is now good"

echo -e "############################################\n"

#Ensure that Root's Path Does Not Include World or Group-Writable Directories
echo -e "33. Ensure that Root's Path Does Not Include World or Group-Writable Directories
\n############################################"
find /usr/bin /usr/sbin/ /sbin/ /bin/ /root/  -type d \( -perm -g+w -o -perm -o+w \) -exec chmod g-w {} \;
-exec chmod o-w {} \;

echo "Root's Path Does Not Include World or Group-Writable Directories"

echo -e "############################################\n"

#Enable auditd Service
echo -e "34. Enable auditd Service \n############################################"
serv=auditd

sstat=$(pidof $serv | wc -l )
```

```
409
410 if [ $sstat -gt 0 ]
411
412 then
413
414 echo "$serv is running fine!!!"
415
416 else
417
418 echo "$serv is down/dead"
419
420 service $serv start
421
422 echo "$serv service is UP now!!!"
423
424 fi
425
426 echo -e "################################################\n"
427
428 #Enable Auditing for Processes Which Start Prior to the Audit Daemon
429 echo -e "35. Enable Auditing for Processes Which Start Prior to the Audit Daemon
\n################################################"
430 grub2-editenv - set "$(grub2-editenv - list | grep kernelopts) audit=1"
431
432 echo "Auditing for Processes Which Start Prior to the Audit Daemon Enabled"
433
434 echo -e "################################################\n"
435
436 #Configure auditd Number of Logs Retained
437 echo -e "36. Configure auditd Number of Logs Retained \n################################################"
438 if grep -q num_logs /etc/audit/auditd.conf; then
439    sudo sed -i 's/.*num_logs.*/num_logs = 5/g' /etc/audit/auditd.conf
440 else
441     echo "num_logs = 5" >> /etc/audit/auditd.conf
442 fi
443
444 echo "Auditd Number of Logs Retained Configured"
445
446 echo -e "################################################\n"
447
448 #Configure auditd space_left Action on Low Disk Space
449 echo -e "37. Configure auditd space_left Action on Low Disk Space
\n################################################"
450 if grep -q "^space_left_action" /etc/audit/auditd.conf; then
451    sudo sed -i 's/^space_left_action.*/space_left_action = email/g' /etc/audit/auditd.conf
452 else
453     echo "space_left_action = email" >> /etc/audit/auditd.conf
454 fi
455
456 echo "Auditd space_left Action on Low Disk Space Configured"
457
458 echo -e "################################################\n"
459
460 #Configure auditd max_log_file_action Upon Reaching Maximum Log Size
461 echo -e "38. Configure auditd max_log_file_action Upon Reaching Maximum Log Size
\n################################################"
462
463 if grep -q max_log_file_action /etc/audit/auditd.conf; then
464    sudo sed -i 's/.*max_log_file_action.*/max_log_file_action = rotate/g' /etc/audit/auditd.conf
465 else
466     echo "max_log_file_action = rotate" >> /etc/audit/auditd.conf
467 fi
468
469 echo "Auditd max_log_file_action Upon Reaching Maximum Log Size Configured"
470
471 echo -e "################################################\n"
472
473 # Configure auditd admin_space_left Action on Low Disk Space
474 echo -e "39. Configure auditd admin_space_left Action on Low Disk Space
\n################################################"
475 if grep -q admin_space_left_action /etc/audit/auditd.conf; then
476    sudo sed -i 's/.*admin_space_left_action.*/admin_space_left_action = single/g' /etc/audit/auditd.conf
477 else
478     echo "admin_space_left_action = single" >> /etc/audit/auditd.conf
479 fi
480
481 echo "Auditd admin_space_left Action on Low Disk Space Configured"
482
483 echo -e "################################################\n"
484
485 #Configure auditd to use audispd's syslog plugin
486 echo -e "40. Configure auditd to use audispd's syslog plugin
\n################################################"
487 if grep -q active /etc/audit/plugins.d/syslog.conf; then
488    sudo sed -i 's/.*active.*/active = yes/g' /etc/audit/plugins.d/syslog.conf
489 else
490     echo "active = yes" >> /etc/audit/plugins.d/syslog.conf
```

```
fi

sudo service auditd restart

echo "Auditd to use audispd's syslog plugin Configured"

echo -e "################################################\n"

# Configure auditd Max Log File Size
echo -e "41. Configure auditd Max Log File Size  \n##############################################"
if grep -q "\bmax_log_file\b" /etc/audit/auditd.conf; then
   sudo sed -i 's/\b.*max_log_file\b.*/max_log_file = 6/g' /etc/audit/auditd.conf
else
    echo "max_log_file = 6" >> /etc/audit/auditd.conf
fi

echo "Auditd Max Log File Size Configured"

echo -e "################################################\n"

#Configure auditd mail_acct Action on Low Disk Space
echo -e "42. Configure auditd mail_acct Action on Low Disk Space
\n##############################################"
if grep -q action_mail_acct /etc/audit/auditd.conf; then
   sudo sed -i 's/.*action_mail_acct.*/action_mail_acct = root/g' /etc/audit/auditd.conf
else
    echo "action_mail_acct = root" >> /etc/audit/auditd.conf
fi

echo "Auditd Number of Logs Retained Configured"

echo -e "################################################\n"

#Ensure auditd Collects System Administrator Actions
echo -e "43. Ensure auditd Collects System Administrator Actions
\n##############################################"
Files=$(find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/sudoers.d/ -p wa -k
actions"  {} \;)
Files1=$(find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/sudoers -p wa -k actions"
{} \;)
Files2=$(find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/sudoers.d/ -p wa -
k actions"  {} \;)
Files3=$(find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/sudoers -p wa -k
actions"  {} \;)

for i in $Files;
do
  echo "-w /etc/sudoers.d/ -p wa -k actions" >> $i;
done

for i in $Files1;
do
  echo "-w /etc/sudoers -p wa -k actions" >> $i;
done

for i in $Files2;
do
  echo "-w /etc/sudoers.d/ -p wa -k actions" >> $i;
done

for i in $Files3;
do
  echo "-w /etc/sudoers -p wa -k actions" >> $i;
done

echo "Auditd Collects System Administrator Actions"

echo -e "################################################\n"

#Record Events that Modify User/Group Information
echo -e "44. Record Events that Modify User/Group Information
\n##############################################"
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/group -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/group -p wa -k
audit_rules_usergroup_modification" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/passwd -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/passwd -p wa -k
audit_rules_usergroup_modification" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/gshadow -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/gshadow -p wa -k
audit_rules_usergroup_modification" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/shadow -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/shadow -p wa -k
audit_rules_usergroup_modification" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/security/opasswd -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/security/opasswd -p wa -k
audit_rules_usergroup_modification" >> $line; done
```

```
561
562 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/group -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/group -p wa -k
audit_rules_usergroup_modification" >> $line; done
563 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/passwd -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/passwd -p wa -k
audit_rules_usergroup_modification" >> $line; done
564 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/gshadow -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/gshadow -p wa -k
audit_rules_usergroup_modification" >> $line; done
565 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/shadow -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/shadow -p wa -k
audit_rules_usergroup_modification" >> $line; done
566 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/security/opasswd -p wa -k
audit_rules_usergroup_modification"  {} \; | while read line; do echo "-w /etc/security/opasswd -p wa -k
audit_rules_usergroup_modification" >> $line; done
567
568 echo "Events that Modify User/Group Information Recorded"
569
570 echo -e "################################################\n"
571
572 #System Audit Logs Must Be Owned By Root
573 echo -e "45. System Audit Logs Must Be Owned By Root  \n################################################"
574 sudo chown root /var/log/audit
575 sudo chown root /var/log/audit/*
576
577 echo "System Audit Logs are Owned By Root"
578
579 echo -e "################################################\n"
580
581 #Ensure auditd Collects Information on Exporting to Media (successful)
582 echo -e "46. Ensure auditd Collects Information on Exporting to Media (successful)
\n################################################"
583 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S mount
-F auid>=1000 -F auid!=unset -F key=export"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
mount -F auid>=1000 -F auid!=unset -F key=export" >> $line; done
584 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S mount
-F auid>=1000 -F auid!=unset -F key=export"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
mount -F auid>=1000 -F auid!=unset -F key=export" >> $line; done
585
586 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S mount -F
auid>=1000 -F auid!=unset -F key=export"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S mount
-F auid>=1000 -F auid!=unset -F key=export" >> $line; done
587 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S mount -F
auid>=1000 -F auid!=unset -F key=export"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S mount
-F auid>=1000 -F auid!=unset -F key=export" >> $line; done
588
589 echo "Auditd Collects Information on Exporting to Media (successful)"
590
591 echo -e "################################################\n"
592
593 #Record Events that Modify the System's Mandatory Access Controls
594 echo -e "47. Record Events that Modify the System's Mandatory Access Controls
\n################################################"
595 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/selinux/ -p wa -k MAC-
policy"  {} \; | while read line; do echo "-w /etc/selinux/ -p wa -k MAC-policy" >> $line; done
596
597 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/selinux/ -p wa -k MAC-policy"  {}
\; | while read line; do echo "-w /etc/selinux/ -p wa -k MAC-policy" >> $line; done
598
599 echo "Events that Modify the System's Mandatory Access Controls Recorded"
600
601 echo -e "################################################\n"
602
603 #Record Attempts to Alter Process and Session Initiation Information
604 echo -e "48. Record Attempts to Alter Process and Session Initiation Information
\n################################################"
605 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /var/run/utmp -p wa -k session"
{} \; | while read line; do echo "-w /var/run/utmp -p wa -k session" >> $line; done
606 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /var/log/btmp -p wa -k session"
{} \; | while read line; do echo "-w /var/log/btmp -p wa -k session" >> $line; done
607 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /var/log/wtmp -p wa -k session"
{} \; | while read line; do echo "-w /var/log/wtmp -p wa -k session" >> $line; done
608
609 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /var/run/utmp -p wa -k session"  {} \;
| while read line; do echo "-w /var/run/utmp -p wa -k session" >> $line; done
610 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /var/log/btmp -p wa -k session"  {} \;
| while read line; do echo "-w /var/log/btmp -p wa -k session" >> $line; done
611 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /var/log/wtmp -p wa -k session"  {} \;
| while read line; do echo "-w /var/log/wtmp -p wa -k session" >> $line; done
612
613 echo "Attempts to Alter Process and Session Initiation Informatin Recorded"
614
615 echo -e "################################################\n"
616
617 #Record Events that Modify the System's Network Environment
```

```
618 echo -e "49. Record Events that Modify the System's Network Environment
\n#############################################"
619 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/sysconfig/network -p wa -k
audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-w /etc/sysconfig/network -p wa -k
audit_rules_networkconfig_modification" >> $line; done
620 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/hosts -p wa -k
audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-w /etc/hosts -p wa -k
audit_rules_networkconfig_modification" >> $line; done
621 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/issue.net -p wa -k
audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-w /etc/issue.net -p wa -k
audit_rules_networkconfig_modification" >> $line; done
622 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/issue -p wa -k
audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-w /etc/issue -p wa -k
audit_rules_networkconfig_modification" >> $line; done
623 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
sethostname,setdomainname -F key=audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-a
always,exit -F arch=b64 -S sethostname,setdomainname -F key=audit_rules_networkconfig_modification" >> $line;
done
624 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
sethostname,setdomainname -F key=audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-a
always,exit -F arch=b32 -S sethostname,setdomainname -F key=audit_rules_networkconfig_modification" >> $line;
done
625
626 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/sysconfig/network -p wa -k
audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-w /etc/sysconfig/network -p wa -k
audit_rules_networkconfig_modification" >> $line; done
627 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/hosts -p wa -k
audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-w /etc/hosts -p wa -k
audit_rules_networkconfig_modification" >> $line; done
628 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/issue.net -p wa -k
audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-w /etc/issue.net -p wa -k
audit_rules_networkconfig_modification" >> $line; done
629 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/issue -p wa -k
audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-w /etc/issue -p wa -k
audit_rules_networkconfig_modification" >> $line; done
630 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
sethostname,setdomainname -F key=audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-a
always,exit -F arch=b64 -S sethostname,setdomainname -F key=audit_rules_networkconfig_modification" >> $line;
done
631 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
sethostname,setdomainname -F key=audit_rules_networkconfig_modification"  {} \; | while read line; do echo "-a
always,exit -F arch=b32 -S sethostname,setdomainname -F key=audit_rules_networkconfig_modification" >> $line;
done
632
633 echo "Events that Modify the System's Network Environment Recorded"
634
635 echo -e "#################################################\n"
636
637
638 #Make the auditd Configuration Immutable
639 echo -e "50. Make the auditd Configuration Immutable \n#############################################"
640 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-e 2"  {} \; | while read line; do
echo "-e 2" >> $line; done
641
642 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-e 2"  {} \; | while read line; do echo "-
e 2" >> $line; done
643
644 echo "Auditd Configuration Immutable"
645
646 echo -e "#################################################\n"
647
648 #Ensure auditd Collects Information on Kernel Module Loading and Unloading
649 echo -e "51. Ensure auditd Collects Information on Kernel Module Loading and Unloading
\n#############################################"
650 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
init_module,finit_module,delete_module -F key=modules"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S init_module,finit_module,delete_module -F key=modules" >> $line; done
651 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
init_module,finit_module,delete_module -F key=modules"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S init_module,finit_module,delete_module -F key=modules" >> $line; done
652
653 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
init_module,finit_module,delete_module -F key=modules"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S init_module,finit_module,delete_module -F key=modules" >> $line; done
654 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
init_module,finit_module,delete_module -F key=modules"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S init_module,finit_module,delete_module -F key=modules" >> $line; done
655
656 echo "Auditd Collects Information on Kernel Module Loading and Unloading"
657
658 echo -e "#################################################\n"
659
660 #Ensure auditd Collects File Deletion Events by User
661 echo -e "52. Ensure auditd Collects File Deletion Events by User
\n#############################################"
662 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F key=delete"  {} \; | while read line; do
```

```
echo "-a always,exit -F arch=b32 -S rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F
key=delete" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F key=delete"  {} \; | while read line; do
echo "-a always,exit -F arch=b64 -S rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F
key=delete" >> $line; done

find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F key=delete"  {} \; | while read line; do
echo "-a always,exit -F arch=b32 -S rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F
key=delete" >> $line; done
find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F key=delete"  {} \; | while read line; do
echo "-a always,exit -F arch=b64 -S rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F
key=delete" >> $line; done

echo "Auditd Collects File Deletion Events by User"

echo -e "##############################################\n"

#Ensure auditd Collects Information on the Use of Privileged Commands
echo -e "53. Ensure auditd Collects Information on the Use of Privileged Commands
\n##############################################"
file=$(sudo find / -xdev -type f -perm -4000 -o -type f -perm -2000 2>/dev/null)

for i in $file
do
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F path=$i -F perm=x
-F auid>=1000 -F auid!=unset -k privileged"  {} \; | while read line; do echo "-a always,exit -F path=$i -F
perm=x -F auid>=1000 -F auid!=unset -k privileged" >> $line; done

find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F path=$i -F perm=x -F
auid>=1000 -F auid!=unset -k privileged"  {} \; | while read line; do echo "-a always,exit -F path=$i -F perm=x
-F auid>=1000 -F auid!=unset -k privileged" >> $line; done
done

echo "Auditd Collects Information on the Use of Privileged Commands"

echo -e "##############################################\n"

#Record Attempts to Alter Logon and Logout Events
echo -e "54. Record Attempts to Alter Logon and Logout Events
\n##############################################"
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /var/log/tallylog -p wa -k
logins"  {} \; | while read line; do echo "-w /var/log/tallylog -p wa -k logins" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /var/run/faillock -p wa -k
logins"  {} \; | while read line; do echo "-w /var/run/faillock -p wa -k logins" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /var/log/lastlog -p wa -k
logins"  {} \; | while read line; do echo "-w /var/log/lastlog -p wa -k logins" >> $line; done

find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /var/log/tallylog -p wa -k logins"  {}
\; | while read line; do echo "-w /var/log/tallylog -p wa -k logins" >> $line; done
find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /var/run/faillock -p wa -k logins"  {}
\; | while read line; do echo "-w /var/run/faillock -p wa -k logins" >> $line; done
find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /var/log/lastlog -p wa -k logins"  {}
\; | while read line; do echo "-w /var/log/lastlog -p wa -k logins" >> $line; done

echo "Attempts to Alter Logon and Logout Events Recorded"

echo -e "##############################################\n"

#Record Events that Modify the System's Discretionary Access Controls - fsetxattr
echo -e "55. Record Events that Modify the System's Discretionary Access Controls - fsetxattr
\n##############################################"
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S fsetxattr -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S fsetxattr -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

echo "Events that Modify the System's Discretionary Access Controls - fsetxattr Recorded"

echo -e "##############################################\n"

#Record Events that Modify the System's Discretionary Access Controls - lremovexattr
echo -e "56. Record Events that Modify the System's Discretionary Access Controls - lremovexattr
\n##############################################"
```

```
715 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
716 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
717
718 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S lremovexattr
-F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
719 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S lremovexattr
-F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
720
721 echo "Events that Modify the System's Discretionary Access Controls - lremovexattr Recorded"
722
723 echo -e "##############################################\n"
724
725 #Record Events that Modify the System's Discretionary Access Controls - fremovexattr
726 echo -e "57. Record Events that Modify the System's Discretionary Access Controls - fremovexattr
\n##############################################"
727 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
728 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
729
730 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S fremovexattr
-F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
731 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S fremovexattr
-F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
732
733 echo "Events that Modify the System's Discretionary Access Controls - fremovexattr Recorded"
734
735 echo -e "##############################################\n"
736
737 #Record Events that Modify the System's Discretionary Access Controls - chmod
738 echo -e "58. Record Events that Modify the System's Discretionary Access Controls - chmod
\n##############################################"
739 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S chmod
-F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
chmod -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
740 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S chmod
-F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
chmod -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
741
742 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S chmod -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
chmod -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
743 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S chmod -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
chmod -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
744
745 echo "Events that Modify the System's Discretionary Access Controls - chmod Recorded"
746
747 echo -e "##############################################\n"
748
749
750 #Record Events that Modify the System's Discretionary Access Controls - lchown
751 echo -e "59. Record Events that Modify the System's Discretionary Access Controls - lchown
\n##############################################"
752 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
lchown -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S lchown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
753 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
lchown -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S lchown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
754
755 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S lchown -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
lchown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
756 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S lchown -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
lchown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
757
758 echo "Events that Modify the System's Discretionary Access Controls - lchown Recorded"
759
760 echo -e "##############################################\n"
761
762 #Record Events that Modify the System's Discretionary Access Controls - lsetxattr
763 echo -e "60. Record Events that Modify the System's Discretionary Access Controls - lsetxattr
\n##############################################"
```

```
764 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
765 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
766
767 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S lsetxattr -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
768 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S lsetxattr -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
769
770 echo "Events that Modify the System's Discretionary Access Controls - lsetxattr Recorded"
771
772 echo -e "##############################################\n"
773
774 #Record Events that Modify the System's Discretionary Access Controls - fchownat
775 echo -e "61. Record Events that Modify the System's Discretionary Access Controls - fchownat
\n##############################################"
776 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
777 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
778
779 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S fchownat -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
780 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S fchownat -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
781
782 echo "Events that Modify the System's Discretionary Access Controls - fchownat Recorded"
783
784 echo -e "##############################################\n"
785
786 #Record Events that Modify the System's Discretionary Access Controls - fchown
787 echo -e "62. Record Events that Modify the System's Discretionary Access Controls - fchown
\n##############################################"
788 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
fchown -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S fchown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
789 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
fchown -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S fchown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
790
791 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S fchown -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
fchown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
792 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S fchown -F
auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
fchown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
793
794 echo "Events that Modify the System's Discretionary Access Controls - fchown Recorded"
795
796 echo -e "##############################################\n"
797
798
799 #Record Events that Modify the System's Discretionary Access Controls - removexattr
800 echo -e "63. Record Events that Modify the System's Discretionary Access Controls - removexattr
\n##############################################"
801
802 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b32 -S removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
803 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F
arch=b64 -S removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
804
805 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S removexattr
-F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
806 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S removexattr
-F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
807
808 echo "Events that Modify the System's Discretionary Access Controls - removexattr Recorded"
809
810 echo -e "##############################################\n"
811
812 #Record Events that Modify the System's Discretionary Access Controls - chown
813 echo -e "64. Record Events that Modify the System's Discretionary Access Controls - chown
\n##############################################"
814
```

```
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

echo "Events that Modify the System's Discretionary Access Controls - chown Recorded"

echo -e "##############################################\n"

#Record Events that Modify the System's Discretionary Access Controls - fchmod
echo -e "65. Record Events that Modify the System's Discretionary Access Controls - fchmod \n##############################################"
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

echo "Events that Modify the System's Discretionary Access Controls - fchmod Recorded"

echo -e "##############################################\n"

#Record Events that Modify the System's Discretionary Access Controls - setxattr
echo -e "66. Record Events that Modify the System's Discretionary Access Controls - setxattr \n##############################################"

find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

echo "Events that Modify the System's Discretionary Access Controls - setxattr Recorded"

echo -e "##############################################\n"

#Record Events that Modify the System's Discretionary Access Controls - fchmodat
echo -e "67. Record Events that Modify the System's Discretionary Access Controls - fchmodat \n##############################################"
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done
find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod" >> $line; done

echo "Events that Modify the System's Discretionary Access Controls - fchmodat Recorded"

echo -e "##############################################\n"

# Record Attempts to Alter the localtime File
echo -e "68. Record Attempts to Alter the localtime File \n##############################################"
find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-w /etc/localtime -p wa -k audit_time_rules"  {} \; | while read line; do echo "-w /etc/localtime -p wa -k audit_time_rules" >> $line; done
```

```
865
866 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-w /etc/localtime -p wa -k
audit_time_rules"  {} \; | while read line; do echo "-w /etc/localtime -p wa -k audit_time_rules" >> $line;
done
867
868 echo "Attempts to Alter the localtime File Recorded"
869
870 echo -e "################################################\n"
871
872 #Record attempts to alter time through adjtimex
873 echo -e "69. Record attempts to alter time through adjtimex
\n################################################"
874 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
adjtimex -F key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S adjtimex -F
key=audit_time_rules" >> $line; done
875 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
adjtimex -F key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S adjtimex -F
key=audit_time_rules" >> $line; done
876
877 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S adjtimex -F
key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S adjtimex -F
key=audit_time_rules" >> $line; done
878 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S adjtimex -F
key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S adjtimex -F
key=audit_time_rules" >> $line; done
879
880 echo "Attempts to alter time through adjtimex Recorded"
881
882 echo -e "################################################\n"
883
884 # Record attempts to alter time through settimeofday
885 echo -e "70. Record attempts to alter time through settimeofday
\n################################################"
886 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
settimeofday -F key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
settimeofday -F key=audit_time_rules" >> $line; done
887 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
settimeofday -F key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
settimeofday -F key=audit_time_rules" >> $line; done
888
889 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S settimeofday
-F key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S settimeofday -F
key=audit_time_rules" >> $line; done
890 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S settimeofday
-F key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S settimeofday -F
key=audit_time_rules" >> $line; done
891
892 echo "Attempts to alter time through settimeofday Recorded"
893
894 echo -e "################################################\n"
895
896 #Record Attempts to Alter Time Through stime
897 echo -e "71. Record attempts to alter time through stime \n################################################"
898 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S stime
-F key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S stime -F
key=audit_time_rules" >> $line; done
899
900 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S stime -F
key=audit_time_rules"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S stime -F
key=audit_time_rules" >> $line; done
901
902 echo "Attempts to alter time through stime Recorded"
903
904 echo -e "################################################\n"
905
906 #Record Attempts to Alter Time Through clock_settime
907 echo -e "72. Record attempts to alter time through clock_settime
\n################################################"
908 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
clock_settime -F a0=0x0 -F key=time-change"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
clock_settime -F a0=0x0 -F key=time-change" >> $line; done
909 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
clock_settime -F a0=0x0 -F key=time-change"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
clock_settime -F a0=0x0 -F key=time-change" >> $line; done
910
911 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
clock_settime -F a0=0x0 -F key=time-change"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
clock_settime -F a0=0x0 -F key=time-change" >> $line; done
912 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
clock_settime -F a0=0x0 -F key=time-change"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
clock_settime -F a0=0x0 -F key=time-change" >> $line; done
913
914 echo "Attempts to alter time through clock_settime Recorded"
915
916 echo -e "################################################\n"
917
918 #Ensure auditd Collects Unauthorized Access Attempts to Files (unsuccessful)
```

```
919 echo -e "73. Ensure auditd Collects Unauthorized Access Attempts to Files (unsuccessful)
\n##############################################"
920
921 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access" >> $line; done
922 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F
key=access"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F
key=access" >> $line; done
923 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access" >> $line; done
924 find /etc/audit -maxdepth 1 -type f -name "audit.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F
key=access"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F
key=access" >> $line; done
925
926 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access" >> $line; done
927 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b32 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F
key=access"  {} \; | while read line; do echo "-a always,exit -F arch=b32 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F
key=access" >> $line; done
928 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access" >> $line; done
929 find /etc/audit/rules.d/ -type f -name "*.rules" -exec grep -Le "-a always,exit -F arch=b64 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F
key=access"  {} \; | while read line; do echo "-a always,exit -F arch=b64 -S
creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F
key=access" >> $line; done
930
931 echo "Auditd Collects Unauthorized Access Attempts to Files (unsuccessful)"
932
933 echo -e "##############################################\n"
934
935 # Enable the NTP Daemon
936 echo -e "74. Enable the NTP Daemon \n##############################################"
937 serv=chronyd
938
939 sstat=$(pidof $serv | wc -l )
940
941 if [ $sstat -gt 0 ]
942
943 then
944
945 echo "$serv is running fine!!!"
946 echo "NTP Daemon is enabled"
947
948 else
949
950 echo "$serv is down/dead"
951
952 service $serv start
953
954 echo "$serv service is UP now!!!"
955 echo "NTP Daemon is enabled"
956
957 fi
958 echo -e "##############################################\n"
959
960
961 #Specify a Remote NTP Server
962 echo -e "75. Specify a Remote NTP Server \n##############################################"
963 var_multiple_time_servers="0.rhel.pool.ntp.org,1.rhel.pool.ntp.org,2.rhel.pool.ntp.org,3.rhel.pool.ntp.org"
964 if ! `grep -q ^server /etc/chrony.conf` ; then
965     if ! `grep -q '#[[:space:]]*server' /etc/chrony.conf` ; then
966       for i in `echo "$var_multiple_time_servers" | tr ',' '\n'` ; do
967         echo -ne "\nserver $i iburst" >> /etc/chrony.conf
968       done
969     else
970       sed -i 's/#[ ]*server/server/g' /etc/chrony.conf
971     fi
972   fi
```

```
973
974   echo 'Remote NTP is specified'
975   echo -e "#############################################\n"
976
977 #Set SSH Idle Timeout Interval
978 echo -e "76. Set SSH Idle Timeout Interval \n#############################################"
979 sudo sed -i 's/.*ClientAliveInterval.*/ClientAliveInterval 300/g' /etc/ssh/sshd_config
980 echo 'ClientAliveInterval is changed to 300 (5min)'
981 echo -e "#############################################\n"
982
983 #Disable SSH Root Login
984 echo -e "77. Disable SSH Root Login \n#############################################"
985 sed -i '0,/PermitRootLogin/{s/.*PermitRootLogin.*/PermitRootLogin no/}' /etc/ssh/sshd_config
986 echo 'PermitRootLogin is changed to no'
987 echo -e "#############################################\n"
988
989
990
991 #Set SSH Client Alive Max Count
992 echo -e "78. Set SSH Client Alive Max Count \n#############################################"
993
994 sudo sed -i 's/.*ClientAliveCountMax.*/ClientAliveCountMax 0/g' /etc/ssh/sshd_config
995
996 echo 'ClientAliveCountMax is changed to 0'
997 echo -e "#############################################\n"
998
999
1000#Disable SSH Access via Empty Passwords
1001echo -e "79. Disable SSH Access via Empty Passwords \n#############################################"
1002
1003sudo sed -i 's/.*PermitEmptyPasswords.*/PermitEmptyPasswords no/g' /etc/ssh/sshd_config
1004
1005echo 'PermitEmptyPasswords is changed to no'
1006echo -e "#############################################\n"
```