

Октобар 2014. Број 29

ЛИБРЕ!

Часопис о слободном софтверу



OWASP

The Open Web Application Security Project

SeraphimDROID



Login



ЈОШ ИЗДВАЈАМО

**0 hackerspace-y
Ubuntu Mate 14.10**



Creative Commons Ауторство-Некомерцијално-Делити под истим условима

Реч уредника

Општи конкурс

Очигледно је да *ValCCop 2k14* није вратио довољно енергије у ЛиБРЕ! пројекат - приватни послови су још увек јачи. „Стари“ аутори су се дефинитивно уморили и исцрпели су теме за писање. Осипање редакције у волонтерском пројекту какав је ЛиБРЕ!, нормална је и природна ствар. Зато је за опстанак оваквог пројекта неопходно стално освежавање редакције новим сарадницима са новом енергијом, новим темама и идејама.

Можда смо мало затечени ситуацијом јер нам се одједном редакција преполовила. За такву ситуацију нисмо били припремљени. Морамо опет да поновимо да је ово сложен пројекат који не може да опстане без десет активних аутора, три лектора и два графичара. То нас укупно доводи до цифре од петнаест активних чланова редакције. Испод те цифре посао на редовном излажењу часописа постаје претежак и осећа се пад у квалитету и у редовности изласка часописа.

Док смо се ми мучили да скупимо чланке за овај број, добили смо неколико порука подршке. Основна порука те подршке могла би се сажети у реченицу: „Ма колико вам понекад било тешко, а знамо да јесте, немојте да дозволите да наш часопис буде угашен!“ Читаоци осећају да имамо проблема, али не желе да се пројекат угаси. Хвала вам на тој подршци!

Једини излаз из овог стања у пројекту је долазак нових сарадника. Првенствено нам је потребно што више нових аутора - конкретно, аутора који ће скупљати актуелне *FLOSS* вести, аутора који ће тестирати нове верзије *FLOSS OS*-а (*GNU/Linux*, *BSD*, *Solaris*) и писати о њима, аутора који ће тестирати нове *GNU* софтвере и писати о њима, затим аутора који знају нешто о новим технологијама заснованим на *FLOSS* решењима, аутора који *FLOSS* користе у пословне сврхе и који желе то искуство да поделе са другима, аутора туторијала *FLOSS* софтвера, аутора који познају *AndroidOS*, *Android FLOSS* апликације као и друге *FLOSS* мобилне оперативне системе, аутора који познају *open hardware* итд.

Такође нам је потребан један *web* администратор који ће побољшати и одржавати нашу *web* инфраструктуру. Он би и једном месечно помогао око објаве



часописа.

Без лектора рад аутора не би био потпун. Аутори можда не морају да знају сва правописна правила. За њих је битније да имају потребно знање о теми о којој пишу. Исправку правописних грешака треба препустити онима који то боље знају - лекторима. Зато је конкурс за ЛиБРЕ! лектора константно отворен. Ово је прилика да студенти филологије, свршени и несвршени, стекну ново искуство и добију прву референцу за будућу каријеру.

Рачунамо на то да је лакше попунити редакцију новим људима, него поново покретати часопис од почетка. Реновирање смо започели новим изгледом овог броја часописа. Обнављање редакције омогућиће да се следећи број са садржајем бољег квалитета објави на време. Нови сарадници треба што пре да нам се јаве на нашу већ познату адресу електронске поште - [libre \[et\] lugons \[dot\] org](mailto:libre@lugons.org). Ми ћемо свим заинтересованима изаћи у сусрет, објаснити им како и шта се очекује од нових сарадника. Искрено се надамо да ће ова акција успети и да ће обезбедити сигурнију будућност ЛиБРЕ! пројекта.

До читања,

ЛиБРЕ! тим

Број: 29

Периодика излажења: месечник

Извршни уредник: Стефан Ножинић

Главни лектор:

Александар Божиновић

Лектура:

Јелена Муњћан

Милена Беран

Александра Ристовић

Графичка обрада:

Дејан Маглов

Иван Радељић

Дизајн: White Circle Creative Team

Аутори у овом броју:

Вељко Симић

Дејан Маглов

Петар Симић

Александар Божиновић

Гаврило Продановић

Стефан Ножинић

Остали сарадници у овом броју:

Марко Новаковић

Михајло Богдановић

Почасни чланови редакције:

Жељко Попивоца

Жељко Шарић

Александар Станисављевић

Никола Харди

Владимир Попадић

Контакт:

IRC: #floss-magazin на irc.freenode.net

Е-пошта: libre@lugons.org

Садржај

Вести

стр. 6

Пул слободе

О hackerspace-у

стр. 10

Представљамо

Ubuntu Mate 14.10

стр. 15

Како да...?

libGDX: „Java game development framework“ (5. део)

Увод у програмски језик C (6. део)

стр. 21

стр. 26

Ослобађање

У потрази за идеалном дистрибуцијом:

Још критеријума за одабир идеалне дистрибуције

стр. 29

Интернет, мреже комуникације

Енкриптована електронска пошта (4. део)

Shellshock

стр. 36

стр. 43

Mobilni kutak

OWASP SeraphimDROID

Интервју са Николом Милошевићем -

ментором пројекта *OWASP Seraphimdroid*

стр. 45

Моћ слободног
софтвера





GRAPHIC DESIGN & WEB DEVELOPMENT



УДРУЖЕЊЕ ЛИНУКС КОРИСНИКА НОВОГ САДА



REGIONALNI
LINUX PORTAL

linuxzasve.com

LUTHERUS

Et in Arcadia ego!



ictcasopis.ict.edu.rs



Grupa korisnika GNU/Linux operativnih sistema u Lovcencu

info i tutorijali na srpskom
lubunturs.wordpress.com



Вести

17. септембар 2014.

Учење *Linux*-а преко видео серијала

Браћа *Jared* и *Neilsen* покренули су пројекат снимања видео материјала који треба да научи младе рачунарским наукама.

Користан линк: <http://j.mp/1x7FLmk>



20. септембар 2014.

Такмичење „Креирање наставних садржаја коришћењем слободног софтвера”

Удружење професора информатике и удружење грађана „*Linux centar*” са финансијском подршком Министарства спољне и унутрашње трговине и телекомуникација Републике Србије реализују пројекат под називом „Креирање наставних садржаја коришћењем слободног софтвера”.

Користан линк: <http://j.mp/1skxTHB>



24. септембар 2014.

GNOME 3.14

Стигла је нова верзија овог окружења радне површи са новим могућностима и ажурираним апликацијама.

Користан линк: <http://j.mp/1zySgZc>

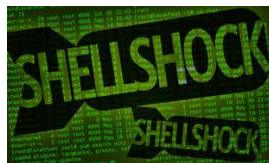




24. септембар 2014.

Сигурносни пропуст у *Bash* командном интерпретеру

Откривен је сигурносни пропуст у *Bash* командном интерпретеру који може да угрози огроман број *Linux* сервера али исто тако и доста рачунара са *OS X* оперативним системом компаније *Apple*.



Користан линк: <http://j.mp/1EmxQ5r>

25. септембар 2014.

LibreOffice 4.3.2

Стигла је нова верзија овог канцеларијског пакета са преко осамдесет исправки грешака.

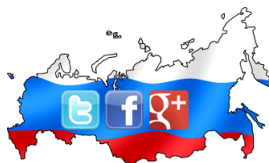


Користан линк: <http://j.mp/10B6Xfd>

26. септембар 2014.

Русија захтева регистрацију друштвених мрежа

Русија ће захтевати да *Facebook*, *Google* и *Twitter* своје платформе региструју као друштвене мреже иначе ће бити блокирани.



Користан линк: <http://j.mp/1opkiSX>

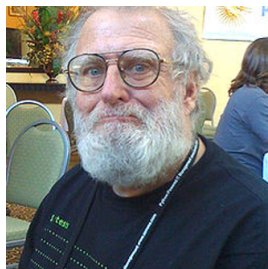
Вести

29. септембар 2014.

Финансирање опоравка *John-a Draper-a*

Покренута је акција за финансирање лечења овог хакера који је познат по томе што је први ухаковао америчку телефонску мрежу и тако је, едукујући *Steve-a Wozniak-a* о томе, омогућио њему и *Steve-y Jobs-y* да оснују *Apple*.

Користан линк: <http://j.mp/1tNdTlo>

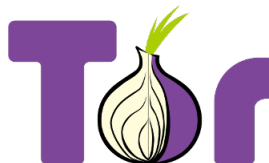


29. септембар 2014.

Tor за широке народне масе

Пројекат *Tor* разматра интеграцију у *Firefox* претраживач и да тиме омогући лакше коришћење *Tor-a* за веће народне масе.

Користан линк: <http://j.mp/1GuG0Lc>



30. септембар 2014.

Linux фондација планира пројекат који ће омогућити убрзано представљање нових производа у индустрији

Овај пројекат су одржале многе велике фирме у смислу финансијске и друге помоћи. Неке од тих фирми су *AT&T*, *HP*, *Dell*, *Cisco*, *Intel* и многе друге.



Користан линк: <http://j.mp/1skyngT>



14. октобар 2014.

Firefox 33 доноси нове могућности

Последња стабилна верзија овог популарног претраживача интернета доноси предлагање претраге (енг. *Search suggestion*) и подршку за *OpenH264*.



Користан линк: <http://goo.gl/63WIQo>

14. октобар 2014.

Red Hat најавио нову верзију RHEL 6.6

Red Hat Enterprise Linux (RHEL) 6.6 доноси дијапазон побољшања који повећавају системске перформансе на физичком, виртуелном и *cloud* окружењу.



Користан линк: <http://goo.gl/dK9MRZ>

23. октобар 2014.

Ubuntu слави десет година постојања

Објављена је верзија 14.10 под кодним именом *Utopic Unicorn*. Нова верзија обележава десетогодишњицу постојања ове дистрибуције.

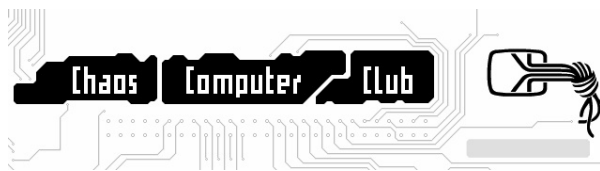


Користан линк: <http://goo.gl/631Gjw>

Пул слободе

О hackerspace-у

Аутор: Стефан Ножинић





Шта је *hackerspace*?

Hackerspace је место где се различити људи сличних интересовања могу окупити, дружити и размењивати искуства и знања која се тичу технологије, науке, уметности и осталих области. *Hackerspace* је често руковођен заједницама и непрофитним организацијама, мада се у последње време могу приметити и комерцијални *hackerspace*-ови као и они унутар великих фирми које се баве технологијом. Одсуство великих *hackerspace*-ова се не може приметити ни код универзитета, а уз њих често иду и најразличитије лабораторије. Није редак случај да се у овим просторијама могу пронаћи неки од најгенијалнијих људи данашњице као што су *Richard Stallman* и *Mitch Altman*. *Hackerspace* се најчешће финансира од стране сопствених припадника кроз чланарину. Ако је *hackerspace* организован на универзитету, финансирање је могуће и из буџета универзитета. Није редак случај да се финансирање обавља кроз сарадњу са фирмама и кроз донације и спонзорства.

Како направити један?

Eric Michaud, оснивач *HacDC hackerspace*-а, написао је серијал од седам делова о томе како направити сопствени *hackerspace*. Ми вам преносимо тај серијал укратко:



Пре свега, важно је нагласити да су оваква места распоређена свуда по свету и да је данас много лакше добити помоћ и подршку, него што је то било могуће пре двадесет година када је интернет био мање распрострањен и када ова идеологија није била толико активна. Како *Eric* предлаже, прво што је потребно приликом оснивања *hackerspace*-а, јесте одлука о томе коме је иницијално намењен тај *hackerspace*. Ово наизглед може представљати проблем оснивачима који нису сигурни коме је намењен и кога очекују, односно, не могу унапред да предвиде дешавања. Треба напоменути да то није толико битно, касније се заједница може ширити, али је битно основати иницијални тим окупљен око једне идеје, а касније се он може, по потреби, проширити. Оваква одлука даје могућности да се одреде алати и сам потребан материјал у почетку. На пример, ако је ваш *hackerspace* намењен развијању вештачке интелигенције, потребно је обезбедити рачунаре и интернет, али није толико неопходно обезбедити неке друге алате који се не употребљавају у тој области превише.

Пулс слободе

Следеће што је потребно, јесте, наравно, простор. Простор може да зависи од тога чиме ће се ваш *hackerspace* бавити, али углавном је потребно да има основна средства за почетак као што су напајање 220V AC, добро осветљена просторија тако да може да улази природно светло, тамна просторија (или могућност да се направи иста), вода и приступ сваког дана у недељи. Исто тако, пожељно је направити листу потребних алата, опреме и свега што ће бити неопходно. Не треба предвиђати будућност и узети све у обзир са идеологијом „можда затреба касније”, већ је потребно обезбедити основне алате и опрему, а касније их надоградити по потреби и према финансијским могућностима. Сам простор може бити било шта и било где, тако да испуњава ваше услове. Простор могу обезбедити, на пример, школа и универзитет, али можете и своју гаражу учинити *hackerspace*-ом.





Како је локација битна из више разлога, потребно је за сваку могућу селекцију видети предности и мане, а ми вам нудимо кратку листу предлога за локацију вашег *hackerspace*-а:

1. Изнајмљени простор - потребан је новац, али је зато обезбеђен основним стварима наведеним горе.
2. Гаража - што да не? Доноси уштеду у новцу у смислу да не морате плаћати ренту.
3. Део ваше фирме у којој сте запослени - одлично место ако ваша фирма има могућности и воље да вам обезбеди простор.
4. Универзитет - ако сте студент, можете се распитати има ли воље за тако нешто и тако успут пронаћи нове чланове тима.



Када знате где ћете набавити опрему, алате и сам простор, потребно је организовати тим и поделити задатке. Много је важно да сваки члан буде одговоран и да испуњава своја обећања како би заједница заживела. Потребно је бити пажљив и узети у обзир друге обавезе осталих чланова, али и остали чланови треба да буду организовани и да испуњавају своје обавезе.

После свега долази и она болна тачка - новац. Новац је потребан ако узимате простор, плаћате вођење „папирологије“, или плаћате опрему, а исти се може прикупити на више начина:

1. Чланарина.
2. Продаја производа.

Пулс слободе

3. Донације.
4. Предавања и радионице.
5. Корпоративна спонзорства.
6. Изнајмљивање простора.

Како *Eric* препоручује, увек је паметно чувати одређени износ новца „за сваки случај”, ако вам се деси да изгубите спонзора и слично. Када је све ово готово, хаковање може да почне.

Корисни линкови:

- [1] <https://www.adafruit.com/blog/2012/11/12/how-to-start-a-hackerspace/>
- [2] <https://hackerspace.rs/>
- [3] <http://oosm.org/>





Ubuntu Mate 14.10



Аутор: Дејан Маглов

Двадесет и трећег октобра објављена је нова редовна стабилна верзија *Ubuntu 14.10* кодног назива *Utopic Unicorn*. *Ubuntu* нас је већ навикао да на сваких шест месеци избацује ново стабилно издање. Такође смо се навикли на то да верзија након *LTS* издања не доноси ништа спектакуларно ново. Ове верзије обично исправљају уочене грешке у *LTS* издању и тек наговештавају неке нове, будуће функционалности.

Можда и не бисмо писали текст о *Ubuntu 14.10* да ово издање није јубиларно. Наиме, пре тачно десет година, у октобру 2004. године, објављена је прва стабилна верзија *Ubuntu 4.10* кодног назива *Warty Warthog*.

Из историје

Иако *Warty Warthog* можда није била идеална *Linux* дистрибуција, ипак је она покренула *Linux* револуцију. *Canonical* који стоји иза пројекта *Ubuntu*, од самог старта пројекта имао је у глави револуционарну идеју да понуди *Linux* дистрибуцију, која ће бити намењена обичном кориснику, што у то време није био чест случај. *Debian*, који је послужио као основа за *Ubuntu*, имао је скоро све потребне алате да приближи *Linux* обичном кориснику, али сам није био толико једноставан за инсталирање, конфигурирање и коришћење.

Већ први *Ubuntu* је донео „живи“ инсталациони диск,



Представљамо

графички инсталер, *synaptic* као графички пакет менаџер, *Gnome 2* као главно графичко окружење са и без укључених графичких ефеката, једноставне графичке алате за прилагођавање изгледа графичког окружења, за конфигурисање мреже, конекције са интернетом, лепу колекцију прединсталираног софтвера и тако даље. Почетник практично, ако није хтео, није морао да улази у терминал да би подесио свој систем.

Оваква филозофија је одмах привукла бројну заједницу око овог пројекта, а то је „лансирало” *Ubuntu* у сам врх популарности међу многобројним *Linux* дистрибуцијама. Потврду добре филозофије је донело време. Поред *Debian*-а, *Ubuntu* је постао најчешћа основа за нове *Linux* дистрибуције. Неке од тих деривата *Ubuntu*-а су постале још једноставније за обичног корисника тако да је *Ubuntu* од филозофије „*Linux* за свакога” можда мало занемарио кориснике старијег хардвера, а акценат ставио на иновације и освајање што веће ширине хардвера од десктоп и лаптоп рачунара преко сервера, таблета, мобилних телефона и других уређаја.

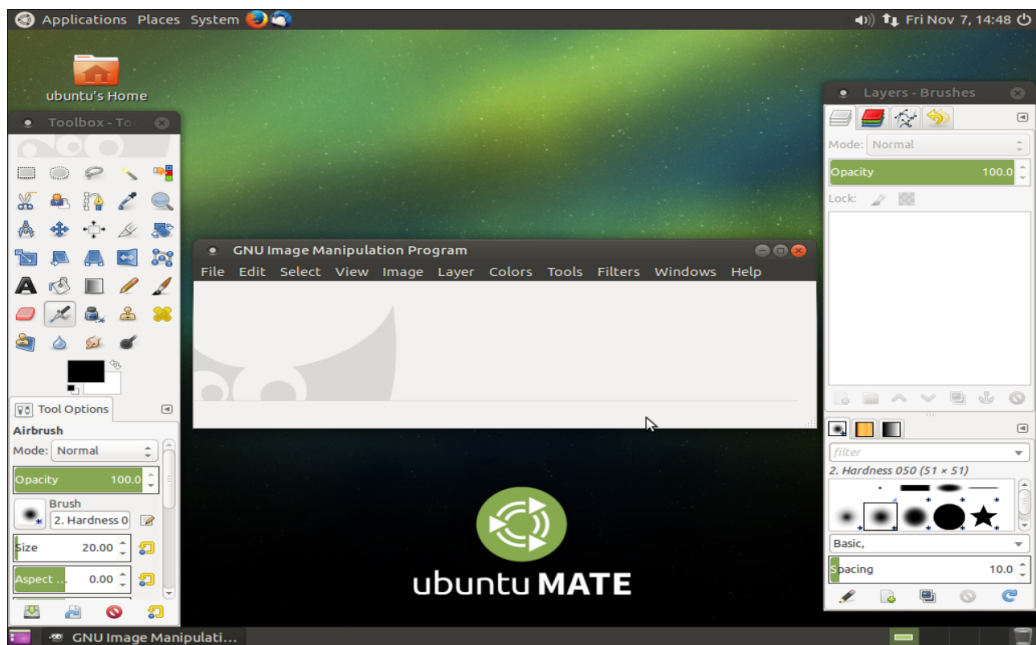




Ubuntu Mate 14.10

Упоредо са објављивањем званичних *Ubuntu* „укуса“¹ (*Kubuntu*, *Xubuntu*, *Lubuntu*, *Ubuntu Gnome* и други) ових дана је објављен један од, за сада, незваничних *Ubuntu* „укуса“ - *Ubuntu Mate 14.10*.

Ubuntu Mate доноси класични „укус“ старог *Ubuntu*-а са *Gnome 2* графичким окружењем који смо сви толико волели. Да се подсетимо, *Mate* је форк *Gnome 3* који доноси поново изглед и функционалност *Gnome 2* графичког окружења. Осим што *Mate* „гађа“ *Gnome 2* носталгичаре, он нам доноси и мању хардверску захтевност у односу на *Gnome 3*. Мања захтевност *Mate* графичког окружења гарантује већу позорност, не само носталгичара него и корисника са слабијим хардвером.



¹ **Укус** (енг. *flavor*) представља врсту *Ubuntu* система. Врсте се међусобно разликују по графичком окружењу и по прединсталираном софтверу и алатима. У основи сваке врсте је *Ubuntu* оперативни систем.

Представљамо

Mate је задржао изузетну конфигурабилност *Gnome 2* окружења. Не само да је могуће прилагодити изглед *Mate*-а по сваком укусу, већ је и то јако лако урадити помоћу графичких програма за ту намену. *Mate* одлично сарађује са *Compiz* и *Emerald* менаџерима прозора што доноси још могућности за унапређење изгледа овог графичког окружења.

Спој одличне *Ubuntu* основе и *Mate* графичког окружења гарантује потпуни успех овог *Ubuntu* „укуса“ који ће врло брзо, сигурни смо у то, постати део званичних *Ubuntu* „укуса“. Иако је штета што прва верзија *Ubuntu Mate*-а није изашла шест месеци раније заједно са *LTS* верзијом *Ubuntu*-а, овако темпиран излазак се савршено поклапа са годишњицом *Ubuntu*-а и враћа нас у носталгичне дане *Ubuntu*-а без графичког окружења *Unity*.

Карактеристике

Као и званични *Ubuntu* „укуси“, *Ubuntu Mate* долази као *ISO* слика за „живи“ инсталациони диск или *USB* стик величине *1GB* (живи *DVD*). *ISO* слика се може преузети са странице пројекта <https://ubuntu-mate.org/>. Корисницима су на располагању *ISO* верзије за 32-битне и 64-битне рачунаре (*PC*) као и 64-битна верзија за *Apple Mac* рачунаре.

По спецификацији минимални хардверски захтеви за *Ubuntu Mate 14.10* су:

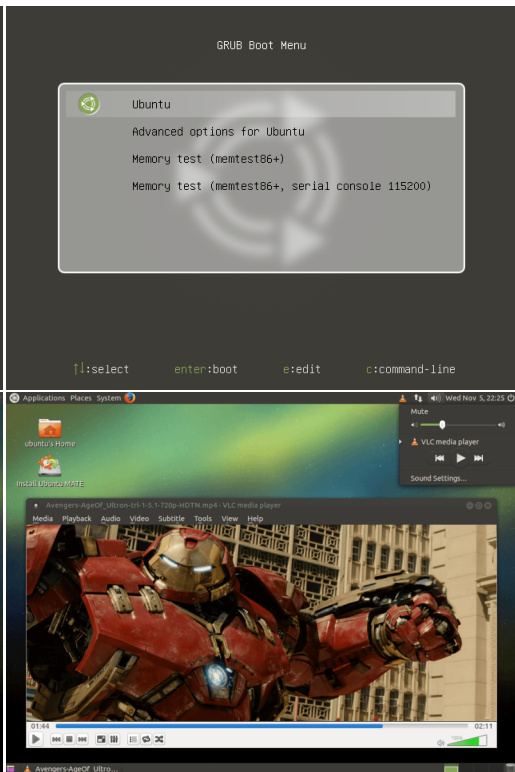
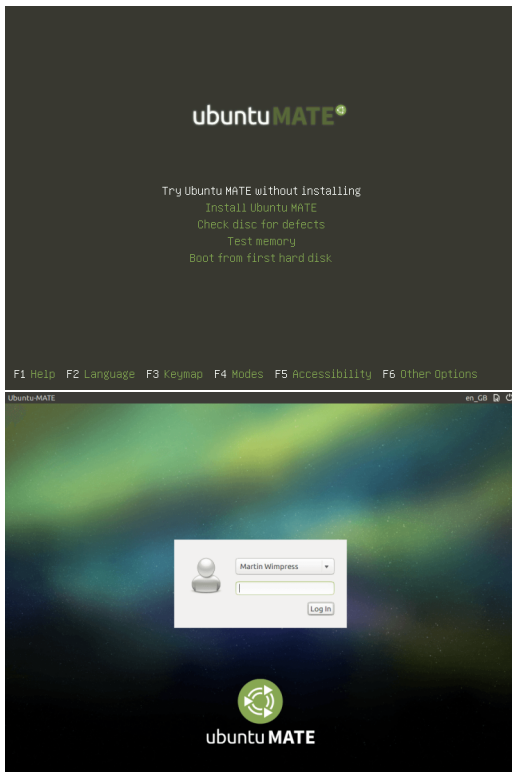
- *Pentium III 750MHz*
- *512MB RAM*-а
- *8GB* празног простора на тврдом диску
- Бутабилни *DVD-ROM* читач/писач
- *SVGA* видео адаптер резолуције *1024x768*

Иако је незванични *Ubuntu* пројекат, *Ubuntu Mate* живи диск је рађен истим алатима као и сви остали *Ubuntu* „укуси“. Самим покретањем „живог“ диска, осим што има промењене боје, ни по чему се не разликује од своје друге званичне браће. Исто се може рећи и за сам графички инсталер. Ко је једном инсталирао било који новији *Ubuntu* „укус“, неће имати проблема ни са инсталацијом *Ubuntu Mate*-а.

Свеже инсталирани *Ubuntu Mate* неће својим подразумеваним изгледом разочарати ни захтевније кориснике. Аутори су се озбиљно потрудили да до детаља дизајнерски „испеглају“ ову верзију *Ubuntu*-а. И за нас је било



изненађење кад смо увидом у тим који је реализовао *Ubuntu Mate*, видели да су за овако добар изглед одговорна и два човека са наших простора – Гоце Митевски и Иван Пејић. За оне још захтевније, *Ubuntu Mate* има сасвим довољно алата, припремљених тема и *wallpaper*-а, тако да корисник може да га уреди баш по свом укусу.



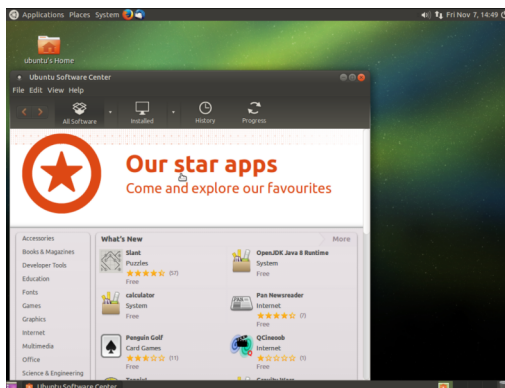
Ubuntu Mate 14.10 од прединсталираног софтвера доноси:

- *Kernel 3.16*
- *Firefox 33*
- *Thunderbird 31.2*
- *HexChat 2.10.0*
- *Pidgin 2.10.9*
- *Atril PDF reader 1.8.1 (Мате верзија Evince-а)*

Представљамо

- *ShotWell* 0.20.1
- *Caja* 1.8.2 (*Mate* верзија *Nautilus-a*)
- *Pluma* 1.8.1 (*Mate* верзија *Gedit-a*)
- *Brasero* 3.10.0
- *Totem* 3.10.1
- *Rhythmbox* 3.0.3
- *Cheese*

Овај пакет прединсталираног софтвера је углавном стандардан у осталим *Ubuntu* „укусима“, и обезбеђује добру основу за даљи рад на овом оперативном систему. За све остало што кориснику може да недостаје, задужен је *Ubuntu Software Center* 13.10, који је стандардни графички пакет менаџер у свим издањима *Ubuntu-a*.



Утисци

Први утисак је да *Ubuntu Mate* јесте пријатно изненађење. Првенствено је намењен десктоп рачунарима са нешто слабијим хардвером. Иако спецификација тврди да он може да ради на рачунарима са 512MB RAM-а, ми нисмо убеђени да би тај рад био много угодан. На „леру“ *Ubuntu Mate* „троши“ око 300MB RAM-а, тако да за неки озбиљнији рад нема много простора. Овако слаб рачунар би пречесто посезао за SWAP простором што би га трагично успорило, зато је наша препорука да је апсолутни минимум за иоле озбиљнији рад на *Ubuntu Mate* рачунару 1GB радне меморије.

Ubuntu Mate прети да буде први избор за *Ubuntu* класик носталгичаре. Потребно је само да сачекамо да овај систем мало сазри и да „преболи стандардне дечије болести“. До тада ће, највероватније, постати и званични *Ubuntu* „укус“ што ће му обезбедити и сигурнију будућност.



libGDX

„Java game development framework“

(5. део)

Аутор: Гаврило Продановић

У претходним бројевима говорили смо о *LibGDX*-у обазирјући се углавном само на техничку страну овог *framework*-а и тиме смо заобишли остале компоненте које су сачињене углавном од естетских ствари као што су документација, популарност и заједница која се развила око овог пројекта док га је у исто вријеме креирала. Може се сматрати да је рађање *LibGDX*-а почело као „приватно путовање“ *Mario*-а *Zechner*-а, средином 2009. године када је почео да се интересује за писање игрица на *Android* платформи, да би тек 6. марта 2010. године *LibGDX* угледао свјетлост дана као *open source* пројекат, од када су и други могли видјети код ове библиотеке и могли да га користе. У априлу



Како да...?

исте године *LibGDX* је постао тимски пројекат и почео је са убрзаним развојем да би данас постао један озбиљан *framework* за развој игрица, на шта указује велики број развијених игрица у њему.

Колико год неки *framework* био моћан и једноставан за коришћење, документација ће увјек бити потребна оним програмерима који га користе. На *wiki*-ју овог пројекта може се наћи велики број различитих упутстава која улазе у различите аспекте *framework*-а, а постоји и један чланак који ће вас увести у основе *LibGDX*-а креирајући са вама једноставну игрицу. Други подједнако битан извор документације је *javadoc* у којем ћемо наћи све класе и њихове методе које су дефинисане. Иако је већина метода у *javadoc*-у добро документована, наишли смо на неколико којима недостаје опис, али логиком смо могли лако да закључимо чему служе. *Source code* је велики извор сазнања, а у склопу њега постоје званични примјери које можемо да покренемо и видимо на дјелу. У случају да негдје запнемо у развоју, постоји званичан форум и *IRC* канал на коме је углавном весело и влада пријатељско расположење. *LibGDX* је заживјео на осталим мјестима као што је *StackOverflow*, гдје се може наћи рјешење за велики број проблема и нејасноћа на које наиђемо за вријеме развоја у овом *framework*-у.

The screenshot shows the GitHub repository page for `libgdx / libgdx`. The page is titled "Table of Contents" and features a "Wiki Style Guide" section. The "Developer's Guide" is expanded, showing a list of topics:

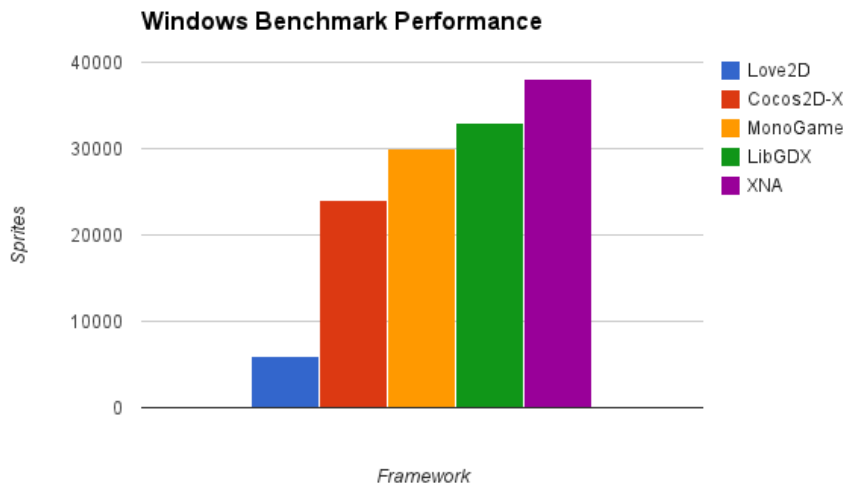
- **Developer's Guide**
 - Introduction
 - Goals & Features
 - Community & Support
 - Contributing
 - Games Built with Libgdx
 - Setting up your Development Environment (Eclipse, IntelliJ IDEA, NetBeans)
 - Creating, Running, Debugging and Packaging your Project
 - Eclipse
 - IntelliJ Idea
 - NetBeans
 - Commandline
 - Updating your libGDX version
 - Adding extensions and 3rd party libraries
 - Improving your Gradle workflow
 - Working from Source
 - Running Demos
 - Running Tests
 - Building libgdx from Source
 - The Application Framework
 - The Life-Cycle
 - Modules Overview
 - Starter Classes & Configuration
 - Querying



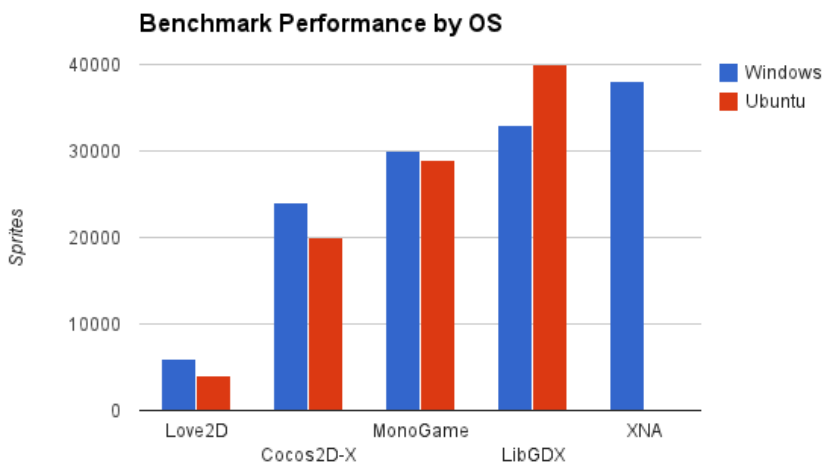
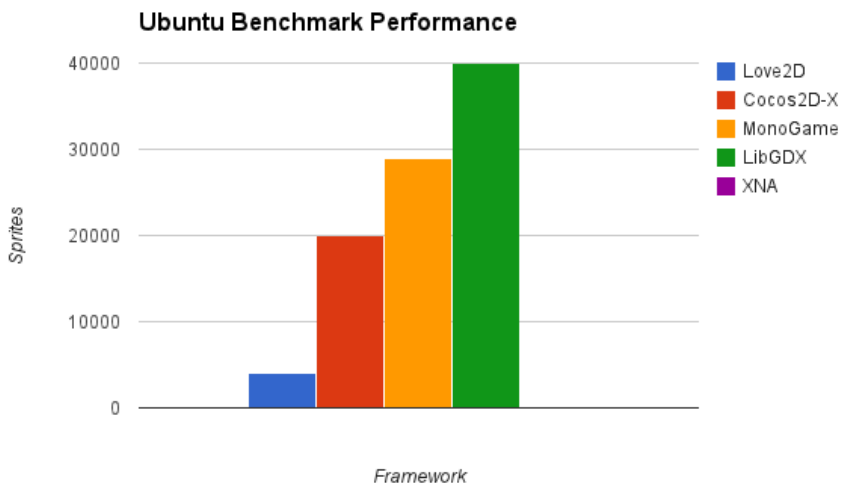
У случају да не развијате игрицу отвореног кода и желите да је монетаризујете, на *wiki*-ју ће вам помоћи чланак о томе како да интегришете *AdMob* и *AirPush* у свој *LibGDX* пројекат. Постоји подршка за сервисе као што су *Swarm*, *Google Play Store* и *Nextpeer*. На питање да ли је могуће доћи до профита од *LibGDX*-а, позитиван одговор даје одређен број игрица, које се могу наћи на *Play Store*-у. Базиране су на овом *framework*-у и прешле су преко пола милиона преузимања (прим.аут.). У суштини, да ли ће игрица бити успешна, зависи од вас и ваше идеје и од тога како је имплементирате. Око првих корака у промовисању игрице помоћи ће вам могућност да је додате у галерију матичног сајта овог *framework*-а.

За развијање видео игрице увијек битна ствар је број фрејмова у секунди. Из нашег скромног личног искуства можемо рећи да са овим *framework*-ом никада нисмо имали проблема по питању *FPS*-а ако смо разумно користили ресурсе са рационално написаним кодом и да смо добили сасвим стабилан *fps* у једноставнијим игрицама на доста старим и јефтним *Android* уређајима. Пошто детаљније тестирање захтјева читав један чланак, позајмићемо се са неколико графова који представљају тестирање *LibGDX*-а са упоредо другим библиотекама урађеном крајем 2012. године:

<http://www.sparkrift.com/2012/1/love2d-vs-allegro-vs-clanlib-vs-libgdx-vs-cocos2d-x-vs-monogame-vs-xna-vs-sfml> .



Како да...?





Вјерујемо када би се у скорије вријеме поновило тестирање, да би *LibGDX* дао још боље резултате.

За крај топло препоручујемо *LibGDX* за све оне који желе да брзо своје идеје спроведу у квалитетно дјело, са тим да морамо напоменути да *LibGDX* нас не форсира много да неке ствари радимо тачно на одређен начин, него нам оставља програмерску слободу да одаберемо онако како нама прија.



Преглед популарности *GNU/Linux* /*BSD* дистрибуција за месец октобар

Distrowatch

1	Mint	2400<
2	Ubuntu	2262<
3	openSUSE	1949>
4	Debian	1824<
5	Mageia	1501<
6	CentOS	1388<
7	Fedora	1383>
8	Arch	1187<
9	elementary	1006>
10	LXLE	984<
11	Puppy	922>
12	Lubuntu	895<
13	Kali	888<
14	Bodhi	797<
15	Android-x86	783>
16	Zorin	730=
17	FreeBSD	692<
18	ReactOS	672>
19	SteamOS	663>
20	PCLinuxOS	627<
21	Xubuntu	604>
22	Tails	582<
23	Lunar	574>
24	ROSA	571=
25	Deepin	571=

Пад <
 Пораст >
 Исти рејтинг =
 (Коришћени подаци са Distrowatch-a)

Како да...?

Увод у програмски језик C

(6. део)

Аутор: Вељко Симић

Показивачи

Меморију рачунара можемо да посматрамо као низ меморијских локација. Меморијске локације обележавамо бројевима од 0 па до капацитета меморије и називамо их меморијским адресама. Показивачи су променљиве које као вредност садрже адресу меморијске локације неке променљиве. Показивач такође има и тип. Тип показивача је тип вредности на коју показивач показује (референцира). Показивачи се дефинишу као и сви остали подаци, наводе се његов тип (тачније тип вредности на коју референцира) и назив, с тим што се пре његовог назива пише „*” - која означава да је нека променљива показивач. Дефинишимо показивач *p* типа *int:int *p*;

Меморијској локацији неког елемента приступа се навођењем знака „&” испред променљиве. Ако је *x* променљива типа *int*, следећом наредбом ћемо доделити показивачу *p* адресу *x*-а.

```
p=&x;
```

Уколико желимо да приступимо вредности која се налази на адреси коју референцира *p*, то ћемо учинити коришћењем „*”. Ако је променљива *a* типа *int*, следеће две наредбе ће учинити да вредност *a* има исту вредност као *x*.

```
p=&x;
a=*p;
```

Функција *sizeof(x)* за параметар прима променљиву или тип неког податка, а као резултат враћа величину меморијског простора коју заузима прослеђен



параметар. Вероватно вам се намеће питање чему потреба за показивачима. Потреба показивача је огромна, а ми ћемо у овом броју навести основне и најкоришћеније примере. Нека је задатак да напишемо функцију која замењује вредности два броја.

```
void razmeni (int a,int b){
    int tmp = a;
    a = b;
    b = a;
}
```

Уколико тестирамо ову функцију, приметимо да функција не ради оно што бисмо желели.

```
int a = 5, b = 7;
razmeni(a,b);
printf ("%d %d",a,b);
```

Као резултат рада ове три линије, добићемо испис на екрану: 5 7. Да бисмо објаснили ову појаву, најпре ћемо појаснити како ради функција. Функција заправо ради са копијама вредности својих аргумената и такво прослеђивање се назива прослеђивање аргумената по вредности. Вредности можемо променити унутар функције уколико као параметар наведемо њихову меморијску локацију, па би исправна верзија претходно написане функције изгледала овако:

```
void razmeni (int* a,int* b){
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
```

У овом случају прослеђујемо меморијску локацију променљивих:

```
razmeni (&a, &b);
```

Низови

При дефиницији низа морамо навести број елемената тог низа. Мана тога је што

Како да...?

морамо да скоро увек резервишемо више него што нам треба (за сваки случај), ако то не урадимо, веома лако можемо доћи у ситуацију да немамо довољан број елемената низа. У том случају програм изазива грешку *segmentation fault* - то је грешка која се јавља уколико покушамо да приступимо меморијској локацији којој није дозвољен приступ. На пример, затражимо вредност елемента $n[5]$, где је n дефинисан као низ од три елемента.

У оваквим ситуацијама користи се динамичко алоцирање меморије, тачније додељивање више меморијских локација једном показивачу. Дефинисање низа од n елемената извршава следећи део кода:

```
int *niz, n;
scanf("%d", &n);
niz = malloc(n*sizeof(int));
for (int i=0; i<n; i++)
    scanf("%d", niz+i);
for (int i=0; i<n; i++)
    printf("%d ", *(niz+i));
free(niz);
```

Анализирајмо код. Као што видите, имамо две непознате функције *malloc()* и *free()*. Функција *malloc* алоцира простор тражене величине и враћа показивач на њега, у нашем примеру желимо да заузмемо простор за n података типа *int*. Функција *free* ослобађа заузету меморију. i -том члану низа приступамо тако што ћемо приступити i -тој меморијској локацији низа $*(niz+i)$. Следећа два исказа су еквивалентна $*(niz+0)$ и $*niz$. У следећем броју ћемо причати о сложеним типовима података, тј. о структурама података и имплементацији најпознатијих сложених структура: листа, стек, стабло и граф.

Learn C Programming



У потрази за идеалном дистрибуцијом:

Још критеријума за одабир идеалне дистрибуције (2. део)

Аутор: Дејан Маглов

До идеалне дистрибуције може се доћи насумичним испробавањем или по препоруци искуснијих корисника. Под препоруком подразумевамо бирање најпопуларније дистрибуције на *distrowatch.org*. Ако је дистрибуција најпопуларнија на *Distrowatch*-у, то није случајно. То значи да је велики број корисника препознао ту дистрибуцију као добру, једноставну, са добром подршком, са добром документацијом и лепог изгледа. Овде су кључне речи „велики број корисника” јер то и даље нису сви корисници. Ако најпопуларнија дистрибуција не ради добро код вас, то а priori не значи да *GNU/Linux* не ваља. *GNU/Linux* може да покреће и управља суперкомпјутерима, серверима, затим десктоп, лаптоп и нетбук рачунарима, таблетима, паметним кућама, аутомобилима, паметним кућним апаратима али не са истим кернелом, пакетом истих *GNU* софтвера и подешавањима. Универзална покривеност свих постојећих хардвера је немогућа у једном парчету софтвера, зато постоје бројне модификације основне архитектуре софтвера, а на нама је да пронађемо одговарајуће компоненте и комбинацију истих за наш хардвер.



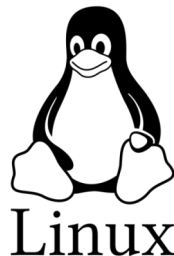
У овом броју настављамо да идентификујемо критеријуме за одабир идеалне дистрибуције и објашњавамо шта сваки од тих избора доноси. Критеријуми за избор су бројни. Само правим избором моћи ћете на крају да кажете да имате

Ослобађање

приближно идеалну дистрибуцију за вас.

Критеријум избора по процесорској архитектури

Неки од критеријума директно су везани за хардвер који поседујете. У прошлом броју смо поменули кернел, графичко окружење и њихов утицај на рад постојећег хардвера. Постоји још један критеријум који је директно везан за постојећи хардвер. Реч је о процесорској архитектури. Иако постоје бројне процесорске архитектуре, обичног корисника интересује само најраспрострањеније архитектуре процесора за *PC* рачунаре (десктоп и лаптоп рачунари), нетбук рачунаре, таблете и мобилне телефоне. Говоримо о *ARM* процесорској архитектури за нетбук рачунаре, таблете и мобилне телефоне, *PCpower* за моторола процесоре и *x86, x86-64* за *Intel*-ове и *AMD* процесоре за десктоп рачунаре. За сваки од ових процесорских архитектура постоји развијен посебан *Linux* кернел.



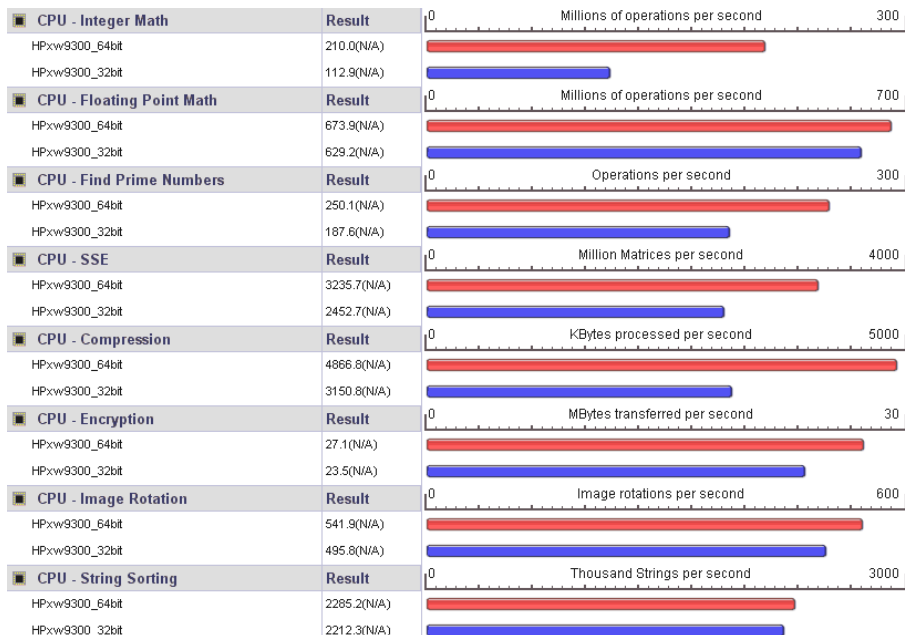
На први поглед, избор правог система по процесорској архитектури је лак. Принцип је једноставан – бирамо кернел према процесору који поседујемо. Компликације настају само у случају да поседујете 64-битни *Intel*-ов или *AMD*-ов процесор *x86* архитектуре. У том случају имате дупли избор: *x86* 32-битни кернел или *x86-64* 64-битни кернел. Оба кернела ће радити на том процесору.

Потребна нам је додатна варијабла која ће нам помоћи да се одлучимо за прави кернел. Додатна варијабла је *RAM* (радна меморија). По дефиницији 64-битни кернел је бржи од 32-битног, јер барата са дупло више података у једном кораку. Барата са 64-битном бинарном речи уместо 32-битном. То ипак не значи да је дупло бржи, јер то зависи од самог процеса. Ако барата са малим подацима за које је довољно и 32 бита, број манипулација са подацима ће бити једнак као и код 32-битног система и повећање брзине неће бити примећено. Када упростимо и направимо аналогију: рецимо, имамо два камиона, један има носивост од две тоне а други од четири тоне. Већи камион ће бити бржи ако треба превести педесет тона растресите робе. Међутим, већи камион неће бити ништа бржи ако је потребно превести укупно само две тоне, или ако запремински не може да прихвати више од једног комада од две тоне у једном превозу, јер то може да одради и мањи камион у једнаком броју пролаза. Према томе, ако софтвер није



У потрази за идеалном дистрибуцијом

тако реализован да искористи предности веће бинарне речи, корисник неће приметити предност 64-битног система. Са друге стране, већа бинарна реч било да је пуна информација, или само до пола искоришћена, троши пун меморијски простор. Зато 64-битни систем троши више RAM-а у процесима где бинарне речи нису у потпуности искоришћене.



У случајевима лошег или 32-битног софтвера, 64-битни систем не ради брже, а поврх тога троши више меморије. Да ли има предности? Наравно да има. Бржи је са остатком апликација које су нативно 64-битне, а тада троши једнаку количину меморије као и 32-битни систем који у 32-битној интерпретацији те апликације троши два меморијска места да би пренео 64-битну информацију.

Постоји још једно ограничење 32-битног система. Он због 32-битне речи може да адресира максимално 3.2GB RAM-а. Стога, наша препорука је да корисници 64-битног процесора са мање од 3GB RAM-а користе 32-битни систем јер су уштеде меморије знатне. Они који имају преко 4GB RAM-а, њима препоручујемо 64-битни систем због предности у брзини.

Ослобађање

Они који имају тачно 4GB RAM-а, у малом су проблему. Њима уштеда RAM-а који доноси 32-битни систем значи можда више од евентуалног повећања брзине. Са друге стране, због немогућности индексирања свих 4GB, морали би да се одрекну 0.8GB RAM-а јер ће им у 32-битном систему бити доступно само 3.2GB. Међутим, постоји решење да 32-битни систем искористи и тих 0.8GB. Чак и 32-битни процесори већ одавно имају уграђену PAE (енг. *Physical Address Extension*) подршку која им омогућава да индексирају више од 3.2GB, тако да постоје и 32-битни Linux кернели са PAE подршком. Најчешће нису стандардни део GNU/Linux 32-битног система. Изузеци су Red Hat Enterprise Linux / CentOS, Ubuntu и Mint који од 2012. године користе 32-битни Linux кернел са PAE подршком као стандардни 32-битни кернел. Остале дистрибуције користе non-PAE кернеле, али већина њих има у ризницама PAE кернел који се може накнадно инсталирати.

Сада имате потпуну информацију о начину бирања идеалне дистрибуције са нагласком на расположиви хардвер. Остало вам је сада да бирате систем према софтверским решењима.

Још неколико информација о критеријуму избора преко пакет менаџера

У прошлом броју смо објаснили чему пакет менаџери служе и које функције имају. Рекли смо и да је Linux софтвер тимски играч који има своју специфичну функцију и задатак, али носи исти дрес као све остале апликације у том тиму, има исту апликацију за одржавање, користи исте апликације за управљање хардвером итд. Windows апликације су индивидуалне - оне све функције имају интегрисане у свом софтверу осим функција који су део самог оперативног система. Рекли смо да оваква особина слободног софтвера компликује његову инсталацију, али зато изузетно штеди складишни простор на тврдом диску.

Наставак приче о пакет менаџерима је, у ствари, прича о дистрибуцијама. Да разјаснимо: имамо три старе главне дистрибуције (*Debian*, *Slackware* и *Red Hat*), неколико новијих независних већих дистрибуција (*Gentoo*, *Puppy* и *Arch*), нешто мањих независних пројеката, а све остало су деривати ових дистрибуција (<http://futurist.se/gldt/wp-content/uploads/12.10/gldt1210.svg>). Управо поглед на пакет менаџер открива којој фамилији припада дотична дистрибуција.

Филозофија FLOSS-а која охрабрује проучавање кода и вршење измена, допринела је развоју бројних форкова основног кода главних дистрибуција и стварању нових



У потрази за идеалном дистрибуцијом

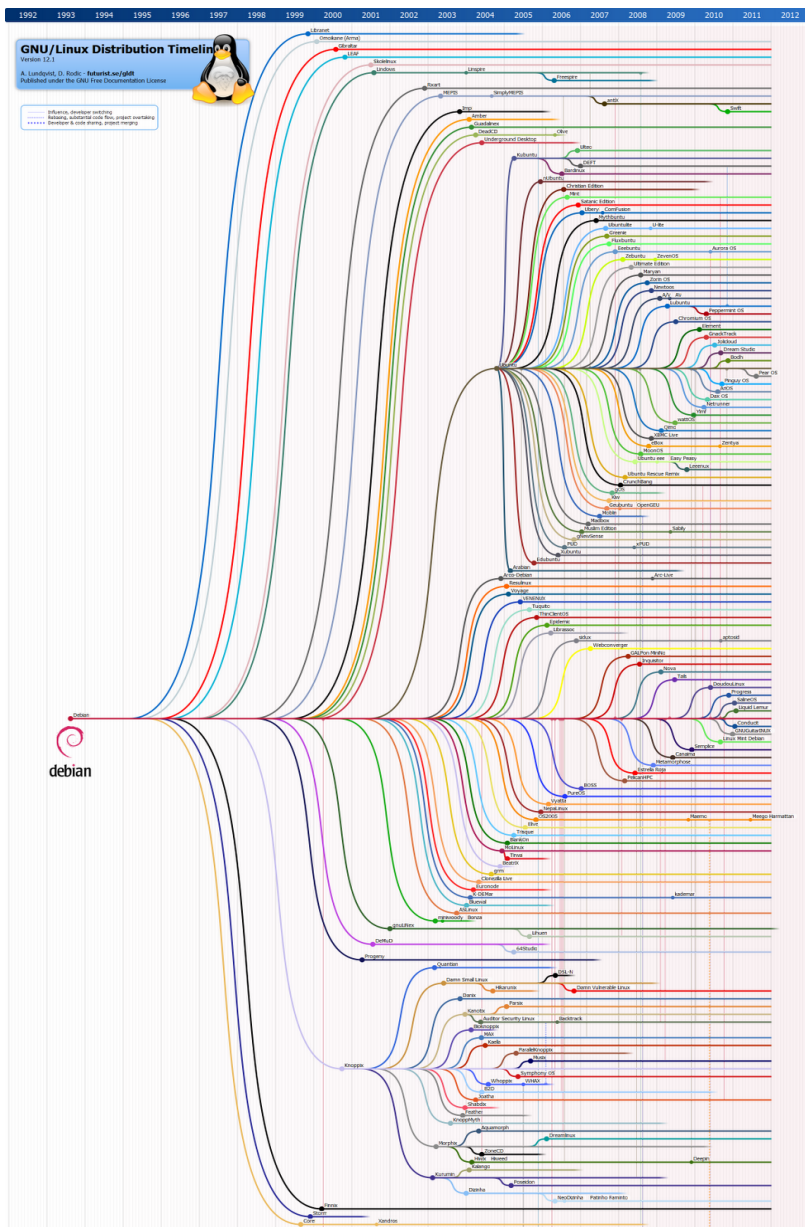
деривата од основних система. Већина тих деривата је настала као потреба за променом филозофије основног система. На пример, филозофија *Debian*-а је строго канонизована и подразумева строго држање стабилног и сигурног, искључиво слободног софтвера унутар свог система. *Ubuntu* ту филозофију мења у примену новијег софтвера са не тако провереном сигурношћу, укључивањем бесплатног власничког софтвера са циљем да задовољи више потреба корисника. Овакве разлике у филозофији се не могу помирити под истим „кровом” и због тога настаје нови *Debian*-ов дериват, *Ubuntu*. Није ово једини разлог настанка *Ubuntu*-а из *Debian*-а, али је добра илустрација како настају нови деривати.



Искуство стечено упознавањем функционисања само једног деривата једне фамилије *GNU/Linux*-а може се применити са минималним разликама на целу фамилију. Разлика између фамилија је много већа. Не гарантујемо да ће се почетник лако снаћи преласком из *Debian*-ове фамилије у *Red Hat* фамилију (или било коју другу), мада је и даље у питању *GNU/Linux*. Основа је иста, али до неких ствари се долази на другачији начин, што је ствар филозофије сваке фамилије понаособ.

Пошто је немогуће у основној инсталацији система очекивати сав потребан софтвер који нама треба, прво што ће нам затребати, јесте услуга пакет менаџера. Зато упознавање са пакет менаџером јесте једна од основних и

Ослобађање





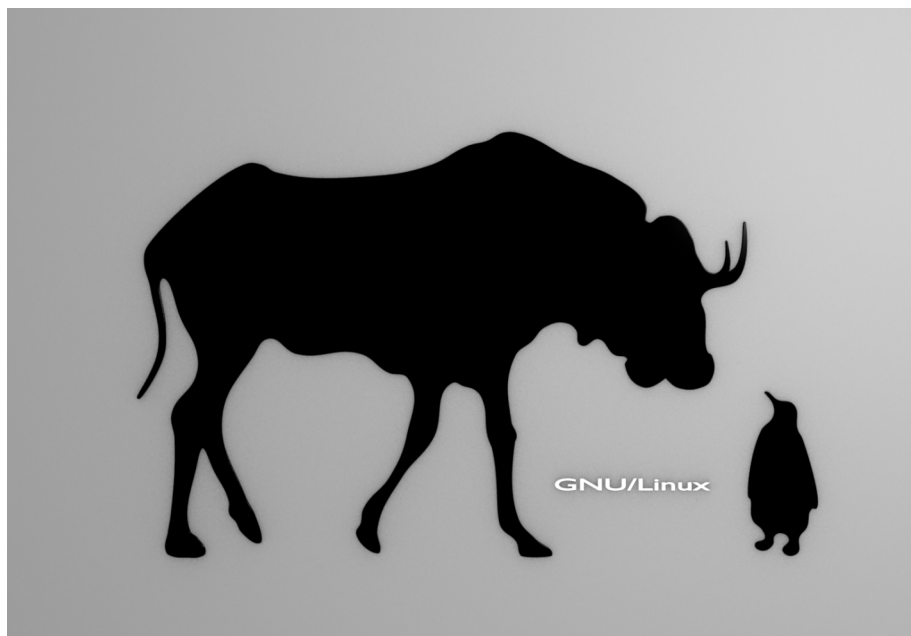
У потрази за идеалном дистрибуцијом

најважнијих ствари у савладавању *GNU/Linux* дистрибуције. Ако нам пакет менаџер одговара, на остале функције система ћемо се много лакше навићи.

Избор пакет менаџера који нам највише одговара, ограничава избор идеалне дистрибуције на само ту фамилију *GNU/Linux* система. То може да нам олакша избор, али смо још далеко од краја избора барем у случају *Debian*-ове фамилије и његових *dpkg* и *apt* пакет менаџера. Једноставност форковања *Debian*-а и постојање алата за тај посао довела је до тога да 60-70% свих деривата *GNU/Linux* оперативних система чине управо деривати, директно или индиректно, *Debian OS*-а.

За крај епизоде

И даље ћемо бити у потрази за идеалном дистрибуцијом. Остало је још много критеријума за избор и надамо се да ћемо ипак на крају доћи до приближно идеалног резултата. Приметили сте да углавном не сугеришемо шта је добро за вас. Дајемо вам чињенице, а на вама је да сами одаберете шта је идеално за вас.



Енкриптована електронска пошта (4. део)

Аутор: Петар Симовић

Шифровање поште више није баук - знамо то да применимо на више начина. Оно што нас сада занима, јесте да унапредимо нашу приватност и да некако прикријемо наш идентитет током размене електронске поште, тј. да страном посматрачу наше мреже (нпр. *ISP - internet service provider*) што више отежамо да открије са киме се дописујемо. Нешто слично ради и *Darkmail* за мобилне телефоне, који је недавно променио своје име у *DIME (Dark Internet mail environment)* и представио је свој истоимени протокол за размену поште на *DEF CON*-у.

Постоје у основи четири начина да се дописујете са неким путем електронске поште, а да трећа страна не зна са киме заиста комуницирате. Примарни циљ оваквих мрежа и протокола јесте да прикрију идентитет пошиљаоца и примаоца порука, што није било могуће употребом самог шифровања садржаја поруке. Ови начини су још познати као четири типа анонимних „римејлера“ (енг. *remailer*). Римејлери су прослеђивачи порука и служе да поруку од пошиљаоца приме и пошаљу је примаоцу, што је довољно да знамо о њима за сада.

Један начин је путем псеудонимних римејлера (енг. *pseudonymous remailer* или *num server*), а остала три начина су путем три типа римејлера. Први тип римејлера је познат и као Сајферпанк римејлер (енг. *Cypherpunk remailer*), други као Миксмастер римејлер (енг. *Mixmaster remailer*) и трећи као Миксминион римејлер (енг. *Mixminion remailer*). Ми ћемо се углавном бавити Миксмастер римејлерима, док ћемо о осталима рећи само по нешто.

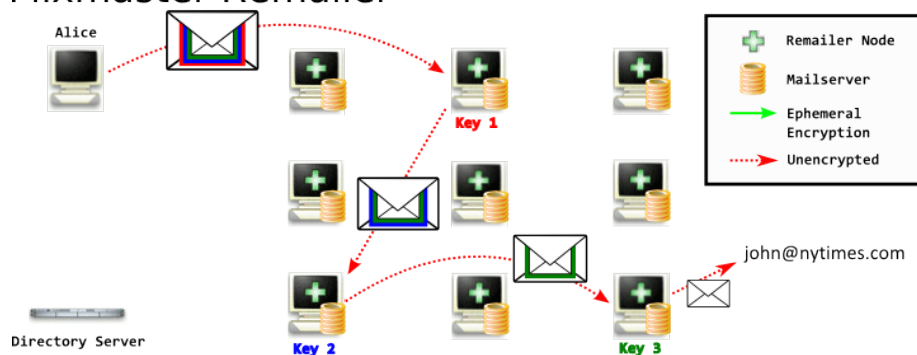
Num сервери, уопштено говорећи, само прикривају ваш прави идентитет тако што вам додељују ново име или псеудоним (зато се зову *pseudonymous remailers*) како бисте под тим надимком могли да псеудо-анонимно постујете поруке на



Енкриптована електронска пошта

Usenet-у (користе се *Usenet* групе за постовање, а то је нека врста заједничког мејл сандучета где су све поруке које се постују шифроване), или се дописујете порукама. Нећемо овде улазити у детаље функционисања, само ћемо рећи да постоје и напредније технике које овој врсти римејлера дају (условно речено) предност. Замисао је да свако ко жели да прочита пошту намењену себи са, рецимо *Usenet* групе *alt.anonymous.messages* (<http://goo.gl/taVCi5>), мора претходно да преузме све поруке које су икада постављене на групу и проба да их дешифрира једну по једну својим приватним кључем, а дешифрираће ону која је за њега намењена тј. која је шифрована његовим приватним кључем. На тај начин прикрива информацију о томе да ли уопште прима поруке као и колико их је примио јер сваки пут преузима све и све поруке покушава да дешифрира. Упутства за креирање *Nym* сервера можете наћи на следећим веб странама <http://goo.gl/A9yG5B>, <http://goo.gl/2EQMwI>.

Mixmaster Remailer

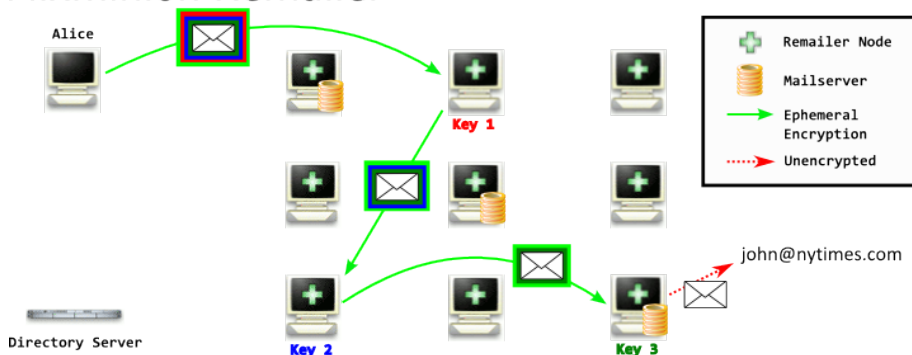


Сајферпанк римејлер је први тип римејлера који ради са мејловима исто што и *proxy* сервер са веб саобраћајем који му долази од корисника. Наиме, овај тип римејлера преузима хедере (енг. *headers*) мејл поруке коју је примио како би прикрио извор тј. пошиљаоца. Хедери мејл порука садрже адресу пошиљаоца и разне друге метаподатке, зависно од мејл протокола који се користи који одају информације страном посматрачу ко, са киме и кад комуницира и у којој мери. Осим тога, корисник претходно одабере кроз које ће Сајферпанк римејлере порука да се рутира и исту шифрује јавним кључевима редом оних рачунара које је одабрао. Овде се може увидети аналогија са *Tor* мрежом, иако сте то помислили и ви нисте много промашили, само што је овај протокол/систем намењен искључиво за слање порука. Треба напоменути да овај тип римејлера

Интернет мреже и комуникације

има многе мане које су превазиђене у другој и трећој генерацији анонимних римејлера, те се практично више не користи, нити се сматра довољно безбедним. Миксмастер припада другој генерацији анонимних римејлера и већином се користи и данас. Користи неке напредније технике за избегавање деанонимизације корисника путем напада временских и величинских корелација и анализа корисничких комуникација унутар мреже. У преводу, страни посматрач мреже може надгледати величине порука које су послате са вашег рачунара, величине података које примају римејлери и које исти римејлери шаљу даље до величине порука које последњи (излазни) римејлер шаље и коме шаље, и на тај начин схватити да се ви дописујете са одређеном особом, иако не могу прочитати садржај поруке јер је шифрована. У основи је идеја да се порука дели у пакете фиксне величине (20480 бајтова, што омогућава употребу RSA-4096 шифровање за сваки пакет). Сваки пакет се појединачно шифрује кључевима Миксмастер сервера кроз које ће се порука рутирати. Миксмастер сервери који примају поруке, дешифрују један слој шифровања, онда сазнају коме даље треба да проследи поруку и тако све до последњег миксмастер сервера који поруку коначно шаље примаоцу. Осим тога, Миксмастер сервери не обрађују поруке одмах након што их приме, него чекају одређену количину порука да се накупи, онда све поруке измешају и пошаљу све одједном или једну по једну. На тај начин посматрач не може да повеже поруке јер све имају исту величину, а римејлери их не шаљу по примању него после времена условљеног само количином порука које кроз тај сервер пролазе. Овако су избегнути временски и количински напади и анализе мреже, али и овај систем има мане и унапређења које примењује трећа генерација римејлера. Више на <http://goo.gl/RjSZYL> и <http://goo.gl/tBV570>

Mixminion Remailer





Енкриптована електронска пошта

Миксминион римејлери уводе још неке новине у односу на Миксмастер. Једна од најкориснијих јесте да је осим порука и сама комуникација између римејлера шифрована (користи се SSL) заједно са ефемерном енкрипцијом (за сваку комуникацију генеришу се и користе посебни кључеви) која у основи омогућава PFS (енг. *Perfect Forward Secrecy*) о којој је било речи у прошлом делу. Такође, делимично се решава проблем римејлер сервера код Миксмастера и Сајферпанка који рутирају поруке, јер се рутирање може вршити само кроз сервере, док Миксминион омогућава рутирање кроз кориснике, али је последњи (излазни) римејлер потребно да буде посвећени сервер.

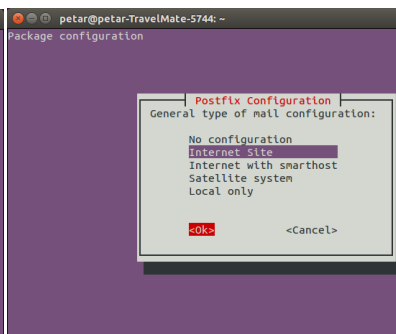
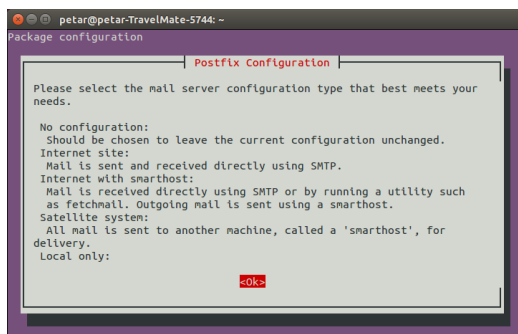
Што се Миксмастера тиче, инсталација је веома једноставна:

```
sudo apt-get install mixmaster
```

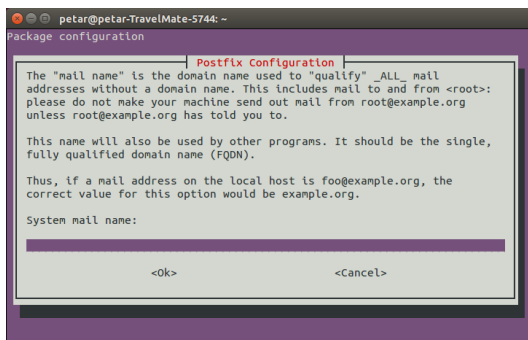
После инсталације требало би га ажурирати и покупити јавне кључеве Миксмастер сервера командом:

```
sudo mixmaster-update mixmaster-update --verbose
```

У току инсталације би требало да вам се покрене и инсталација **Постфикс** програма на који се **Миксмастер** ослања за слање порука. Уколико се ово пак не деси, требало би да **Постфикс** уклоните са вашег система и пустите да га **Миксмастер** у току инсталације сам инсталира.



Интернет мреже и комуникације

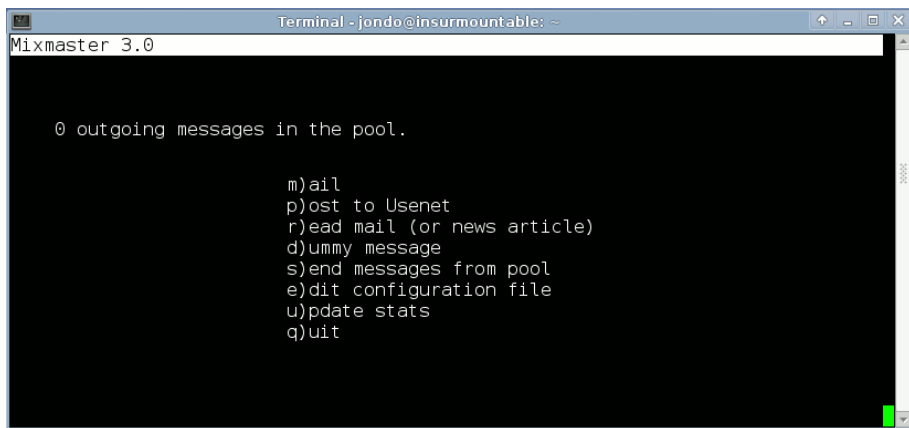


На трећој слици није приказано никакво име за *System mail name*, а ви овде не мењајте ништа, само оставите име како вам понуди и притисните *Ok*. Инсталација ће се наставити и убрзо завршити.

Сада је **Миксмастер** инсталиран, а из терминала се покреће командом:

```
mixmaster
```

Тада би екран требало да изгледа овако:



За почетак је потребна опција *m* како бисмо започели састављање нове поруке за слање. Унесите мејл адресу примаоца (па притисните Ентер), затим унесите наслов поруке (па опет Ентер), након чега би екран интерфејс требало да изгледа слично овоме:



Енкриптована електронска пошта

```

Terminal - jondo@insurmountable: ~
Mixmaster 3.0 - sending mail

c)chain: awxcnx,*,*,*                (reliability:  n/a )
r)edundancy:  2 copies

d)estination:  anonymous123@provider.tld
s)ubject:  Test message

pgp encry)ption:  no

e)dit message      f)ile      q)uit w/o sending
  
```

Сада имате опцију *edit message* за састављање текста поруке, па ће се притиском на *e* дугме на тастатури отворити предефинисан едитор на вашем систему унутар терминала (у већини случајева то је *vim*). Сада би требало да можете да испишете садржај поруке (уколико *vim* одбија да пише текст, прво притисните *i*, па почните да куцате текст). По завршетку писања текста притисните *Esc* на тастатури (*vim* се још неће затворити), па онда унесите команду *:wq* да бисте сачували исписани текст у *vim*-у и вратили се у претходни мени (друга слика у овом тексту) где би требало да вам се сада појави још једна опција у левом доњем делу терминала, а то је *mail message* коју сада бирамо и враћамо се у првобитни мени (прва слика у овом тексту). Пре него што притиснемо *s* и пошаљемо поруку, можемо приметити да се у дну терминала испод опција појавио и текст сличан овом *Chain: devurandom,foton1,inwtx,dizum*. То је уствари насумично одабран пут кроз друге римејлере кроз које ће се порука рутирати пре него што дође до одредишта тј. примаоца, чију сте мејл адресу имплицитно унели пре писања текста поруке. Сада можемо притиснути *s* да бисмо послали поруку, и изаћи из Миксмастера притиском на *q* опцију.

Поруке могу да путују дуго у зависности колики је проток кроз римејлере, па их немојте очекивати сваког часа убрзо после слања. Може се догодити да у току инсталације **Миксмастера** не буде покренута инсталација **Постфикса**, што значи да је већ инсталиран, па је онда потребно подесити га да ради са **Миксмастером** ако знате, а ако не, најбоље би било да **Постфикс** прво уклоните са система командом

Интернет мреже и комуникације

```
sudo apt-get purge postfix
```

па да тек онда инсталирате **Миксмастер** који ће у току своје инсталације покренути инсталацију **Постфикса**.

Оперативни системи који акценат стављају на анонимност, сигурност и приватност попут *Whonix*-а и *JonDo*-а, подразумевано долазе са претходно инсталираним и подешеним Миксмастер римејлер програмом спремним за коришћење.

Уколико желите да допринесете малој мрежи Миксмастер или Миксминион сервера, можете посветити један *Raspberry Pi* у ту сврху и на њему подесити римејлер сервер јер овакви сервери не захтевају боље рачунаре.

Наравно, ништа није 100% сигурно ни анонимно, па тако ни анонимни римејлери, али умногоме повећавају ниво безбедности, сигурности и анонимности за оне који то воле, или имају нешто да крију. Основни проблем код римејлера је мали број корисника што узрокује низак ниво анонимности јер корисници немају ту привилегију као код *Tor*-а да се коришћењем сервиса стапају са осталим корисницима чији се број мери у милионима насупротив римејлерима који због мале популарности пате од проблема „Малог света“.

За додатни софтвер аутор препоручује <http://goo.gl/jLKIWU> . Аутор такође препоручује видео са ****DEF CON***-а прошле године на тему деанонимизације анонимних порука <http://goo.gl/gzhLo7>

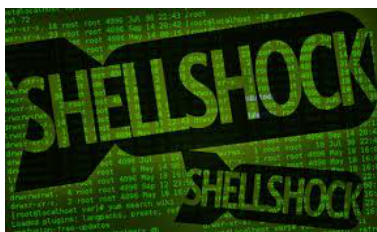




Shellshock

Аутор: Петар Симовић

Није прошло много после катастрофе зване „Heartbleed bug” (OpenSSL рањивости), а ево нас опет, остали смо затечени пред новом рањивошћу званом „Shellshock” и кодним именом „CVE-2014-6271”. Овај баг је заправо рањивост у BASH прогарму, програму који свакодневно користимо директо или индиректо, а да тога нисмо ни свесни.



BASH (*Bourne-again shell*) је стандарни програм отвореног кода стар преко двадесет година који је подаразумевано инсталиран уз већину **nix* оперативних система. Он је „*command-line based*” програм који служи за извршавање команди и скрипти од стране корисника или програма упућених оперативном систему односно машини коју корисник употребљава. Проблем се јавља у свим верзијама овог програма до верзије 4.3 (можете проверити своју верзију следећом командом:

```
bash -version
```

, а да ли је ваш BASH рањив можете проверити покретањем следеће команде након чега би требало да испише само други коментар, тј. у нашем примеру текст „Bash is not vulnerable”

```
env VAR='() { :; }; echo Bash is EXPLOITABLE!' bash -c "echo Bash is not vulnerable"
```

Уколико вам је BASH рањив, требало би само да ажурирате верзију командом за *debian* кориснике

```
sudo apt-get update && sudo apt-get install
```

Интернет мреже и комуникације

или уколико имате инсталиран *aptitude*:

```
aptitude update; aptitude safe-upgrade -y; reboot
```

или за *Red-Hat* кориснике:

```
sudo yum update bash
```

Као што сте можда могли да приметите из тест примера рањивости, проблем је у томе што *BASH* дозвољава у одређеним ситуацијама извршавање прослеђене му произвољне команде без икаквих провера. Наиме, познато је да се захтеви које веб сервери добијају обрађују локално на машини на којој је сервер. У таквим случајевима се обрада захтева препушта локалном *CLI*-у (*command-line interface*) који извршава захтев на машини. У већини случајева подразумевани *cli* је баш *BASH*. Специфично *Apache* сервери који имају „*mod_cgi*” и „*mod_cgid*” модуле омогућене или *openSSH* сервери који имају „*ssh*” омогућен су рањиви. Такође *DHCP* клијенти у уколико су повезани са злонамерним *DHCP* сервером, *qmail* сервери, и *IBM*-ов *restricted shell* за који је избачен *patch* : <http://goo.gl/AHS2gh> , и многи други. Занимљиво је рецимо да је и *OpenVPN* такође рањив (<http://goo.gl/GbOzj2>) и *VMWare* (<http://goo.gl/U2w9p7>), а *NGiNX* сервери су остали имуни и безбедни. Цела ситуација подсећа много на скорашњи фамозни „*Heartbleed bug*” баш због велике распрострањености програма који се ослањају на *BASH* и броја уређаја који га користе. И као и код „*Heartbleed bug*”-а коначна листа рањивих уређаја и софтвера се не зна јер је број уређаја који користе *Linux* у неком облику веома велики и вероватно се броји у хиљадама милиона.

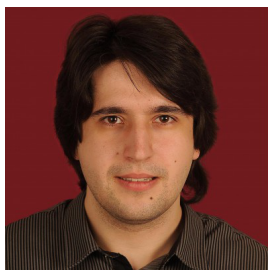
Мада оваква рањивост сама по себи није нова ни напредна, лако је решива, чак шта више проблем се лако решава у једном до два корака било на десктоп рачунарима, било на серверима, проблем је у распрострањености и веома широкој употреби. Проблем не могућности крпљења (*patching*) је у самој великој распрострањености уређаја тзв. „*embedded devices*” за које ажурирање не би било ни лако ни једноставно, а да не говоримо о „*firmware*”-у. Такви уређаји можда никада неће бити закрпљени, као што је и у овом тренутку после шест месеци није отклоњена „*Heartbleed*” рањивост са свих уређаја, и можда никад неће, јер је потребно да се у свакој фирми и кући у свету ажурира софтвер на сваком рачунару или уређају, што није баш изводљиво. Алтернатива ажурирању је замена уређаја за новије што је свакако финансијски идатак који се не одобрава баш олако.



OWASP SeraphimDROID

Интервју са Николом Милошевићем - ментором пројекта *OWASP Seraphimdroid*

Аутор: Стефан Ножинић



Да је заједница окупљена око слободног софтвера спремна да да свој одговор на тренутно стање поводом рачунарске сигурности, показују бројни пројекти отвореног кода, а међу њима се налази и *OWASP SeraphimDROID* - апликација за *Android* платформу, која служи као безбедоносни консултант корисника и која указује кориснику на разне сигурносне опасности о којима често не размишљамо у свакодневном коришћењу наших "паметних" телефона. ЛиБРЕ! је ступио у контакт са једним од аутора и ментором овог пројекта, Николом Милошевићем. Никола је био љубазан да нам одговори на неколико питања.

ЛиБРЕ!: Никола, објасни нам мало детаљније која је сврха *OWASP SeraphimDROID* пројекта и шта апликација ради?

Никола: *OWASP SeraphimDROID* је *open source* пројекат који првенствено треба кориснике да упозори на опасности по безбедност *Android* уређаја и приватност похрањених података корисника. Такође, друга сврха апликације је да едукује кориснике о ризицима угрожавања приватности и безбедности. Одређене функционалности на оба фронта су имплементирани у првој верзији, али има још доста места за даљи развој у оба правца.

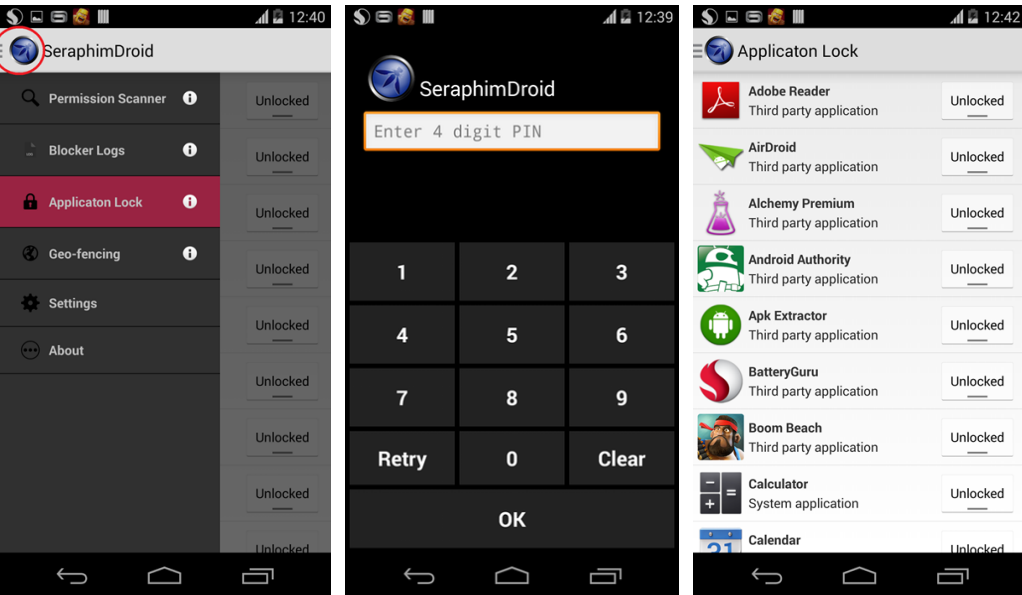
Апликација има:

- механизме да упозори кориснике уколико нека апликација без одобрења корисника покушава да зове премијум бројеве или шаље *SMS* поруке или извршава *USSD* команде,
- има имплементирану:
 - базичну заштиту од *phishing*-а,
 - скенер дозволу (пермисију), са објашњењима које дозволе на који начин могу бити

Мобилни кутак

злоупотребљене,

- даљинско брисање корисничких података у случају губитка телефона,
- лоцирање уређаја у случају губитка итд.



Пројекат је развијен под окриљем *OWASP* (енг.*Open Web Application Security Project*) фондације.

ЛиБРЕ!: Шта је тачно *OWASP* и која је твоја улога у пројекту?

Никола: *OWASP* је *Open Source* заједница. Скраћеница је за *Open Web Application Security Project*. Заједница је основана у Сједињеним Америчким Државама, али се од тад проширила на већину земаља света, где постоје локалне заједнице. Тренутно постоји триста десет локалних заједница у свету. Поред локалних заједница чији је циљ да пруже едукацију и место за дискусију везану за безбедност *software*-а, *OWASP* се бави и развојем *open source* алата за тестирање безбедности, заштиту, као и писање сигурносних стандарда. Највећи пројекти, које вреди поменути, јесу *OWASP Top 10* најчешћих безбедносних пропуста у *web* апликацијама и *OWASP ZAP*, који је постао стандардни алат при *penetration* тестовима (енг. *penetration* - продирање).



Што се моје улоге тиче, ја сам основао OWASP локалну заједницу у Србији пре отприлике три године, али пошто сам морао да се преселим у Манчестер због докторских студија, руковођење OWASP локалном заједницом у Србији је преузео Предраг Цујановић. Такође, ја сам вођа OWASP Seraphimdroid пројекта.



ЛиБРЕ!: Колико је временски трајао развој апликације и на какве проблеме је било могуће наићи током развоја?

Никола: Пројекат OWASP SeraphimDROID започет је пре око годину дана. Прва фаза је била прилично експериментална, где смо покушавали само да разрадиммо концепт и да докажемо себи да је заиста оствариво то што је замишљено. Било је и неких замисли од којих смо морали да одустанемо, јер смо схватили да нам је потребан *root access* уређају за одређену функционалност, што код већине уређаја није омогућено. Као са свим *open source* пројектима без неког озбиљнијег финансирања, проблем је што се пројекат ради у слободно време поред свих других послова и пропратних активности које појединци, који раде на пројекту, имају. Због тога је и развој овог пројекта био прилично дуг, али све се убрзало током *Google Summer of Code* програма, када смо добили финансирање од Google-а да један студент ради на пројекту три месеца пуно радно време. У том периоду апликација је редизајнирана и имплементиран је добар део функционалности које пројекат тренутно поседује.

ЛиБРЕ!: Био си ментор на *Google Summer of Code* програму, можеш ли нам дати мало више детаља о самом такмичењу, ко се може све пријавити, какво је стање са нашим студентима, на чему треба порадити, као и неке савете за оне који планирају да учествују на *GSoc* наредних година?

Никола: Заправо, *Google Summer of Code* није уопште такмичење већ пројекат којим Google покушава да помогне пројекте отвореног кода и којим се покушава подићи свест код студената о отвореном коду. Догађај има две фазе. У првој студенти предлажу пројекте/функционалности на основу одређених смерница које су менторске заједнице отвореног кода дале. Пожељно је пре слања пријаве и предлога пројекта контактирати с потенцијалним ментором и продискутовати о предлогу. У другој фази одабрани студенти

Мобилни кутак

уз менторство заједница отвореног кода раде на развоју предложених функционалности на пројекту. Требало би да студенти током три месеца раде пуно радно време, за шта од Google-а добију 5000\$, уколико су ментори задовољни урађеним послом. Ове године у оквиру *Google Summer of Code* програма учествовало је сто деведесет заједница отвореног кода које су менторисале хиљаду сто седамдесет и три (1173) студената.



Генерални проблем са студентима, како из Србије тако и из других земаља, јесте да нису довољно упознати са захтевима догађаја. Наиме, заједнице отвореног кода током *Google Summer of Code* програма добијају буџет за одређени број студената, који треба да раде на имплементацији нових функционалности пуно радно време три месеца, за шта ће бити лепо награђени (поменутих 5000\$), па тако и заједнице отвореног кода желе то што ефективније да искористе.

Такође, постоји поприлична конкуренција, па самим тим предлози који се шаљу у првој фази, треба да буду јако добри да би били прихваћени на крају. На неколико страна потребно је описати шта се жели имплементирати, како је то технички изводљиво, као и временски план, односно до кад ће сваки део имплементације бити готов.

Постоје и примери прихваћених извештаја код вероватно сваке заједнице која учествује дуже време, *OWASP* учествује већ пет година, стога је добро контактирати с потенцијалним ментором. У неким случајевима је добро допринети пројекту пре самог *Google Summer of Code* програма, јер ће на тај начин студенти боље разумети пројекат, па тиме написати и бољи предлог, а није ретко да се ментори одлучују за студенте са којима су претходно радили.

ЛИБРЕ!: Да ли планирате да наставите сарадњу са студентима који су радили на развоју *OWASP Seraphimandroid-a*?

Никола: Наравно! Морам да кажем да сам имао јако добро искуство са студентом који је радио ове године на пројекту на коме сам био ментор. Није било никаквих проблема, самим тим не постоји разлог да се сарадња не настави.

ЛИБРЕ!: Будући планови - шта да очекујемо?

Никола: Пројекат ће, надам се, даље наставити да се развија. Остало је одрадити неколико функционалности које недостају, попут провере сигурносних подешавања уређаја, које ће, надам се, ускоро бити имплементиране. Такође, тренутна заштита од *phishing-a* је најједноставнија могућа, па је треба унапредити. Тренутно, мислим да није



довољно рађено на едукацијском аспекту који апликација треба да има, па се могу очекивати нове функционалности на том фронту. Постоји и идеја о повезивању са неким од сервиса за проверу потписа апликација на *malware*, али видећемо да ли је то у *open source* окружењу оствариво. Такође, отворени смо за идеје корисника и потенцијалних нових програмера.

ЛиБРЕ!: *SeraphimDROID* је пројекат отвореног кода. Како му можемо помоћи?

Никола: Политика *OWASP*-а, самим тим и политика у овом пројекту, јесте да свако може да се прикључи. До сад није било много места за људе који нису програмери и свакако да је програмерска помоћ и даље остала најпотребнија. Међутим, откако је изашла прва верзија, наравно да има места и за друге професије (дизајн, маркетинг, итд.). Углавном, потребно је контактирати са мном и написати ми чиме бисте желели да допринесете. Идеје су такође добродошле. Мој *e-mail* можете наћи на пројектној страници (https://www.owasp.org/index.php/OWASP_SeraphimDroid_Project).

ЛиБРЕ!: *OWASP Seraphimdroid* апликација се бави заштитом приватности. Можеш ли дати додатне савете како се заштитити поред коришћења оваквих апликација?

Никола: Што се тиче безбедности, највећи број пропуста се дешава због људског фактора, односно, због незнања да нешто може бити опасно. Самим тим, најбоља заштита је едукација. Што се конкретно *Android* уређаја тиче, потребно је едуковати се везано за дозволе, шта која ради и како могу бити злоупотребљене. Такође, приступ *Wi-Fi* мрежама без заштите може бити јако опасан, јер било ко се може представити као *router* и читати ваш саобраћај, па тиме добити и приступ свим вашим налозима које користите у том тренутку. Наравно, могу да поновим и стандардну причу о лозинкама, које треба да буду јаке, односно да имају минимум седам или осам знакова, да садрже мала слова, велика слова, бројеве, специјалне знакове и крв девице (ок, шалим се за ово последње). Од губитка уређаја, па и података на њему, вероватно најбоља заштита је енкрипција. Потребно је осигурати се да су безбедносна подешавања подешена на одговарајући начин. Дobar *anti-malware software* је такође један од фактора који може да допринесе.

За крај, захваљујемо се Николи што нам је изашао у сусрет и представио нашим читаоцима пројекат *OWASP Seraphimdroid*. Наставићемо и убудуће да пратимо његов рад.

ЛИБРЕ!

Часопис о слободном софтверу

Расписује општи конкурс

за попуну редакције часописа

Часопис чека на вас!

Посетите нашу интернет страницу

<https://libre.lugons.org/index.php/pridruzi-se/>

и придружите се тиму са другачијим

погледом на IT технологије

