2 legjobb eljárás kéttényezös azonosításra és jelszó készítésére Linuxon

Írta: Konstantin Ryabitsev - Linux.com, 2017. május 24.

Ahogy azt az előző cikkben említettem, a webböngészők jelentik a legnagyobb és legjelentősebb támadási felületet a Linux munkaállomásnál. Már áttekintettünk néhány hasznos módszert, amit a Linux rendszergazdák alkalmazhatnak a böngésző feltörése hatásainak csökkentésére, úgymint Wayland-re frissítés, eltérő böngészők alkalmazása a munka és a fokozott biztonságú oldalakra, illetve Firejail-lal „sandbox” készítése a linuxos alkalmazások köré.

Most néhány további eljárást nézünk meg a kéttényezős azonosításra és jelszó készítésére, illetve használatára.

Használj Fido U2F-et weblap kéttényezős azonosítására

A Fido U2F egy szabványfejlesztés, amelynek célja a kéttényezős azonosítás alkalmazására és a hitelesítőadat-halászat megelőzésére szolgáló eljárások kidolgozása. A szokásos egyszeri jelszó (Regular OTP) hatástalan olyan esetekben, amikor a támadó képes elérni, hogy a jelszavadat és azonosítódat hivatalosnak beállított rosszindulatú oldalon add meg.

Az U2F protokoll tárolja az oldalazonosító adatokat egy USB-kulcson, ami megakadályozza, hogy véletlenül egy támadónak add meg egyszerre a jelszavadat és az egyszeri kulcsodat a hivatalostól eltérő weboldalon. A következő oldalon megnézheted , hogy mely szolgáltatók nyújtanak támogatást a Fido U2F számára.

*** dongleauth.info**

Tudd, hogy nem minden böngésző támogatja az U2F-képes hardver kulcsokat és ha „sandbox”-szal, vagy virtualizációval izolálod a böngésződet, akkor lehetséges, hogy komoly erőfeszítéseket igényel az USB-kulcs alkalmazásból USB-n keresztüli elérés engedélyezése.

Jelszókezelők

Egyedi, erős, véletlenszerűen létrehozott jelszó alkalmazása alapvető elvárás a csapatod összes tagjával szemben. Hitelesítőadat-lopás mindig előfordulhat – akár feltört számítógépen, ellopot adatbáziscsomagon, távoli behatoláson, vagy számos egyéb eszközön keresztül. Soha sem szabad azonos hitelesítőt használni eltérő oldalalkon.

Böngészőbe épített jelszókezelő

A böngészők rendelkeznek jelszómentő eljárással, ami elég biztonságos és szinkronizálható a készítő biztosította felhőtárolón keresztül, miközben az adatot a felhasználó által megadott jelszóval titkosítva tárolja. Ugyanakkor ennek az eljárásnak komoly hátulütői vannak:

1. a böngészők között nincs együttműködés;
2. nincs mód az azonosítók megosztására a csapat tagjai között.

Számos, jól támogatott, ingyenes, vagy olcsó jelszókezelő van, ami beépül többféle böngészőbe, platformfüggetlen és csoportos megosztást is ad (általában fizetett szolgáltatásként). Könnyen lehet megoldást találni keresőmotorok segítségével.

Önálló jelszókezelők

Az egyik legnagyobb hátránya a böngészőbe integrált jelszókezelőknek maga a tény, hogy az alkalmazás részét képezik, amik nagy valószínűséggel a behatolók célpontja. Ha emiatt aggódsz (és kell is), akkor két eltérő jelszókezelőt használj –

egyet a weblapokhoz, ami a böngészőbe integrált, és egyet, ami önálló alkalmazásként fut. Az utóbbi használható az érzékenyebb hitelesítő adatok számára, mint a rendszergazda-jelszó, adatbázis jelszavak, egyéb rendszerszintű hitelesítők stb.

Különösen hasznos lehet olyan eszköz, amivel megoszthatsz rendszergazdai hitelesítőket a csoportod más tagjaival (szerver rendszergazda-jelszó, ILO-jelszó, adatbázis adminisztrátori jelszó, rendszerbetöltő jelszó stb.)

Néhány eszköz segíthet neked ebben:

* KeePassX, ami a 2. verzióban jelentősen fejlődött a csoportmegosztást tekintve;

* Pass, ami szövegfájlt használ PGP titkosítással és Git-be integrált;

* Django-Pstore, ami GPG-t használva osztja meg a hitelesítőket rendszergazdák között.

* Hiera-Eyaml, ha a rendszeredben már használsz Puppet-et, a titkosított Hiera adattároló részeként, hasznos módja lehet a szerver-, vagy szolgáltatás-hitelesítők követésének.

A sorozat következő, egyben záró cikkében azzal foglalkozom, hogyan helyezzük biztonságba az SSH és PGP privát kulcsokat – egy további alapvető lépése a Linux rendszer-adminisztrátori munkaállomás lehetséges behatolók elleni megvédésének.