

مدیریت رمز ها و کلید های محرمانه

سویل - نیما

بهار ۱۴۰۰

مقدمه

ما در طبیعت برای شناخت هر پدیده ای ابتدا یک الگوی ذهنی میسازیم تا در برخوردهای بعدی این پدیده برای ما یک الگوی شناخته شده باشد، سپس در گذر زمان و تکرر تقابل با این پدیده منجر به تکمیل اطلاعات ذهنی ما میشود و عقبه رشته اطلاعات ما به تدریج تکمیل تر میشود. به عبارتی منحنی تجربه منجر به تکمیل و اصلاح اطلاعات ما میشود و اینچنین ما الگویی شناخته شده را به تاریخچه خود می افزاییم ، وجه اشتراک بین همنوعان (پروتکلی ارتباطی مانند زبان) بین ما منجر به اشتراک این اطلاعات میشود و این الگوی شناخته شده تبدیل به یک الگوی جمعی میشود.

این الگوهای مرجع و مشترک منجر میشود ما قدرت تشخیص و تمیز دادن را پیدا بکنیم ، در ابتدا این مساله بسیار ساده تر بوده است زیرا پیچیدگی و تنوع در نمودار زمان همیشه صعودی بوده است ، تشخیص هم نوع، تشخیص دشمن یکی از کلاسیک ترین های تاریخ بشری است. در طول زمان طبیعتا معایبی در تشخیص الگو ها پیدا شده است و این منجر به اصلاح و پیشرفت روش های تشخیص شده است ، در طرف مقابل پدیده ها هم پیچیده و هوشمند تر شده اند.

در زمان های نچندان دور برای شناخت یک فرد یک اسم از او کافی بود، بعد ها اسم خانوادگی دیگری کنار این اسم آمد تا به شناخت کمک کند و امروز گرچه در دایره کوچک نام و نام خانوادگی شما کارساز است ولی شما بدون کد ملی قابل شناسایی نیستید. همیشه تصدیق این الگو ها برای بشر پر از چالش بوده است، برای تصدیق اصالت این الگوها همیشه راه حل هایی داشته است تا از اصالت الگو اطمینان حاصل پیدا کند.

اینجا بوده است که چیزی که ما در حال حاضر از آن به عنوان گذرواژه یا پسورد یا پسفریز نام میبریم پدید آمده است و روز به روز روش ها و مدل های جدیدی از آن اصلاح و تکامل یافته است. با نگاهی به طبیعت و پیشینه مان چنان به نظر می آید که این مدل ریشه در تاریخ طبیعی داشته است و پا به پای ما تا به امروز تکامل یافته است. امروز ما در جهان حال حاضر هر کدام شناسه ای داریم (برای نمونه : شناسه توییتر) که کاملاً یکتا و منحصر بفرد است و حتماً برای تصدیق اصالت آن به رمز عبور (password) داریم . با کوچکترین نگاهی اهمیت پسورد یا گذرواژه را در جهان حاضر به سادگی میتوان درک کرد.

در حال حاضر پسورد عموماً رشته ای از کاراکترها ، اعداد و علائم است که هرچقدر پیچیده تر (تنوع عدد ، کاراکتر و علائم + طول بیشتر) باشد منجر به امنیت بیشتری میشود، معمولاً امروزه همه ما انسانها دستکم چندین پسورد داریم و در مدیریت آنها دچار مشکل هستیم. راه و روش های ما برای مدیریت پسوردها از نوشتن و مخفی کردن در یک گوشه ای از کاغذ گرفته تا حفظ کردن آن همه دارای ایرادهای اساسی است، لذا برای بهینه سازی این مساله ابزارهای مدیریت پسورد توسعه داده شده اند تا ما را در مدیریت پسوردهایمان یاری کنند.

- سویل

اهمیت استفاده از رمز های متفاوت و پیچیده

موضوع استفاده از رمز های پیچیده، هم برای سازمان ها و هم برای افراد از اهمیت بالایی برخوردار است. در بیشتر موارد، قواعد انتخاب رمز از سمت سازمان ها و سرویس ها (وبسایت ها، اپلیکیشن ها و ...) مشخص می شوند و کاربران آن سازمان ها، وادار به رعایت این قواعد میشوند. اما از طرفی، انتخاب رمز پیچیده باید به عنوان یک ذهنیت، برای افراد نیز جا بیافتد. زیرا یکی از بزرگترین تهدید ها، حملات به زیرساخت سازمان ها هست و اگر قواعد پیچیدگی رمز مشخص شده توسط سازمان، به اندازه کافی کامل و قوی نباشند، همچنان اطلاعات شخصی شما در خطر خواهند بود.

حملات متفاوتی می توانند باعث فاش شدن رمز ها و کلید های محرمانه شما بشوند. در ادامه به صورت کلی، مروری بر روش های پیدا کردن یا حدس زدن رمز های توسط بازیگر های مخرب در دنیای اینترنت می پردازیم. دقت داشته باشید که هیچ یک از موارد اشاره شده، قطعی نیستند و همچنان عامل «شانس» در این روش ها اهمیت بسیار بالایی دارد و اگر در انتهای این مطلب تصمیم گرفتید که «رمز من به مقداری پیچیده هست که هیچکس تابحال موفق به حدس زدن آن نشده است»، حتما اینکه «شاید شما خوش شانس بوده اید» را در نظر بگیرید!

کشف رمز با حدس زدن بخش ها

بسیار از افراد طبق عادت، هنگام انتخاب رمز، به اطلاعات محرمانه (!) زندگی خودشان یا افراد نزدیکشان فکر میکنند و در فرایند ساخت یک رشته بعنوان رمز، از اطلاعاتی مانند تاریخ تولد، نام شهر، نام حیوان خانگی، نام افراد نزدیک، شماره

تلفن ها، خوراکی مورد علاقه، نام مدرسه یا مکان های به یاد ماندنی، جملات و شعار های مربوط به گروه های همفکر و مواردی از این قبیل استفاده میکنند. معمولا افراد با ترکیب کردن چند مورد از موارد اشاره شده اقدام به ساخت رمز می کنند. برخی افراد پا را یک قدم فراتر می گذارند و با تغییر دادن برخی کاراکتر ها در رشته تولید شده، اقدام به پیچیده-تر کردن رمز می کنند؛ شاید هم صرفا برای رسیدن به قواعد پیچیدگی تاکید شده توسط سازمان، نیاز به ایجاد این تغییرات دارند.

به مثال زیر توجه کنید:

رمز ساخته شده در ذهن کاربر:
lovepoppy14121994

رمز تغییر یافته برای اضافه کردن پیچیدگی:
L0v3poppy14121994

همونطور که میبینید، رمز ساخته شده شامل ۳ بخش مختلف هست. بخش اول، کلمه «love» هست که در ادامه با استفاده از قواعد 1337، تبدیل به «L0v3» شده است. بخش دوم، احتمالا نام حیوان خانگی فرد «poppy» هست و بخش سوم هم احتمالا تاریخ تولد خود فرد یا یکی از افراد نزدیک به فرد است «14121994».

نکته قابل توجه در ساختار این رمز این مورد هست که، این رمز احتمالا از قواعد و محدودیت های پیچیدگی رمز بیشتر از ۹۰ درصد سازمان ها با موفقیت عبور میکند. بیشتر از ۸ کاراکتر، شامل اعداد و شامل حروف بزرگ و کوچک انگلیسی می باشد. تنها موردی که شاید در برخی سازمان ها بعلاوه قواعد بالا، مورد بررسی قرار میگیرد، حضور کاراکتر های خاص، مثل (!, @, #, \$, %, ^) می باشد که در این

مورد، رمز ساخته شده توسط فرد فرضی ما، نیاز به یک تغییر کوچک دیگر خواهد داشت.

حال اجازه دهید با هم بررسی کنیم که چطور یک بازیگر مشکوک یا مخرب، می تواند رمز شما را حدس بزند.

رایج ترین نوع حملات هدفمند به افراد، استفاده از روشی معروف به Brute Force می باشد. در این نوع حمله، هکر سعی میکند با تولید یک لیست از تمام رمز های ممکن و تست کردن تک تک آنها روی وبسایت یا اپلیکیشن مورد نظر، رمز شما را حدس بزند. به این لیست از رمز ها، اصطلاحاً دیکشنری (Dictionary) هم گفته می شود. اگر هکر یک فرد خاص را هدف قرار داده باشد، هکر می تواند با تولید یک دیکشنری مخصوص برای فرد مورد نظر، حمله خود را تا حد ممکن بهینه بکند. اگر بخواهیم خط فکری هکر را با هم بررسی کنیم، چیزی شبیه به این خواهد بود.

هکر فرد را می شناسد و اطلاعاتی مثل صفحات شبکه های اجتماعی فرد، اینکه حیوان خانگی دارد یا خیر، شهر تولد، گروه های همفکر فرد و ... را داراست. هکر همچنین می تواند با تحت نظر گرفتن و استخراج داده از شبکه های اجتماعی فرد، اطلاعاتی مثل تاریخ/محل تولد، نام حیوان خانگی، نام و تاریخ تولد افراد نزدیک به فرد مورد نظر و بسیاری از اطلاعات دیگر را نیز کشف بکند. حال هکر می تواند یک دیکشنری با ترکیبی از این اطلاعات تولید کند. همچنین با توجه به اینکه معمولاً اگر افراد از قواعدی مثل 1337 برای اضافه کردن پیچیدگی به رمز خود استفاده کرده باشند، این موارد نیز کاملاً قابل شبیه سازی هستند و می توان ترکیبات و جایگشت های مختلف شامل آنها را نیز به دیکشنری اضافه کرد. این دیکشنری، در نهایت شامل ترکیبات و جایگشت های مختلف این اطلاعات خواهد بود. ترکیبات تک کلمه ای تا چند کلمه ای.

می توانیم ببینیم که با در دست داشتن اطلاعات کمی از زندگی شخصی افراد، می تواند در خیلی از موارد، رمز های محرمانه آنها را حدس زد و به اطلاعات شخصی شان دسترسی پیدا کرد.

حتی اگر با خودتان فکر میکنید که رمز عبور شما شامل هیچ یک از این اطلاعات شخصی نیست و «به ذهن هیچکس نخواهد رسید»، این مورد را هم در نظر بگیرید که دیکشنری های بسیار بسیار بزرگ (بیش از ده ها گیگابایت) وجود دارند که با در دست داشتن کامپیوتر های نسبتا قدرتمند، می توان تمام موارد موجود در آنها را روی اکانت افراد مختلف امتحان کرد تا در نهایت به رمز فرد رسید. این دیکشنری ها برای مثال شامل «تمام» رشته های ممکن تا ۱۰ کاراکتر یا بیشتر هستند. پس مستقل از اینکه چه کلمات و بخش هایی برای رمزتان انتخاب کرده اید، اگر طولش کمتر از یک حد مشخص باشد، «حتما» قابل حدس زدن خواهد بود.

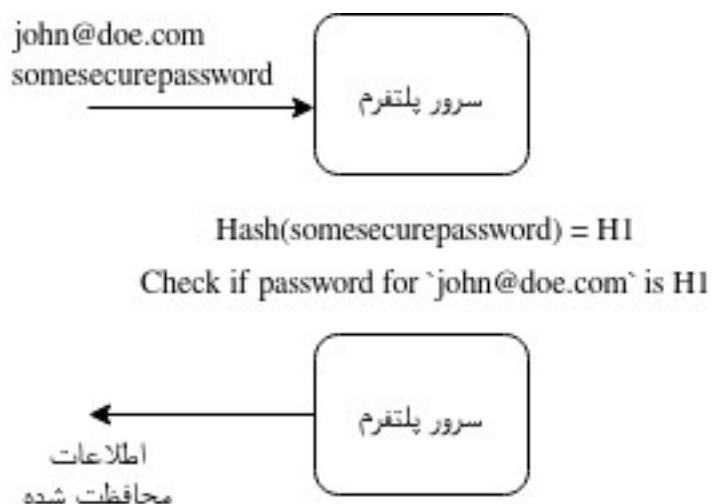
اهمیت استفاده از رمز های متفاوت

موضوع دیگر که اهمیت بسیار زیادی دارد، استفاده از رمز های متفاوت در پلتفرم ها و وبسایت های مختلف است.

بگذارید در ابتدا، اشاره ای به استاندارد ذخیره سازی رمز ها در پلتفرم های آنلاین بکنیم. زمانی که شما یک اکانت در یک پلتفرم می سازید و یک رمز برای محافظت از آن، مشخص می کنید، این رمز باید به نحوی در پایگاه داده پلتفرم مورد نظر ذخیره بشود؛ تا در موارد بعدی که شما اقدام به احراز هویت می کنید، رمز وارد شده توسط شما، با رمزی که بار اول مشخص کردید، «مقایسه» بشود و در صورت مطابقت، سطح دسترسی شما افزایش پیدا بکند.

نکته مهم، نحوه ذخیره سازی و انجام مقایسه در سمت سرور پلتفرم های آنلاین می باشد. بصورت استاندارد، رمز کاربر ها باید بصورت هَش (hash) شده، در

پایگاه داده ذخیره بشود و همچنین مقایسه رمز ها در مرحله احراز هویت، باید مقایسه هَش رمز وارد شده باشد. این موضوع، زمانی اهمیت خود را نشان می دهد که پلتفرم مورد نظر، تحت حمله ای اطلاعات پایگاه داده کاربرانش فاش شود و به دست هکر ها بیافتد. برای مثال، فرض کنیم روش احراز هویت شما به صورت «ایمیل و رمز» باشد. اگر رمز کاربران در پایگاه داده این پلتفرم بصورت هَش نشده ذخیره شده باشند، هکر ها با در دست داشتن ایمیل و رمز عبور شما، می توانند در اکانت شما در پلتفرم به سادگی لاگین بکنند و به اطلاعات شخصی شما دسترسی پیدا کنند یا در موارد دیگر اقدام به «جعل هویت» شما بکنند. اما اگر رمز ها بصورت هَش شده ذخیره شده باشند، هکر ها قادر به بازگردانی هَش نخواهند بود. به تصویر زیر توجه کنید:



پس اگر رمز هَش شده را به عنوان، رمز وارد کنیم، سرور پلتفرم به ما اجازه دسترسی نخواهد داد. چون هر ورودی که بعنوان رمز وارد شود، ابتدا هَش شده و بعد مقایسه می شود و با هَش کردن رمز هَش شده، رشته بی استفاده جدیدی تولید خواهد شد که هیچ ارتباطی با رمز کاربر ندارد.

اما همچنان این موارد در خیلی از پلتفرم ها رعایت نمی شود و در صورت فاش شدن اطلاعات پایگاه های داده این سازمان ها، اطلاعات شخصی شما در خطر خواهد بود. البته باید احتمال ذخیره و سوء استفاده از رمز ها بصورت هَش نشده توسط خود سازمان ها و پلتفرم ها هم در نظر گرفته بشود. چون در صورت یکی بودن رمز وارد شده با رمز اکانت ایمیل یا ترکیب ایمیل و رمز یکسان در پلتفرم های دیگر، همچنان امکان سوء استفاده وجود دارد.

حال تصور کنید که هکری رمز فردی را طی حمله ای به دست آورده است. اگر این فرد مورد حمله، از یک رمز یکسان برای همه یا بیشتر اکانت هایش در پلتفرم های مختلف استفاده کرده باشد، هکر در واقع دسترسی به «تمام» اکانت های این فرد خواهد داشت! برای مثال، اگر رمز اکانت شما در یک وبسایت نامعتبر و گذری در اینترنت مشابه رمز عبور ایمیل اصلی شما باشد، اگر به وبسایت نامعتبر حمله ای صورت بگیرد، شما به سادگی اطلاعات دسترسی به ایمیل شخصی خودتان را نیز به دست هکر ها داده اید.

پس دقت کنید که استفاده از رمز های یکسان، حتی برای چند وبسایت محدود هم ایده خوبی نیست و حتما باید برای تمام اکانت هایی که در پلتفرم های مختلف می سازید، از رمز های عبور متفاوت استفاده کنید.

دردسر تولید رمز و نگهداری امن رمزها

خب با توجه به نکات اشاره شده در بخش های قبلی، احتمالا به این فکر میکنید که دردسر تولید رمز های پیچیده و متفاوت برای ساخت هر اکانت در یک پلتفرم آنلاین، انقدر زیاد است که «به دردسرش نمی ارزد!». اما خب مثل همه کارهای تکراری و طاقت فرسای دیگر در دنیای دیجیتال، حتما ابزاری برای این کار هم وجود دارد که این مسأله را برای شما حل کرده و تولید و نگهداری رمز ها را برای شما ساده می کند.

در دنیای اینترنت، می توانیم ذخیره سازی اطلاعات محرمانه را به دو نوع «سرد» و «گرم» دسته بندی کنیم. زمانی که از ذخیره سازی «سرد» صحبت میکنیم، در واقع به ذخیره سازی این اطلاعات سمت خود کاربران و با اختیار صددرصدی کاربر اشاره میکنیم. در مقابل، ذخیره سازی «گرم» نوعی از ذخیره سازی هست که کاربر برای نگهداری این اطلاعات محرمانه به یک سازمان/نفر دیگر «اعتماد» میکند و وظیفه حفظ امنیت اطلاعات را بر عهده این سازمان میگذارد.

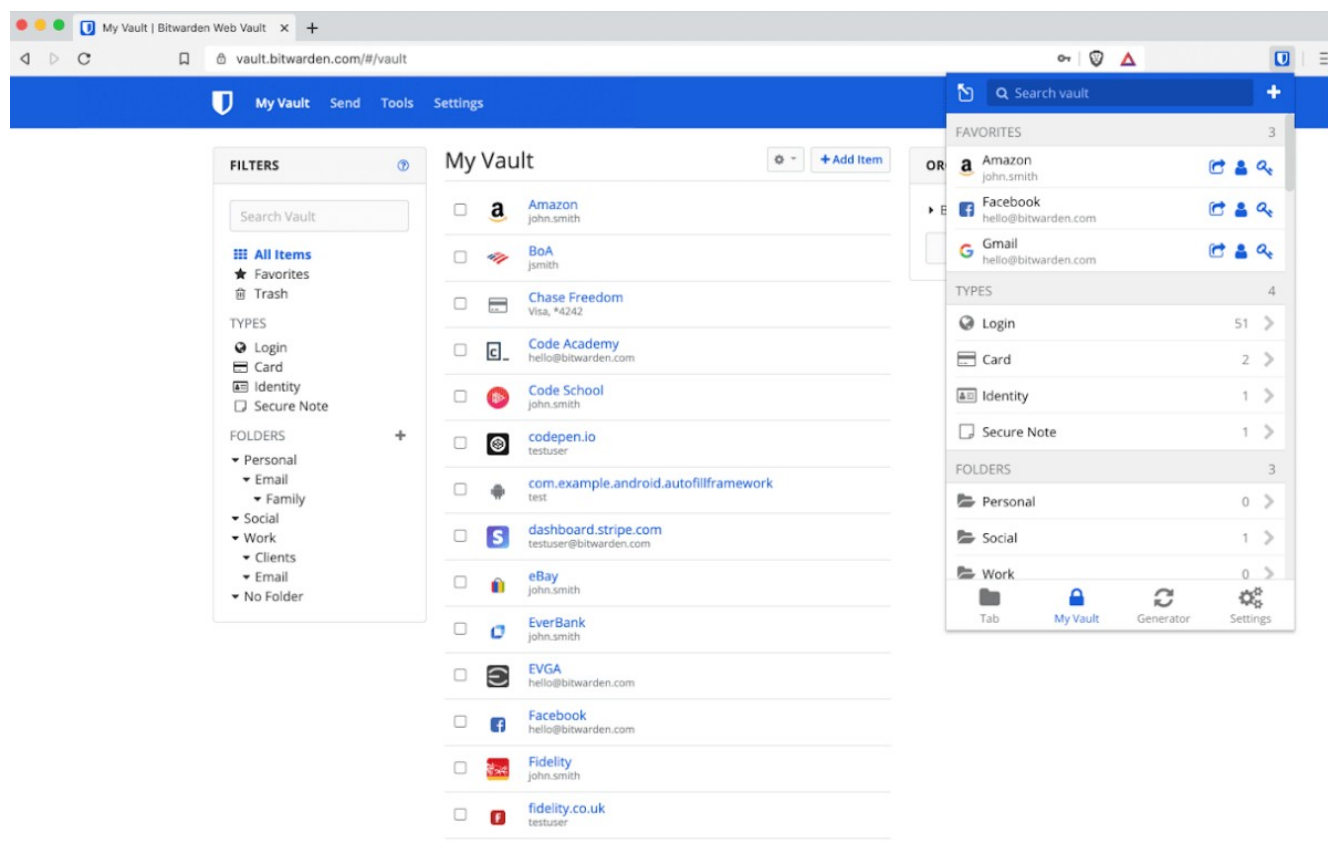
مشخصا نگهداری امن اطلاعات محرمانه به صورت «سرد» نیازمند سطحی از دانش فردی و همچنین رعایت نکاتی به صورت روزمره است که در این باعث می شود افراد، در خیلی از موارد، به یک فرد دیگر اعتماد کرده و دردسر نگهداری این اطلاعات محرمانه را بر عهده آنها می گذارند. در ادامه به معرفی چند مورد از نرم افزار های مدیریت و تولید رمز ها اشاره می کنیم.

این نرم افزار ها در تولید رمز های متفاوت و قوی و پیچیده برای شما کار را ساده می کنند و با ارائه یک رابط آسان، به شما قابلیت تولید و نگهداری رمز برای پلتفرم های مختلف را می دهند. بسیاری از این نرم افزار ها، همگام سازی خیلی خوبی با مرورگر ها دارند که کار را برای کاربر بسیار ساده می کند و همچنین در برخی موارد، قابلیت به اشتراک گذاری این رمز ها میان همکاران یک سازمان را در اختیار کاربر میگذارند.

نرم افزار های تحت وب برای مدیریت رمز ها

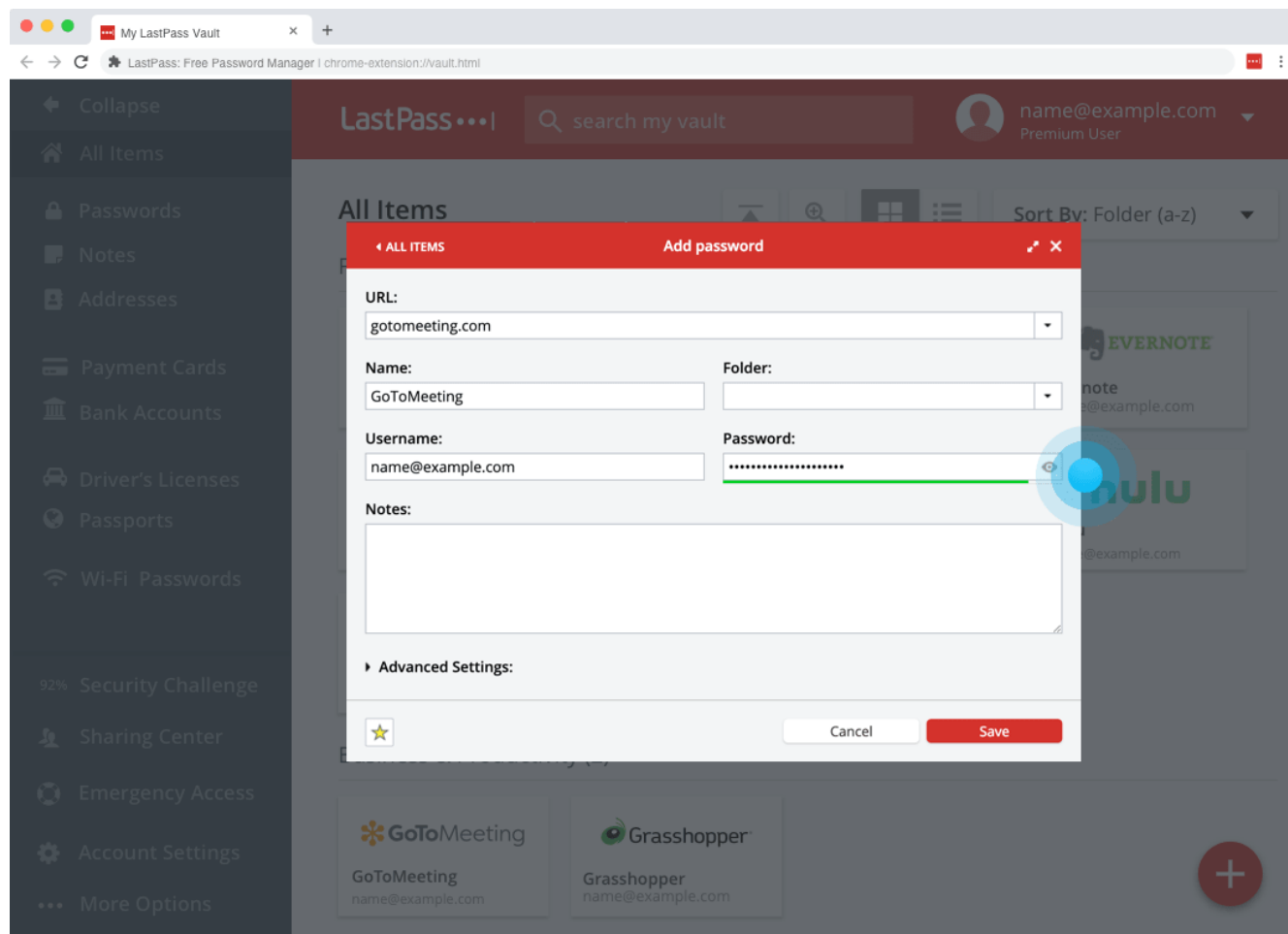
Bitwarden (bitwarden.com)

بیت-واردن یک پلتفرم آنلاین ارائه خدمات مدیریت رمز هست که در عین حال بصورت یک نرم افزار متن-باز نیز در اختیار کاربران قرار دارد و اگر علاقه ای به استفاده و اعتماد به سرویس آنلاین آنها ندارید، می توانید برای خودتان بصورت شخصی راه اندازی کنید و اطلاعات خودتان را روی سرور / کامپیوتر های شخصی خودتان میزبانی کنید.



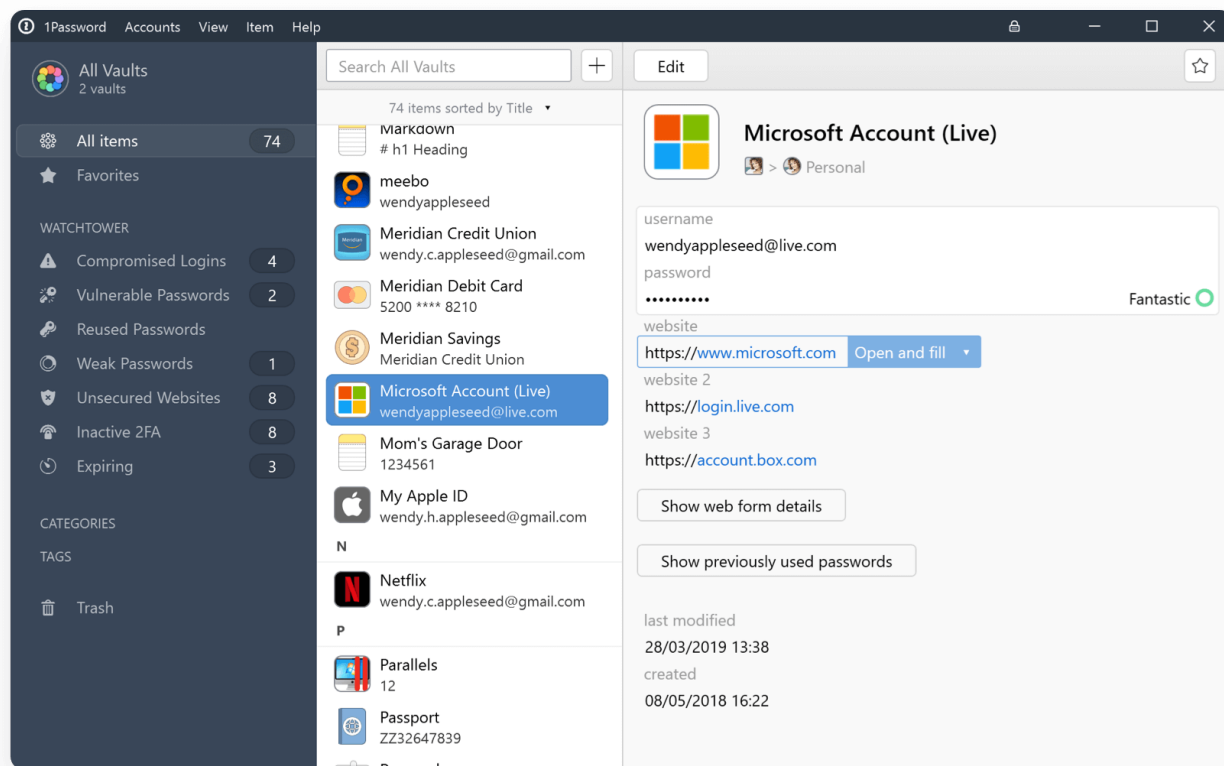
LastPass (lastpass.com)

همانند بیت-واردن، لست-پس نیز یک پلتفرم آنلاین خدمات مدیریت رمز هست و سعی کرده با ارائه افزونه های همگام سازی برای مرورگر های مختلف، تجربه کاربری بسیار ساده ای برای کاربران فراهم کند.



1Password (1password.com)

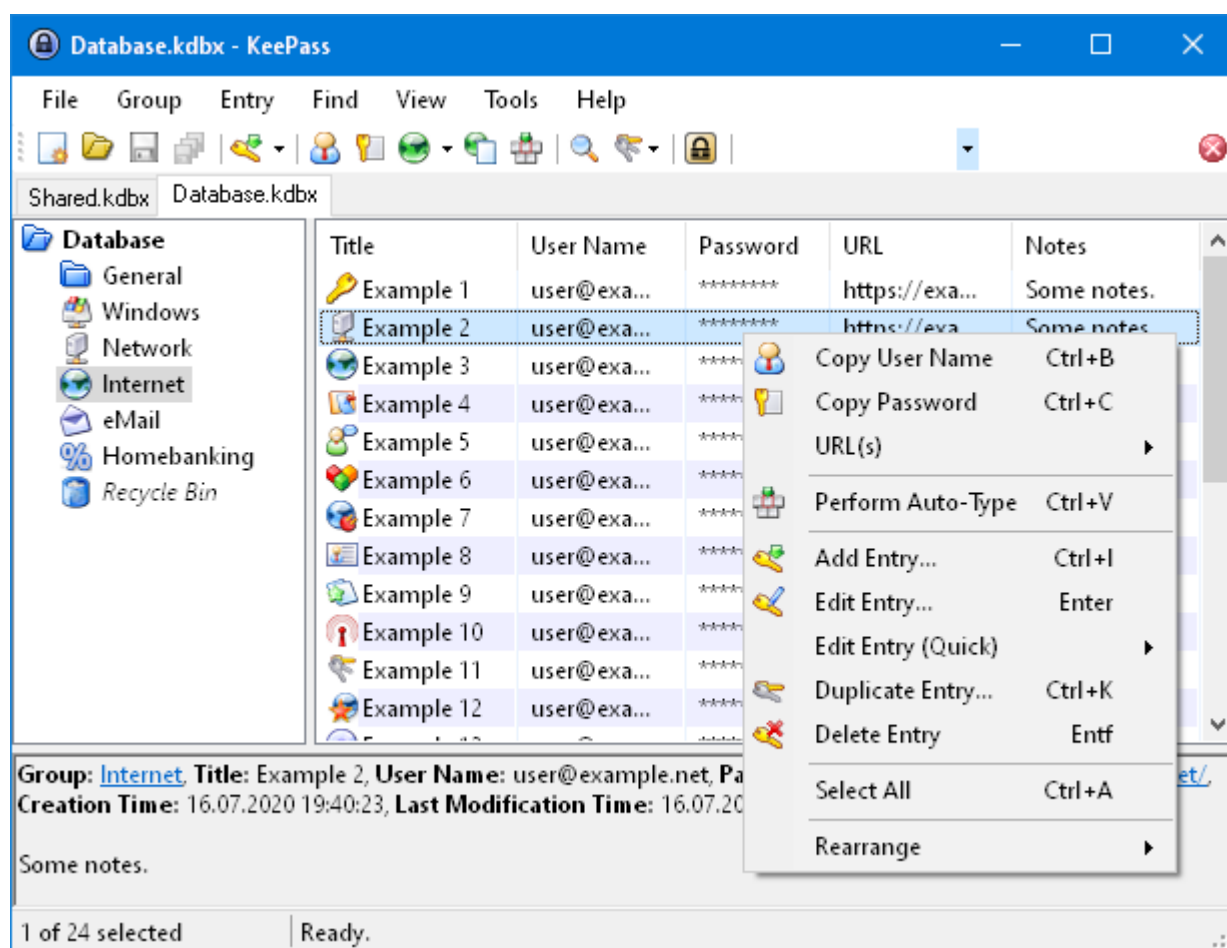
این سرویس هم بسیار مشابه به سرویس لست-پس عمل میکند و می توانید با نصب افزونه های مرورگر برای این سرویس، از تجربه کاربری بسیار ساده ای بهره مند شوید.



نرم افزار های تحت سیستم عامل

KeePass (keepass.info)

این نرم افزار که بر روی تمامی سیستم عامل های موجود قابل استفاده است و کلاینت های متفاوتی دارد، جزو بهترین نرم افزار های مدیریت رمز هست و همچنین بصورت کاملا متن-باز برای عموم قابل دسترسی می باشد. هم بر روی تلفن های همراه و هم بر روی کامپیوتر های شخصی، می توانید از این نرم افزار بهره ببرید.



pass (passwordstore.org)

در انتها، می خواهیم به معرفی نرم افزار pass بپردازیم که با پیروی از فلسفه unix، سعی در انجام «یک» کار به درست ترین روش دارد. این نرم افزار نیز بصورت متن-باز برای عموم قابل دسترسی است.

نکته بسیار جالب در مورد این نرم افزار، خلاقیت آنها در بکارگیری ابزار git برای مدیریت رمز ها می باشد. هنگام استفاده از این نرم افزار، شما می توانید با استفاده از git، تاریخچه تمام تغییراتی که بروی رمز هایتان می دهید را ذخیره کنید و همچنین اگر علاقه داشته باشید، میتوانید با نگهداری این تاریخچه برروی یک سرور git (شخصی یا وابسته به ارگان - مثل github یا gitlab) در آینده روی ماشین ها و کامپیوتر های دیگر، به سادگی به رمز های خود دسترسی پیدا کنید. نکته جالب دیگر در مورد این نرم افزار، استفاده از رمزنگاری کلید-عمومی برای محافظت از رمز ها می باشد.

در مستند سازی این نرم افزار، اشاره شده که «هر رمز» بصورت «یک فایل رمزنگاری شده» ذخیره می شود و کلید عمومی و خصوصی استفاده شده برای رمزنگاری این فایل ها نیز در اختیار شماست و می توانید تنها با داشتن کلید خصوصی خود، به رمز های خود دسترسی پیدا کنید. در این صورت، اگر کلید خصوصی مورد استفاده و فایل رمزنگاری شده در کنار هم نگهداری نشوند، حتی با هک شدن و فاش شدن فایل های رمز های شما، همچنان رمز های شما فاش نخواهند شد، زیرا هر فایل، با استفاده از کلید عمومی شما، رمزنگاری شده است.

این نرم افزار کاملاً با gnupg همگام سازی شده و می توانید با کلید های مدیریت شده توسط gnupg، فایل های رمز خود را، رمزنگاری کنید.

مراحل استفاده روی سیستم عامل های (*nix):

- ساخت کلید های خصوصی و عمومی (در صورت نداشتن کلید)

`gpg --full-generate-key`

```
[nima@thinkpad ~]$ gpg --full-generate-key
gpg (GnuPG) 2.2.25; Copyright (C) 2020 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

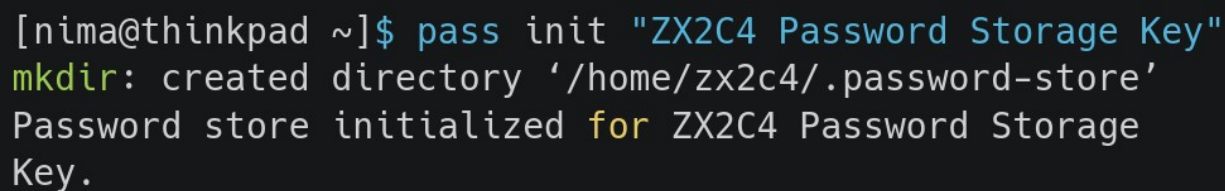
Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
 (14) Existing key from card
Your selection?
```

- توجه مهم به سطح دسترسی مسیر کلید ها

```
[nima@thinkpad ~]$ # Make sure the folder+contents belong to you
[nima@thinkpad ~]$ chown -R $(whoami) ~/.gnupg/
[nima@thinkpad ~]$ find ~/.gnupg -type f -exec chmod 600 {} \;
[nima@thinkpad ~]$ find ~/.gnupg -type d -exec chmod 700 {} \;
```


- راه اندازی نرم افزار pass با استفاده از کلید ساخته شده

pass init GPG KEY ID

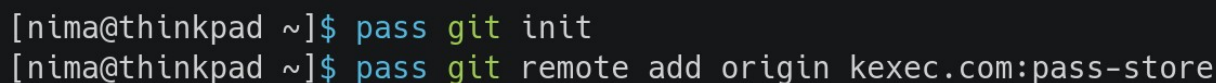


```
[nima@thinkpad ~]$ pass init "ZX2C4 Password Storage Key"
mkdir: created directory '/home/zx2c4/.password-store'
Password store initialized for ZX2C4 Password Storage
Key.
```

- راه اندازی ساختار git برای نرم افزار pass

pass git init

pass git remote add origin kexec.com:pass-store



```
[nima@thinkpad ~]$ pass git init
[nima@thinkpad ~]$ pass git remote add origin kexec.com:pass-store
```

وبسایت:

passwordstore.org

صفحه man:

<https://git.zx2c4.com/password-store/about>