این ماهنامه با حمایت مادی و معنوی اداره کل امور فرهنگی دانشگاه اصفهان چاپ و منتشر شده است.

نشریه علمی روز صفرم
شماره ۱۹ - مهر ۱۴۰۰

صاحب امتیاز:
شاخه دانشجویی انجمن رمز ایران در دانشگاه اصفهان

سردبیر:
محمد آقائی

مدیر مسئول:
الهه رهبران

طراح جلد و صفحه‌آرا:
نوریه سادات مدنیان
محمد آقائی

هیئت تحریریه:
زهرا اشرفی
غزال محسنی
فرید احمدپور مبارکه
فاطمه نشاط‌دوست

اخبار:
سروش ذوالفقاری

ویراستار:
الهه رهبران

t.me/SBISC

SBISC.UI.AC.IR

t.me/CTFPREP

twitter.com/SBISC1

instagram.com/SBISC_UI

درباره انجمن:

شاخه دانشجویی انجمن رمز ایران در دانشگاه اصفهان از سال ۱۳۸۶ فعالیت خود را پیرامون مباحث مرتبط با امنیت اطلاعات آغاز کرد. این انجمن که هم‌اکنون یازده دوره از آغاز فعالیت آن می‌گذرد، تصمیم به انتشار نشریه‌ای با عنوان "**روز صفرم**" گرفته است تا از این طریق بتواند دانش امنیتی در فضای سایبر را به مخاطبان خود منتقل کند.این نشریه به صورت ماهانه و از اردیبهشت ۹۸ منتشر شده است.

کتاب، سفینه‌ای است که اقیانوس بیکران زمان را درمی‌نوردد.

فرانسیس بیکن

هفته کتاب و کتاب‌خوانی مبارک

نهان‌نگاری

Steganography

زهرا اشرفی
ashrafi.zahra81@gmail.com

نهان‌نگاری چیست؟

نهان‌نگاری (استگانوگرافی)، روشی برای حفاظت از داده‌ها و علم مخفی کردن پیام‌ها است که برگرفته از دو کلمه‌ی یونانی Steganos به معنای پوشیده و Graphia به معنای نوشتن می‌باشد. هدف این علم، پنهان کردن ارتباط به وسیله‌ی قرار دادن پیام در یک رسانه‌ی پوششی است به گونه‌ای که کمترین تغییر قابل کشف در رسانه‌ی پوششی ایجاد شود و نتوان موجودیت پیام پنهان در رسانه را آشکار ساخت، به طریقی که هیچ کس جز فرستنده و گیرنده‌ی پیام به وجود پیام مخفی شک نکند.

اولین استفاده‌های پنهان‌نگاری به یونان باستان بازمی‌گردد که توسط یک مورخ یونانی به نام هرودوت به ثبت رسیده است. سربازان یونانی برای انتقال پیام، سر بردگان را می‌تراشیدند و روی پوست سر آنان نقشه یا پیام را خالکوبی می‌کردند و مدتی بعد که موی سر این بردگان بلند می‌شد و روی پیام را می‌پوشاند، آن‌ها می‌توانستند به راحتی از میان سرزمین‌ها و اراضی مربوط به دشمن عبور کنند. در مقصد با تراشیدن مجدد موی سر آنان پیام استخراج می‌شد.

تفاوت نهان‌نگاری و رمزنگاری چیست؟

در رمزنگاری، مهاجم از وجود پیام بر روی یک کانال ارتباطی خبر دارد ولی چون پیام رمز شده است، نمی‌تواند به محتویات آن دسترسی پیدا کند. اما در نهان‌نگاری، مهاجم از وجود اطلاعات در یک پوشش (تصویر، صوت، متن و یا ویدیو) اطلاعی ندارد.

در واقع در رمزنگاری، هدف اختفاء محتویات پیام است و نه به‌طور کلی وجود پیام. در این روش، اطلاعات به رمزمتن (text cipher) تبدیل می‌شود که بدون کلید رمزگشای آن قابل فهمیدن نیست. بنابراین اگر کسی این پیام رمزگذاری شده را رهگیری کند، به راحتی می‌بیند که در آن نوعی رمزگذاری اعمال شده است. از طرف دیگر هدف نهان‌نگاری، مخفی کردن هر گونه نشانه‌ای از وجود پیام است. در این روش فرمت اطلاعات تغییر نمی‌کند اما وجود پیام پنهان می‌شود.

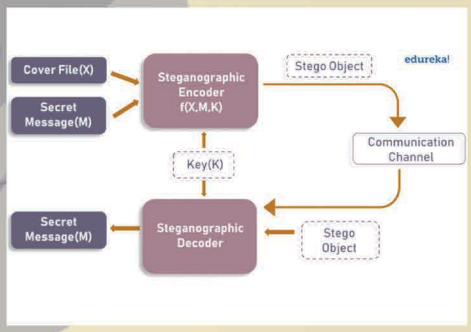
بنابراین می‌توان گفت برای انتقال اطلاعات محرمانه، نهان‌نگاری روش خوبی نیست. چرا که در صورت آشکار شدن مخفی بودن پیام، استخراج پیام مخفی راحت‌تر است.

اجزا یک سیستم نهان‌نگاری

- یک سیستم نهان‌نگاری از ۵ جز اساسی تشکیل می‌شود:
- الگوریتم تعبیه دادن پیام مخفی در پوشش

- پیام مخفی (محرمانه)
- شی پوشش (متن، تصویر، صوت یا ویدئو)
- کلید رمز مشترک (بین فرستنده و گیرنده)
- الگوریتم استخراج داده

کشف شدن هرکدام از اجزای فوق توسط مهاجم، باعث از بین رفتن امنیت سیستم می‌شود، به گونه‌ای که دیگر قابل اعتماد نخواهد بود.
شکل زیر، اجزا و نحوه‌ی عملکرد نهان‌نگاری را به صورت کلی نشان می‌دهد:



همانطور که در شکل بالا مشاهده می‌شود، پیام مخفی (Secret Message) به همراه فایل پوششی (Cover File) وارد رمزگذار (Steganographic Encoder) می‌شود و تابع رمزگذار F(X, M, K)، پیام مخفی را در فایل پوششی قرار می‌دهد و درنتیجه، خروجی رمزگذار، حاوی پیام مخفی است ولی از لحاظ ظاهری، کاملا شبیه به فایل پوششی می‌باشد. برای بازیابی و دیدن پیام مخفی، فایل حامل وارد رمزگشا steganography می‌شود.
قابل توجه است که در نهان‌کاری داده‌ها، ارتباط دو طرف نباید افشا شود. زیرا اگر این ارتباط لو برود به راحتی می‌توان با انجام محاسبات آماری بر روی رسانه یا پوشش مورد استفاده، به اطلاعات مخفی پی برد یا حداقل جلوی انتقال پیام‌ها را در یک کانال ارتباطی گرفت.

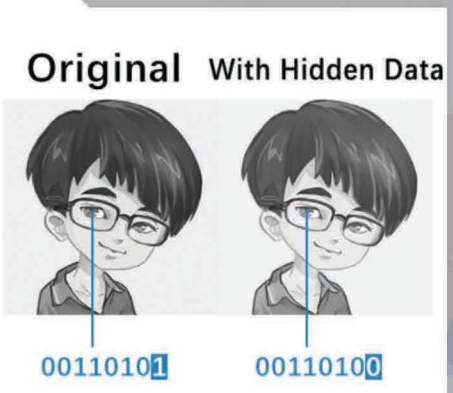
تکنیک‌های نهان‌نگاری

بسته به ماهیت شی پوششی (شی واقعی که داده‌های مخفی در آن تعبیه شده است)، نهان‌نگاری می‌تواند به پنج نوع تقسیم شود:
۱.نهان‌نگاری متن
۲.نهان‌نگاری عکس
۳.نهان‌نگاری ویدیو
۴.نهان‌نگاری صدا
۵.نهان‌نگاری شبکه

در ادامه، نهان‌نگاری در تصویر را به اختصار توضیح می‌دهیم.

نهان‌نگاری در تصویر

به استفاده از تصویر به عنوان شی پوششی برای پنهان کردن پیام، نهان‌نگاری در تصویر گفته می‌شود. در نهان‌نگاری دیجیتال، از تصاویر به طور گسترده به عنوان منبع پوشش استفاده می‌شود زیرا تعداد بسیار زیادی بیت در نمایش دیجیتالی یک تصویر وجود دارد.
به دو عکس زیر با دقت نگاه کنید:



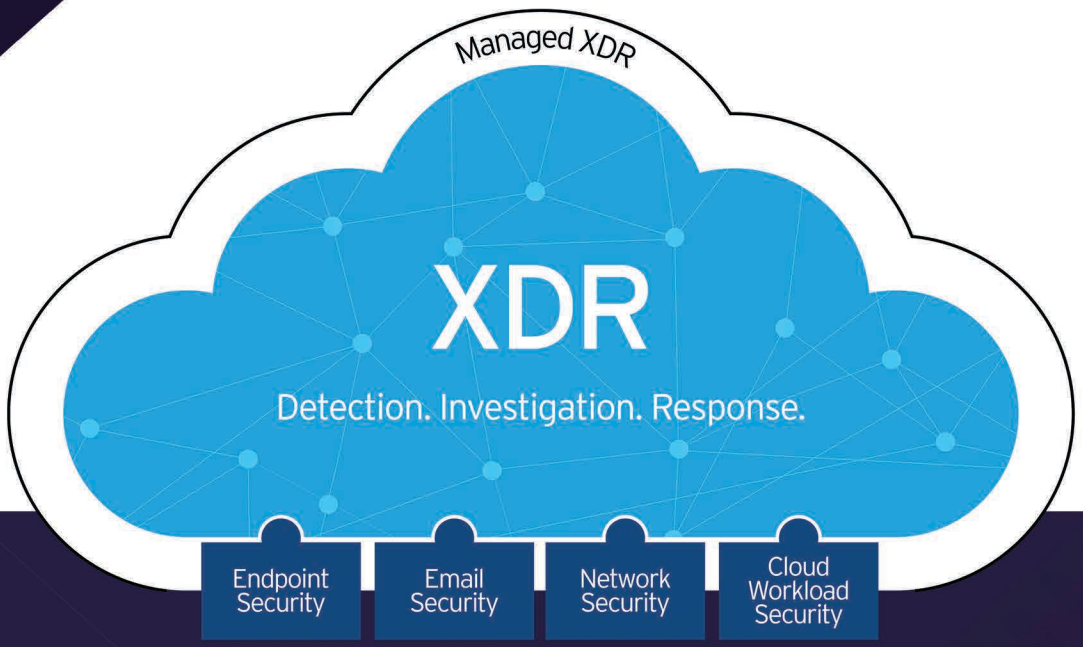
همانطور که مشاهده می‌شود، این دو عکس از لحاظ ظاهری کاملا به یکدیگر شبیه هستند اما عکس سمت راست، حاوی پیامی پنهان می‌باشد.
می‌دانیم هر تصویر از تعداد بسیار زیادی پیکسل تشکیل شده است. در تصاویر سیاه و سفید یا Grayscale، هر پیکسل، یک رشته‌ی ۸ بیتی است که نشان دهنده‌ی رنگ آن پیکسل می‌باشد. به طور مثال رشته‌ی 00000000 نشان دهنده‌ی رنگ سفید و رشته‌ی 11111111 نشان دهنده‌ی رنگ سیاه است. فرض کنید رشته‌ی بیتی یک پیکسل سفید رنگ را از 00000000 به 00000001 تغییر دهیم، یعنی کم‌ارزش‌ترین بیت آن را عوض کنیم. این تغییر بسیار کوچک بوده و برای چشم انسان قابل تشخیص نیست. از این روش، برای قرار دادن پیام پنهان در یک تصویر استفاده می‌شود.
درواقع برای نهان سازی یک پیام مثل "Hello World!" درون یک تصویر، ابتدا باید این پیام را به رشته‌ی بیتی تبدیل کنیم که حاصل آن رشته‌ی بیتی زیر می‌شود:

01101100 01101100 01100101 01001000
01101111 01010111 00100000 01101111
00100001 01100100 01101100 01110010

حال باید با شروع از سمت چپ این رشته، یک به یک بیت‌ها را درون پیکسل‌های تصویر قرار دهیم. به این صورت که بیت مورد نظر را به جای بی‌ارزش‌ترین بیت هر پیکسل قرار می‌دهیم. مثلا اگر اولین بیت 0 باشد و رشته‌ی بیتی اولین پیکسل نیز 01011101 باشد، رشته‌ی بیتی اولین پیکسل به 01011100 تبدیل می‌شود. این روند را تا آن‌جایی ادامه می‌دهیم که تمام بیت‌های پیام مورد نظر، داخل پیکسل‌های تصویر پوششی قرار بگیرند. اکنون پیام مخفی، داخل فایل پوششی قرار گرفته است. در طرف گیرنده نیز باید به اندازه‌ی رشته‌ی بیتی پیغام، شروع به خواندن پیکسل‌ها کرده و بی‌ارزش‌ترین بیت هر پیکسل را خارج نماید. و در نهایت رشته بیتی حاصل را به String تبدیل کند.

روش های دیگری برای نهان‌نگاری در تصویر وجود دارد که در صورت تمایل، پیشنهاد می‌شود آن‌ها را نیز مطالعه کنید. موفق باشید.

منابع:
- https://www.edureka.co/blog/steganography-tutorial
- https://security.tosinso.com/
- https://chenyumin.com/p/how-to-hide-a-secret-message-in-an-image-file-steganography-in-python
- کاربردهایی از پویشده‌نویسی، حدیث ملکی و زینب فرهودی
- مروری بر روشهای پنهان‌نگاری در متن، هدیه ساجدی و شبنم رهبر یعقوبی



سیستم امنیتی هوشمند

XDR

غزال محسنی
ghazalmohseni7997@gmail.com

EDR چیست؟

EDR مخفف Endpoint Detection and Response یک سرویس امنیتی جامع و ادغام شده برای نقاط پایانی[1] است که به طور مداوم و خودکار در حال نظارت و جمع‌آوری داده‌ها از نقاط پایانی و شناسایی فعالیت‌های مشکوک است. به دلیل خودکار بودن، این سرویس تیم امنیتی سازمان‌ها را قادر به شناسایی و برطرف کردن سریع تهدیدها می‌کند. نحوه‌ی عملکرد EDR بدین صورت است که داده‌های مربوط به فعالیت‌ها را از نقاط پایانی جمع‌آوری می‌کند، سپس آن‌ها را تحلیل می‌کند تا بتواند الگوی حمله و تهدید پیدا کند؛ اگر حمله یا تهدیدی پیدا شد به طور خودکار پاسخ می‌دهد و یا به تیم امنیتی اخطار می‌دهد.

XDR چیست؟

XDR[2] (تشخیص و پاسخ گسترده) یک تکنولوژی امنیت سایبری جدید بر پایه پلتفرم ابر و زیرساخت کلان‌داده است که در سال ۲۰۱۸ و توسط Nir Zuk اختراع شد و برای تیم‌های امنیتی سازمان‌ها ویژگی‌هایی مانند انعطاف‌پذیری، مقیاس‌پذیری و اتوماسیون را فراهم می‌کند. همچنین قادر به تشخیص تهدیدها و حفاظت در برابر حملات سایبری و دسترسی‌های غیرمجاز است. XDR را نسخه پیشرفته و پیچیده‌تر EDR نیز می‌نامند.

XDR چگونه کار می‌کند؟

برخلاف EDR که داده‌هایش را فقط از Endpointها به دست می‌آورد، XDR تمامی اطلاعات در Endpointها، ایمیل‌ها، شبکه، سرورها و زیرساخت‌های ابری را جمع‌آوری و بررسی می‌کند، برای همین می‌تواند خیلی سریع‌تر حمله‌ها را شناسایی کند و قبل از آن که باعث به‌وجود آمدن مشکلی شود آن‌ها را خنثی کند.

XDR تشکیل شده از ۳ بخش است:

1. Telemetry and data analysis:

همانطور که گفتیم در این قسمت داده‌ها از نقاط پایانی، ایمیل، شبکه، سرور و فضای ابری استخراج می‌شود. با استفاده از تجزیه و تحلیل داده‌ها و هشدارهایی که از سایر لایه‌ها دریافت کرده است سعی می‌کند جدی‌ترین تهدید را پیدا کند. یک تهدید و حمله خوش‌ساخت فقط در یک لایه عمل نمی‌کند، به عنوان مثال هم در ایمیل و هم در نقطه پایانی به صورت همزمان شروع به حمله می‌کند و بدین‌گونه تشخیص آن برای سیستم‌های پیشین و EDR مشکل است ولی XDR تمامی لایه‌ها را همزمان بررسی می‌کند، اگر تهدید یا حمله‌ای پیدا کرد آن را ایزوله می‌کند تا از گسترش آن جلوگیری کند. هر لایه، سطح حمله[3] متفاوتی دارد.

a. نقاط پایانی:

یک سیستم XDR می‌تواند به شما بگوید تهدید از کدام نقطه پایانی آمده است

و قرار است چگونه در سایر نقاط گسترش پیدا کند. پیدا کردن نفوذ و تهدید با استفاده از شاخص‌های سازش[4] و مشخص کردن هدف نفوذ و مدل حمله توسط شاخص‌های حمله[5] انجام می‌شود.

b. ایمیل:

ایمیل یکی از بزرگ‌ترین سطوح حمله است که معمولا بیشتر از بقیه‌ی لایه‌ها مورد استفاده قرار می‌گیرد، برای همین بررسی ایمیل‌ها باید الویت داشته باشد و XDR خطرات از جانب این لایه را محدود کند؛ به طوری که تهدیدها، حساب کاربری‌های آسیب‌دیده و همچنین الگوی حمله‌ها را مشخص کند. نحوه‌ی برخورد با مشکلات هم بدین صورت است که ایمیل‌ها را قرنطینه می‌کند، حساب کاربری‌ها را مجددا تنظیم می‌کند و فرستندگان ایمیل‌های مخرب را مسدود می‌کند. تهدیدات ایمیل تا زمانی که بر روی لینک موجود در ایمیل کلیک نشود نمی‌توانند هیچ اثری روی نقاط پایانی داشته باشند برای همین قوانین امنیتی ایمیل متوجه نمی‌شوند که محتوای ایمیل یک تهدید است یا متن ساده و به این دلیل هرگونه تهدیدی به راحتی می‌تواند وارد صندوق پیام‌ها شود. اگر نحوه‌ی تشخیص تهدیدها در نقاط پایانی را به ایمیل وصل کنیم می‌توانیم به صورت خودکار در صندوق پیام‌ها به دنبال تهدید و الگوها بگردیم.

c. شبکه:

تهدیدها در شبکه یا گسترش پیدا می‌کنند و یا با سرورهای فرمان و کنترل ارتباط برقرار می‌کنند. پس با تجزیه و تحلیل شبکه می‌توان آسیب‌پذیری‌های اساسی را مشخص کرد، همچنین می‌توان نقاط و دستگاه‌هایی که احتمال بروز حمله از طرف آن‌ها وجود دارد را پیدا کرد. در شبکه ابتدا رفتار مشکل‌ساز پیدا می‌شود، سپس براساس موقعیت آن در شبکه، می‌توان نحوه‌ی ارتباط برقرار کردن مشکل با سایر مشکلات و شیوه‌ی جابه‌جایی آن در شبکه را پیدا کرد.

d. سرور و زیرساخت‌های ابری:

حفاظت از سرورها و زیرساخت‌های ابری اهمیت بسیاری دارد و از طرفی بسیار شبیه به نقاط پایانی است چون در هر دو باید تهدید را بررسی کنیم تا متوجه شویم چگونه وارد شبکه شده است و چگونه گسترش پیدا می‌کند. در سرورها، XDR نحوه‌ی تاثیر تهدید بر حجم زیادی از داده‌ها و نحوه‌ی انتشار تهدید را بررسی می‌کند، درصورت نیاز سرور را جدا می‌کند تا تهدید منزوی شود و سرعت انتشارش کم شود. در محیط ابری نیز فرایندهایی که ممکن است تهدید بتواند از طریق آن‌ها گسترش پیدا کند را شناسایی و متوقف می‌کند.

2. Detection:

XDR قادر است تهدیدها و منشا آن‌ها را پیدا کند، آن‌هایی که نیازمند پاسخ هستند را یا خودش پاسخ دهد یا به تیم امنیتی هشدار دهد و همچنین می‌تواند تهدیدهایی که از نرم‌افزارهای مجاز استفاده می‌کنند را شناسایی کند.

3. Response:

همانند XDR، EDR نیز می‌تواند تهدیدهایی که پیدا کرده است را حذف یا مهار کند ولی تفاوتی که وجود دارد این است که EDR حتما باید تهدید در یک نقطه پایانی باشد در حالی که XDR در هرجایی از لایه‌های ذکر شده می‌تواند تهدید را پیدا و مهار کند.

مزایای استفاده از XDR:

۱. با استفاده از تجزیه و تحلیل بر مبنای هوش مصنوعی و عملیات‌های حفاظت در برابر تهدید می‌تواند حملات شناخته شده و ناشناخته را با حفاظت قوی از نقطه پایانی مسدود کند.

۲. قابلیت مشاهده و بررسی داده‌ها در سرتاسر لایه‌ها را فراهم می‌کند.

۳. با استفاده از تجزیه و تحلیل‌های مبتنی بر هوش مصنوعی، به صورت مداوم و خودکار در حال بررسی و پیدا کردن تهدیدات پیشرفته و پنهان است. همچنین تمام هشدارها بررسی می‌شوند و موارد مهم‌تر انتخاب و بیشتر بررسی می‌شوند.

۴. تهدیدات را بدون آسیب زدن به عملیات‌های در حال انجام از بین می‌برد.

۵. قابلیت بازیابی سریع هاست بعد از یک حمله با استفاده از حذف فایل‌های مخرب و بازیابی مجدد آن‌ها را دارد.

۶. قابلیت‌های اتوماسیون برای کارهای تکراری ارائه می‌دهد.

چرا شرکت‌ها و سازمان‌ها به XDR نیاز دارند؟

مرکز عملیات‌های امنیتی در سازمان‌ها به پلتفرمی نیاز دارند که بتواند به طور هوشمندانه تمام داده‌های امنیتی مربوطه را جمع‌آوری و تهدیدات را مشخص کند. از آنجایی که تهدیدات از شیوه‌های پیچیده‌تری برای دور زدن سیستم‌های امنیتی سنتی استفاده می کنند، سازمان‌ها در تلاش هستند تا راه حلی برای حفاظت از دارایی‌های دیجیتال آسیب‌پذیر درون شبکه بدون استفاده از سیستم امنیتی سنتی، پیدا کنند. در سال‌های اخیر فشار و تنش بر تیم‌های امنیتی بیشتر شده است و کار کردن در خانه نیز باعث فشار مضاعف بر منابع شده است و متخصصان باید بتوانند کارهای بیشتری را با استفاده از منابع کمتری انجام دهند. بنابراین شرکت‌ها هم باید بتوانند امنیت دارایی‌ها را حفظ کنند و هم از فشار موجود بر منابع و کارمندان خود بکاهند. از طرفی ابزارهای امنیتی موجود با این حجم از داده نمی‌توانند عملکرد مناسبی داشته باشند و همچنین توانایی الویت‌بندی کردن هشدارها را ندارند، بنابراین دوباره حجم کاری تیم امنیتی و فشار بر منابع زیاد می‌شود.

XDR در مرحله اول با استفاده از تجزیه و قابلیت اتوماسیون می‌تواند بسیاری از مشکلات را به تنهایی برطرف کند و تنها در صورت نیاز به تیم امنیتی هشدار تهدید را اعلام کند. همچنین به دلیل بررسی همزمان لایه‌ها و منزوی کردن تهدیدهای موجود، سایر حمله‌ها قادر به دور زدن آن نیستند.

[1] Endpoints، دستگاه‌های محاسباتی اند که با شبکه در ارتباط هستند مانند سرورها، موبایل، کامپیوتر

[2] Extended detection and response

[3] Attack surface، تعداد نقاط احتمالی که حتی کاربر تایید هویت نشده نیز می‌تواند از آن وارد سیستم شود و اطلاعات را استخراج کند

[4] indicators of compromise (IOCs)

[5] indicators of attack (IOAs)

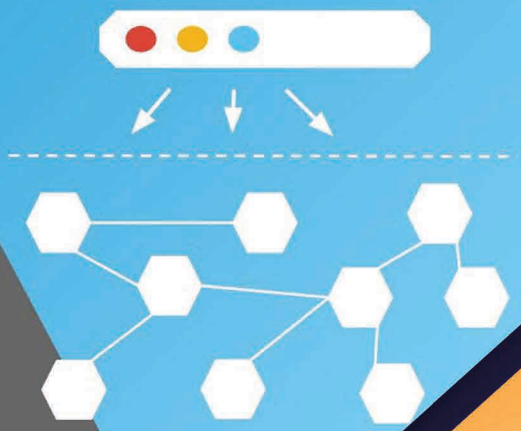
منابع:

• www.trendmicro.com, What Are the XDR Security Layers?

• www.vmware.com, Extended Detection and Response (XDR)

• www.fortinet.com, What is XDR?

• www.mcafee.com, What Is Extended Detection and Response (XDR)?



Service Mesh

مقدمه

از سال ۲۰۱۱ با پیشرفت تکنولوژی مجازی‌سازی نرم‌افزار و سخت‌افزار بسیاری از نوآوری‌ها و پیشرفت‌ها در عرصه‌ی توسعه‌ی نرم‌افزار رقم خورد که با اتخاذ رویکرد microservice توسط شرکت‌هایی چون Netflix، Amazon و ... در این سال آغاز گشت. این رویکرد با شکستن و تجزیه نرم‌افزارهای monolithic به قسمت‌های کوچک‌تر و مستقل که به نام microservice شناخته می‌شوند شروع شد. اما قبل از آن‌که معماری Service Mesh و جایگاه آن شرح داده شود، توضیحاتی درباره معماری microservice و تفاوت آن با monolithic داده می‌شود.

سرویس مش در معماری میکروسرویس

Service Mesh in Microservice Architecture

فرید احمدپور مبارکه

f.ahmadpour2013@gmail.com

Monolithic vs Microservices

در اپلیکیشن‌های monolithic ایده‌ی طراحی به این صورت بود که هسته و کد اصلی نرم‌افزار، همه در یک بسته و به عنوان یک سرویس عمل می‌کرد. اپلیکیشن‌های monolithic از اپلیکیشن‌های کوچک ایجاد، و سپس به صورت معماری چندلایه‌ای که در آن frontend از backend و backend از منابع داده جدا شده‌اند، ساخته می‌شوند. در این معماری frontend تعاملات با کاربر را مدیریت می‌کند، لایه‌ی میانی وظیفه‌ی انجام اعمال منطقی نرم‌افزار را به عهده دارد و لایه‌ی backend دسترسی به داده‌ها را مدیریت می‌کند. به عنوان مثال معماری سه لایه‌ای MVC شامل View، Model و Controller. اما توسعه‌ی اپلیکیشن‌ها به روش monolithic همراه با مشکلات و عیب‌هایی بود. از جمله این مشکلات می‌توان به موارد زیر اشاره کرد:

- فرایند توسعه‌ی بسیار کند
- درک و اصلاح اپلیکیشن سخت و دشوار
- انتشارهای نامرتب و پر از مشکلات
- استقرار مداوم بسیار سخت
- کمبود قابلیت اعتماد بخاطر سختی در آزمایش و تست اپلیکیشن

فرایند توسعه‌ی اپلیکیشن‌ها و پروژه‌ها در اندازه‌ی بزرگ بین تعداد زیادی از تیم‌های کوچکی تجزیه و هماهنگی می‌شود که این تیم‌ها قادر به فعالیت به صورت مستقل هستند. همانطور که اپلیکیشن با گذر زمان بزرگ و بزرگ‌تر می‌شود، مشکلات و پیچیدگی‌هایی که در بالا ذکر شد نیز بیشتر می‌شوند. این امر باعث شد که توسعه‌دهندگان به فکر یک رویکرد و معماری جدید برای توسعه‌ی اپلیکیشن‌ها بیوفتند. این معماری جدید با تعداد کمی از توسعه‌دهندگان در اوایل سال ۲۰۱۴ آغاز شد که امروزه آن را به عنوان معماری microservices می‌شناسند. البته کلمه microservices اولین بار در یک کارگاه مربوط به معماری نرم‌افزار در سال ۲۰۱۱ استفاده شد که به معماری بسیار ابتدایی از microservices امروزی اشاره داشت.

اپلیکیشن‌های بر پایه‌ی microservices مجموعه‌ای از سرویس‌های خود مختارند که هر کدام وظیفه‌ی مشخصی دارند و فقط همان را انجام می‌دهند و در کنار هم عملیات‌های مورد نیاز در یک کسب‌وکار را انجام می‌دهند. در معماری microservices نکات اساسی وجود دارند که به ما در درک بهتر این نوع از معماری کمک می‌کنند. این نکات در زیر لیست شده‌اند:

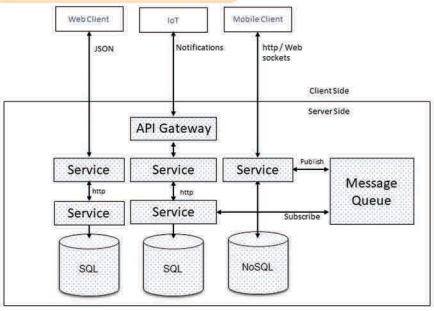
- هر سرویس خودمختار است به طوری که به صورت مستقل توسعه و مستقر می‌شود.
- هر سرویس در صورت نیاز می‌تواند به طور مستقل گسترش داده شود بدون آنکه دیگر سرویس‌ها تغییری کنند.
- هر سرویس بر پایه‌ی یکی از توانایی‌های کسب‌وکار، ارائه و طراحی شده به طوری که هر سرویس یکی از هدف‌های کسب‌وکار را دنبال می‌کند.
- از آن جایی که سرویس‌ها یک runtime مشترک ندارند، می‌توان هر سرویس را با زبان‌های برنامه‌نویسی متفاوت توسعه داد.

دو عامل و ایده‌ی اصلی موثر در تحول و انقلاب این معماری نرم‌افزار، اول قابلیت سازگار شدن با تیم‌های کوچک و دوم ارائه‌ی یک مدل self-service برای استفاده از باقی خدمات است، به طوری که امروزه این نوع معماری توسط بسیاری از کمپانی‌ها و تیم‌ها در سراسر جهان مورد قبول و در حال استفاده است.

سه قابلیت اصلی که این رویکرد در اختیار توسعه دهندگان قرار می‌دهد شامل: گسترش پذیری بالا، جدایی بالا و چابکی فراوان است. این سه مورد موثرترین عوامل در طراحی و پیاده سازی اپلیکیشن‌های توزیع شده‌ی قابل گسترش است.

تفاوت اصلی بین monolithic و microservices این است که در اپلیکیشن‌هایی با معماری microservice، سرویس‌ها آزادانه و به صورت جدا هستند (loosely coupled) و این سرویس‌ها از dumb pipe یا پروتوکل‌هایی مثل REST یا gRPC برای ارتباط با یکدیگر استفاده می‌کنند.

در شکل زیر معماری یک microservice را مشاهده می‌کنید که در آن clientهای مختلف از سرویس‌های یکسانی استفاده می‌کنند. هر سرویس از زبان برنامه‌نویسی یکسان یا متفاوت با بقیه پیاده‌سازی شده است و هر کدام می‌توانند جداگانه گسترش و یا مستقر شوند.



معماری microservice فواید و مزایای بسیاری دارد که در زیر به تعدادی از آن‌ها اشاره شده:

- توسعه‌ی سریع
- گسترش‌پذیری
- استفاده از چندین زبان برنامه نویسی
- مناسب برای تیم‌های cross-functional
- عدم نیاز به تعهد طولانی مدت به یک استک خاص
- کاهش عیب و خطا در نرم‌افزار

البته توسعه اپلیکیشن‌ها با این معماری همراه با چالش‌ها و معایب نیز است. از جمله این معایب:

- پیچیدگی سیستم‌های توزیع شده.
- افزایش مصرف منابع
- دشواری مدیریت نیازمندی‌های هر سرویس.
- دشواری در پیدا کردن ریشه‌ی مشکل طی از کار افتادن یکی از سرویس‌ها
- افزایش فرایندهای عملیاتی (Ops)

Service Mesh Architecture

معماری Service Mesh یک لایه‌ی زیرساخت اپلیکیشن است که روی اپلیکیشن‌های cloud-native قرار می‌گیرد. این معماری از سال ۲۰۱۷ بین توسعه‌دهندگان محبوب شد و به طبع هنوز مفهوم بسیار جوانی است. Service Mesh درواقع یک لایه‌ی انتزاعی بالای اپلیکیشن‌های شما ایجاد می‌کند. برای مثال این امر برای جدا کردن امنیت اپلیکیشن استفاده می‌شود. Service Mesh می‌تواند ارتباطات بین سرویس‌ها را با TLS امن سازی کند. فایده‌ی این امر این است که دیگر توسعه دهنده نیاز ندارد تا خود رمزگذاری و رمزگشایی TLS را پیاده‌سازی کند.

در سال ۲۰۱۷ آقای William Morgan تعریف جامعی از Service Mesh ارائه دادند که در اینجا قسمتی از سخن ایشان آورده شده:

"*A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud-native application.*"

طبق تعریف ایشان Service Mesh یک لایه‌ی زیرساخت منحصر شده برای مدیریت ارتباطات سرویس تا سرویس است. این لایه وظیفه‌ی رساندن درخواست‌ها در طول توپولوژی پیچیده‌ی سرویس‌ها که شامل اپلیکیشن‌های cloud-native امروزه می‌شود را دارد.

می‌توان Service Mesh را به عنوان یک عامل جدا کننده بین Dev و Ops دانست چرا که تیم توسعه، دیگر نیازی به پیاده‌سازی کدهایی که در فاز عملیات استفاده می‌شوند و تیم عملیات نیازی به کامپایل کردن کامل سیستم ندارد. در این صورت این دو فاز می‌توانند به صورت مستقل عمل کنند. این امر تحولی بزرگ در حوزه DevOps نسبت به نسخه‌ها و روش‌های قدیمی‌تر آن ایجاد کرد.

همانطور که پیش‌تر اشاره شد، سرویس‌ها در این معماری باید با یکدیگر ارتباط و تعامل داشته باشند. این امر به کمک Smart Endpoints و Dumb Pipes انجام می‌شود که در ادامه هر کدام توضیح داده شده‌اند.

Smart Endpoints: ارتباط سرویس تا سرویس به کمک DNS رکوردها انجام می‌شود که کل microservice را resolve میکند.

Dumb Pipes: ارتباط سرویس تا سرویس با کمک پروتوکل‌های پایه‌ی شبکه انجام می‌شوند، مانند gRPC، REST، HTTP و... .

قابلیت‌های Service Mesh را می‌توان به دو دسته کلی تقسیم کرد. دسته‌ی اول قابلیت‌های پایه‌ای است مثل service discovery و request balancing.

دسته‌ی دوم علاوه بر قابلیت‌های دسته اول قابلیت‌های دیگری مثل fallback، timeouts و circuit breaking را در اختیار توسعه‌دهندگان قرار می‌دهد. معروف‌ترین ابزارها و پلتفرم‌هایی که قابلیت‌های دسته‌ی دوم را ارائه می‌دهند ابزارهایی چون Linkerd، Istio و Consul هستند و معروف‌ترین ابزار دسته اول Kubernetes است که پلتفرمی کامل برای orchestration می‌باشد.

یک معماری Service Mesh درست و خوب باید قاعده‌هایی بدون نیاز به پیاده‌سازی و نوشتن کد جدید برقرار کند. این قواعد در ادامه شرح داده شده‌اند؛ اما قبل از آن دو مفهوم مربوط به شبکه توضیح داده می‌شود که در درک قواعد گفته شده بسیار کمک کننده است.

سطح کنترل یا Control Plane: سطح کنترل قسمتی از شبکه است که وظیفه‌ی مدیریت، اعمال قوانین، پیکربندی، ثبت و پیدا کردن گره‌ها و ... را دارد. این اعمال روی سطح داده یا Data Plane انجام می‌شوند.

سطح داده یا Data Plane: قسمتی از شبکه که داده‌ی اصلی کاربران را جابه‌جا می‌کند.

در ادامه قواعد اشاره شده نام برده شده و شرح مختصری داده شده‌اند.

Observability: سطح کنترل باید قابلیت مشاهده و مراقبت سرویس‌های درحال اجرا در سطح داده را فراهم کند.

Routing: قوانین مسیریابی باید در سطح کنترل قابل پیکربندی باشد و به تمامی گره‌های سطح انتقال داده شده باشد.

Automatic scaling: سطح کنترل باید به صورت خودکار سرویس‌ها را گسترش دهد تا هنگام بار سنگین مشکلی پیش نیاید.

Separation of duties: رابط کاربری سطح کنترل باید این امکان را به توسعه‌دهنگان بدهد که بتوانند عملیات‌های مدیریت Service Mesh را به طور مستقل انجام دهند.

Trust: امن سازی ارتباطات بین سرویس‌ها و مدیریت گواهینامه‌ها

Automatic service registration and discovery: پیدا کردن سرویس‌ها به صورت خودکار طی استقرار اپلیکیشن‌ها

Resilient: استفاده از قواعد ارتجاعی برای همه گره‌ها. این عمل به عنوان پروکسی جانبی برای مدیریت ترافیک استفاده می‌شود.

منابع:

- Mastering Service Mesh Enhance, secure, and observe cloud-native applications withIstio, Linkerd, and Consul By Anjali Khatri & Vikram Khatri
- Microservice in Action By Morgan Bruce & Paulo A. Pereira

باج‌افزار

Ransomware

فاطمه نشاط‌دوست
Fatemeh.neshatdoust@yahoo.com

باج‌افزار (ransomware) چیست؟

باج‌افزارها یکی از انواع بد افزارها[1] هستند که با استفاده از نوعی کدگذاری داده‌ها، کاربر را به انتشار یا مسدود کردن دسترسی به داده‌ها و سیستم تهدید می‌کنند و در قبال آن خواستار پرداخت مبلغی معین در زمانی مشخص شده هستند و اگر کاربر مبلغ درخواستی را واریز نکند، داده‌ها برای همیشه از بین می‌روند.

این روزها حملات باج‌افزاری، بسیار رایج شده است و شرکت‌های بزرگ در آمریکای شمالی و اروپا قربانی این حملات شده‌اند.

سازمان‌های اطلاعاتی از جمله FBI توصیه می‌کنند قربانیان برای جلوگیری از چرخه‌ی این بدافزارها از پرداخت باج خودداری کنند و همچنین احتمال می‌رود برای آن‌ها حملات مجدد نیز اتفاق بیفتد؛ به‌ویژه اگر باج‌افزار از سیستم پاک نشده باشد.

تاریخچه

قدیمی‌ترین نوع باج‌افزارها در سال ۱۹۸۹ توسعه یافت که در آن زمان، پرداخت باج از طریق نامه‌ی حلزون[2] ارسال و پس از آن، کلید رمزگشایی برای کاربر فرستاده می‌شد.

در سال ۱۹۹۶ باج‌افزاری با عنوان اخاذی رمزنگاری[3] توسط یونگ[4] و یانگ[5] در دانشگاه کلمبیا معرفی شد که این ایده، ابزارهای رمزنگاری مدرن را نشان می‌دهد. یونگ و یانگ اولین حمله‌ی رمزنگاری را در کنفرانس امنیت و حریم خصوصی IEEE ارائه کردند. ویروس آن‌ها حاوی یک شیوه‌ی رمزگذاری عمومی بود که فایل‌های قربانیان را رمزگذاری می‌کرد و قربانی را ترغیب می‌کرد تا هزینه را پرداخت کند و پس از آن، عملیات رمزگشایی و بازگشت فایل‌ها انجام می‌شد.

مجرمان سایبری در طول این سال‌ها، باج‌ها را با شیوه‌هایی غیر معمول درخواست می‌کنند و این رویداد به آن‌ها کمک می‌کند تا همچنان ناشناس باقی بمانند. به عنوان مثال باج‌افزار معروف fusob از قربانیان می‌خواهد که به جای ارزهای معمولی، از کارت‌های هدیه آیتیونز[6] استفاده کنند.

حملات باج افزاری با رشد رمزارزها مانند بیت‌کوین، عمومیت خود را افزایش داده و همچنین فراتر از بیت‌کوین از رمزارزهای دیگری مثل اتریوم[7] ، لایت کوین[8] و ریپل[9] نیز استفاده می‌کنند.

باج افزارها چگونه کار می کنند؟

دو نوع رایج باج‌افزارها رمزگذار[10] و قفل صفحه[11] هستند.

رمزگذارها: همانطور که از نامشان پیداست، داده‌ها را بر روی یک سیستم، رمزگذاری می‌کنند و درنتیجه دسترسی به داده‌ها بدون داشتن رمز، امکانپذیر نخواهد بود.

قفل صفحه: این نوع باج‌افزارها به سادگی، دسترسی به سیستم را با یک صفحه نمایش قفل، مسدود می‌کنند و در این هنگام، قربانیان روی صفحه

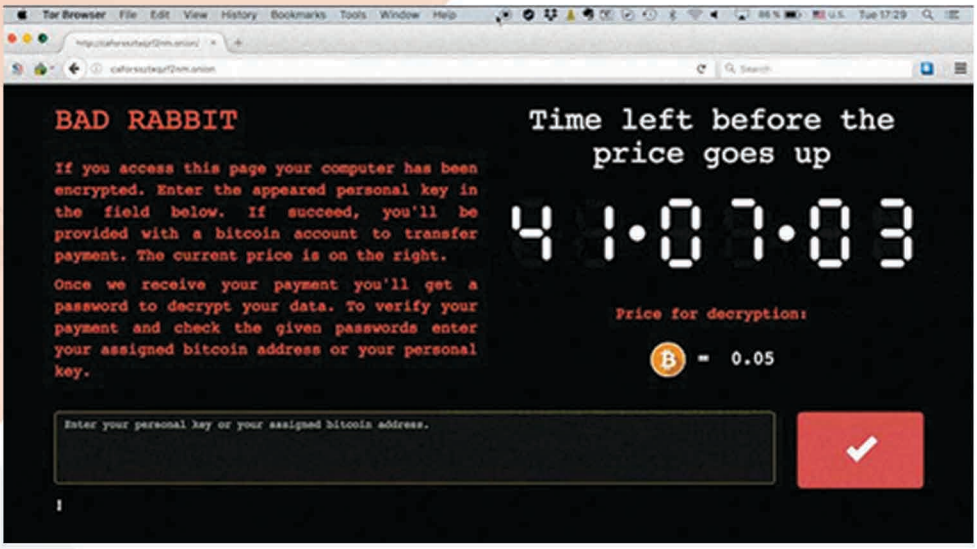
قفل با یک آگهی مواجه می‌شوند که در آن مبلغ و روش پرداخت هزینه و همچنین مهلت داده شده، نوشته شده است. قربانیان پس از پرداخت هزینه، کد مورد نظر برای رمزگشایی را دریافت می‌کنند. همچنین باید توجه داشت که رمزگشایی تضمینی نیست و گزارش‌های متفاوتی از دسترسی به داده‌ها، پس از پرداخت باج ارائه شده است. در برخی مواقع پس از پرداخت باج، دسترسی اطلاعات به کاربر داده می‌شود اما در همان زمان، مجرمان بدافزار دیگری روی سیستم نصب می‌کنند!

رمزگذاری باج‌افزارها به طور فزاینده‌ای، کاربران تجاری و صنایع را مورد هدف قرار داده است. زیرا مشاغل بزرگ، بیشتر از عامه‌ی مردم برای باز کردن قفل‌های سیستم و از سرگیری عملیات روزانه‌شان، هزینه می‌کنند.
حمله ویروس‌ها در باج‌افزارهای سازمانی معمولا با یک ایمیل آغاز می‌شود؛ کاربر یک پیوست را باز می‌کند یا بر روی یک آدرس اینترنتی که مخرب است کلیک می‌کند، سپس یک عامل باج‌افزار نصب می‌شود و شروع به رمزگذاری فایل‌های کاربر می‌کند.

به طور کلی باج‌افزار برای رسیدن به هدف خود، پنج مرحله را طراحی می‌کند که عبارت‌اند از:
۱. سیستم به خطر افتاده است.[12]
۲. باج‌افزار شروع به کنترل رایانه می‌کند.
۳. به قربانی اطلاع داده می‌شود.
۴. باج پرداخت می‌شود.
۵. دسترسی کاربر بازگردانده می‌شود.

بنابراین اگر دستگاهی به اینترنت متصل شود، باید با جدیدترین افزونه‌های امنیتی-نرم‌افزاری به روز شود و دارای ضد بدافزارهایی باشد که باج‌افزار را شناسایی و متوقف می‌کند. سیستم عامل‌های قدیمی مانند ویندوز XP در معرض خطر بیشتری قرار دارند.
به عنوان مثال، در ماه مه ۲۰۲۱، یک گروه هکری به نام DarkSide حمله باج‌افزاری را علیه یکی از بزرگترین خطوط لوله‌ی سوخت[13] در ایالات متحده انجام داد و پس از آن برای مدتی، عملیات توزیع سوخت در چندین ایالت متوقف شد. حدود ۲۰ سال پیش، در دوران پیش از سایبری، چه حمله فیزیکی عظیمی لازم بود تا موجی از کمبود گاز در سراسر شرق کشور ایجاد شود؟!
بر اساس یک برآورد، باج‌افزارها در سال ۲۰۲۱ تقریبا ۲۰ میلیارد دلار برای اقتصاد جهانی هزینه خواهند داشت که نسبت به سال ۲۰۱۵، ۵۷ برابر افزایش یافته است.

چگونه می‌توانیم از باج‌افزارها دوری کنیم؟
۱. اطمینان حاصل کنید که فایروال یا ضد ویروس شما روشن است.
۲. از ورود به وب‌سایت‌های مشکوک خودداری کنید.
۳. هنگام باز کردن پیام‌ها و ایمیل‌های مشکوک هوشیار باشید.
۴. انتخاب نرم‌افزار ضد ویروس از یک شرکت معتبر می‌تواند به محافظت از کامپیوتر در برابر آخرین تهدیدات باج‌افزار کمک زیادی کند.



چه کسانی در خطر باج‌افزارها هستند؟
هر دستگاهی که به اینترنت متصل باشد در خطر باج‌افزارها قرار دارد؛ در واقع باج‌افزارها هر دستگاه محلی و فضای ذخیره‌سازی متصل به شبکه را شناسایی می‌کنند. اگر شبکه محلی، یک شبکه تجاری باشد، باج‌افزار می‌تواند اسناد مهم پرونده‌های سیستم را رمزگذاری کند تا با این اقدام، خدمات و بهره‌وری در شرکت متوقف شود.

[1] malware
[2] snail mail
[3] cryptoviral extortion
[4] Moti Yung
[5] Adam Young
[6] Apple iTunes
[7] Ethereum
[8] Litecoin
[9] Ripple
[10] encryptor
[11] screen locker
[12] The System Is Compromised.
[13] colonial pipeline

منابع:
• www.proofpoint.com/us/threat-reference/ransomware
• edition.cnn.com/2021/09/13/perspectives/ransomware-attacks-cybersecurity/index.html

خبرهای مهم

گزیده اخبار مهر ماه

گردآورنده: سروش ذوالفقاری
zolfaghari.soroush@gmail.com

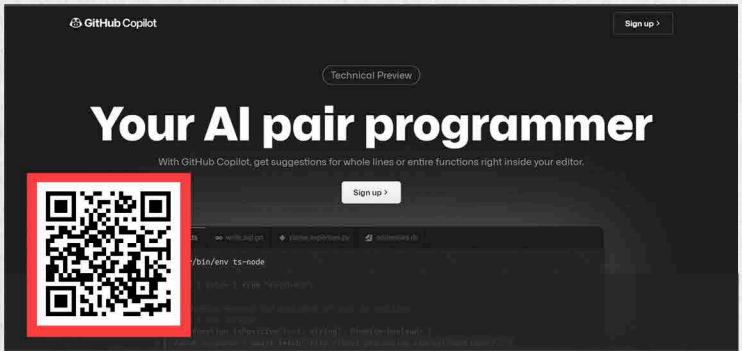
وردپرس 5.8.1 با رفع 3 آسیب پذیری منتشر شد

وردپرس یک سیستم مدیریت محتوای کاملا رایگان و متن‌باز (Open Source) است. این سیستم از امنیت خوبی برخوردار است و تجربه‌ی کاربری بسیار ساده‌ای دارد. به همین خاطر کاربران زیادی از وردپرس برای ساخت وب‌سایت استفاده می‌کنند.
نسخه‌ی جدید با عنوان نسخه‌ی امنیتی منتشر شده و همانطور که گفتیم ۳ آسیب پذیری و ۶۰ اشکال و باگ برطرف شده است.

دست جدید Google برای ارائه‌ی خدمات سایبری به مشتریان

غول فناوری ایالات متحده اعلام کرد در میان نگرانی‌های روزافزون بشر در زمینه امنیت سایبری، این شرکت اقدام به تاسیس یک تیم جدید سایبری کرده است. تیم جدید گوگل بر ایمنی انتقالات ابری، تهدید اطلاعات و خدمات مشاوره امنیتی تمرکز خواهد کرد.

شرکت github و OpenAI یک محصول جدید را ارائه دادند که در واقع یک autocompletion هوشمند است که بر اساس اطلاعاتی که در اختیار آن قرار می‌دهید فانکشن یا الگوریتم مورد نظر شما را برایتان کامل می‌کند. این برنامه برای Visual Studio code و در تاریخ 29 ژوئن 2021 برای اولین بار ارائه شد.



پروژه جدید گیت هاب به نام copilot

مایکروسافت: هکرهای مرتبط با ایران شرکت‌های دفاعی آمریکا و اسرائیل را هدف قرار داده‌اند.

مایکروسافت روز دوشنبه اعلام کرد هکرهای مرتبط با ایران شرکت‌های دفاعی آمریکا و اسرائیل را هدف قرار داده‌اند. مایکروسافت این هکرها را از ماه ژوئیه ردیابی کرده است و اعلام کرد که آن‌ها تقریبا ۲۰ هدف را با موفقیت به خطر انداخته‌اند.
این شرکت معتقد است که این تیم‌ها به احتمال زیاد از منافع ملی ایران حمایت می‌کنند.

رئیس سایبری بریتانیا: روسیه مسئول بیشترین حملات ویرانگر باج‌افزار است.

لندی کامرون، رئیس مرکز ملی امنیت سایبری کشورها (NCSC) هشدار داد که حملات باج‌افزار جز بالاترین خطرها برای مشاغل انگلیس هستند. وی تاکید کرد مجرمان سایبری روسیه و همسایگانش پشت بزرگترین حملات اخاذی از مشاغل انگلستان هستند.

تیم Broadcom's Symantec Threat Hunter اعلام کرد که موفق به کشف یک باج‌افزار (هرچند توسعه نیافته) شده‌اند که هدف حملات آن بیشتر تشکیلات و سرمایه‌گذاری‌های اقتصادی است.

خانواده‌ی جدید باج‌افزارها به نام Yanluowang توسط محققان پیدا شد.

چندی پیش نسخه‌ی جدید کالی لینوکس ۲۰۲۱.۳ به همراه تغییرات و رفع برخی مشکلات منتشر شد. این سیستم‌عامل قدرتمند و معروف، دارای تعداد زیادی ابزار برای تست نفوذ و امکانات بسیاری برای هکرها است.

از مهم‌ترین ویژگی‌های اضافه شده در آپدیت جدید می‌توان به عملکرد بهتر VM در حالت Live و ابزارهای جدید برای شبیه‌سازی و کنترل ساب‌دامین و حملات WIFI اشاره کرد.

حدودا ۷ ابزار جدید به این سیستم‌عامل اضافه شده است که می‌توانید با استفاده از لینک به طور دقیق آن‌ها را بررسی کنید.

هشدار فیشینگ!

کلاهبرداران از نمادهای ریاضی برای لوگوی Verizon استفاده می کنند.

علی رغم تمام پولی که برندهای بزرگ برای طراحی لوگو هزینه می‌کنند، مردم در به یادآوری آن‌ها وحشتناک هستند. و این مسئله کار را برای کلاهبرداران آسان‌تر می‌کند که افراد را فریب دهند تا روی لینک های مخرب کلیک کنند.

شرکت verizon یکی از آن دسته از برندهایی است که اخیرا بسیار مورد حمله‌ی فیشینگ قرار داده شده است و کاربران بسیاری در دام این حملات قرار گرفته‌اند.

نهان‌نگاری و روش‌های آن

سیستم امنیتی هوشمند

سرویس مش در معماری میکروسرویس

باج‌افزار

Cyber news

روز صفرم ترجمه ی عبارت Zero Day می‌باشد که در تعبیر لغوی یعنی روزی که هنوز به آن نرسیده‌ایم و از وجود چنین چیزی هم خبر نداریم، وقتی صحبت از حمله Zero Day می‌شود یعنی در خصوص حمله ای صحبت می کنیم که هیچکس تا کنون آن را شناسایی نکرده است و هیچ دانشی هم در خصوص آن وجود ندارد که چگونه آن را تشخیص و بعضا از بروز آن جلوگیری کنیم.

در این نشریه سعی بر آن است تا زوایای پنهان و ناشناخته در دنیای امنیت اطلاعات مورد بررسی قرار گرفته و به جدیدترین اخبار و تکنولوژی‌های این حوزه پرداخته شود. مخاطبین این نشریه تمامی دانشجویان و افرادی خواهند بود که به حوزه امنیت اطلاعات علاقمند هستند.

برای ارسال مقالات جهت چاپ در نشریه به @elahe_rahbaran در تلگرام پیام دهید.