بسمه تعالی

وزارت ارتباطات و فناوری اطلاعات
سازمان فناوری اطلاعات ایران
معاونت امنیت فضای تولید و تبادل اطلاعات

مرکز ماهر
مرکز مدیریت امداد و هماهنگی عملیات رخدادهای رایانه ای

# معرفی سامانه فریب

## آپا دانشگاه یزد

## سیستم فریب چیست ؟

فناوری فریب یک استراتژی است که مجرمان سایبری که قصد دسترسی به اطلاعات و داده‌های یک شرکت دارند را به سمت یک تله یا طعمه decoy هدایت می‌کند. این فناوری سرورها، اپلیکیشن‌ها و داده‌ها را چنان شبیه‌سازی می‌کند که مجرمان سایبری به این باور که به سیستم و اطلاعات شرکت دسترسی دارند می‌رسند.

## هدف سیستم فریب چیست

هدف سیستم‌های فریب این است که از نفوذ مجرمان سایبری به سیستم یا شبکه و ایجاد خسارت جلوگیری کند.

## روش کار سیستم‌های فریب

روش کار این تکنولوژی این است که تله‌ها و طعمه ههای فریهب را سرتاسر زیرساخت ایجاد می‌کند.
این طعمه‌ها می‌توانند در یک سیستم عامل واقعی و یا شبیه‌سازی شده اجرا شوند و طراحی شده اند تا مجرمان سایبری را چنان فریب دهند که انگار آنها راهی برای نفوذ یافته‌اند.
زمانی که تله کار خودش را کرد ، اطلاعیه هایی به سرور مرکزی فریب ارسال می‌شود که شامل طعمه آلوده شده و حمله‌ای که مجرم سایبری انجام داده است می‌باشد.

## مطالعه در سیستم فریب

منظور از مطالعه این است که تیم IT می‌توانند رفتار مجرمان سایبری را آنالیز کنند. رفتار مجرمان سایبری شامل چگونگی رد شدن از لایه‌ی امنیتی شرکت و وارد شدن به سیستم و تلاش برای دزدیدن اطلاعات مهم و مورد نیاز آنها است.
برخی از شرکت‌ها سرور هایی به عنوان سرور فریب دارند که تمام حرکات مشکوک مجرمان سایبری را ثبت و ضبط می‌کنند و از این اطلاعات برای مطالعه به هدف افزایش امنیت سیستم‌ها استفاده می‌کنند.

## ویژگی منفی فناوری‌های فریب

مجرمان سایبری توانایی این را دارند که حمله‌های خود را در اندازه و مقیاس‌های گوناگونی تنظیم کنند. در این حالت ممکن است اندازه و مقیاس حمله از آنچه سیستم‌های فریب پیش بینی می‌کردند بیشتر باشد و این سیستم‌ها نتوانند حمله را کنترل کنند. در این صورت حمله کنندگان از تقلبی بودن داده‌ها و وجود سیستم فریب آگاه شده و حمله را به سرعت متوقف می‌کنند.

## دلایل اهمیت فناوری‌های فریب

- کاهش attacker dwell time در شبکه

- سرعت بخشیدن به میانگین بازه زمانی بین شناسایی حمله و خنثی کردن آن
- کاهش alert fatigue
- دستیابی هر چه زودتر به post breach detection
- کاهش false positive و ریسک‌ها
- مقیاس‌پذیری
- قابل استفاده در سیستم‌های اینترنت اشیا

## معرفی چند اصطلاح قبلی

- Attacker dwell time: بازه ی زمانی بین نفوذ مجرمان سایبری و شناسایی آن توسط سیستم فریب
- Alert fatigue: هشدارهای بسیار زیادی که به تیم IT داده میشود که اکثر آنها یا اشتباه هستند و یا اصلا ضروری نیستند.
- Post breach detection: مطالعه ی داده و سیستم بعد از وقوع حمله برای جمع آوری اطلاعات و جلوگیری از وقوع حمله در آینده
- False positive: جواب مثبتی که در واقع منفی است!

### Honeypots

یک honeypot شامل یک پایگاه داده ی بزرگ که دارای یوزرنیم و پسورد و دیگر اطلاعات تقلبی است می‌باشد. استراتژی کلی این روش این است که وقتی حمله کننده به شبکه دسترسی پیدا کرد، مسیری را طی کند که از ورودی شبکه به Honeypot است.
وقتی حمله کننده وارد honeypot شد، به تیم IT هشداری داده میشود و حمله خنثی میشود به دلیل اینکه مقیاس و اندازه ی حمله‌ها همواره در حال افزایش است یک honeypot به تنهایی توانایی مقابله با آنها را ندارد. به همین دلیل honeypot ها به تنهایی نمی توانند امنیت یک سیستم را تضمین کنند و حتی توانایی آماده‌سازی داده ی کافی برای کمک به تیم IT را هم ندارند.

### Dynamic Deception

یکی از عوامل مهم یک فناوری فریب موفق این است که باید برای حمله کنندگان ، جدید و غیر قابل تشخیص باشد. برای این کار از هوش مصنوعی و یادگیری ماشین استفاده می‌شود تا محیط پویا باشد. استفاده از هوش مصنوعی باعث می‌شود که تیم IT دیگر نیاز نباشد به طور مداوم سیستم فریب را طراحی کنند.

## سیستم منبع باز

اکنون سراغ معرفی یک سیستم منبع باز می‌رویم که از فناوری فریب استفاده می‌کند. این سیستم DejaVu نام دارد و توسط Bhadresh Patel وHarishRamadoss در سال ۲۰۱۸ تاسیس شده است.

## DejaVu چیست ؟

DejaVu یک فریم‌ورک منبع باز بر اساس فناوری فریب است که با استفاده از این فریم‌ورک می‌توان طعمه‌ها را در سرتاسر زیرساخت پیاده‌سازی کرد.
با استفاده از این فریم‌ورک حتی می‌توان چندین طعمه ی تعاملی(بین کلاینت و سرور) در VLANهای متفاوت در سرتاسر شبکه پیاده‌سازی کرد. برای پیاده‌سازی، مدیریت و کانفیگ کردن طعمه‌ها یک پلت‌فرم تحت وب ایجاد شده است. برای پیاده‌سازی این پلت‌فرم، بیشتر از زبان‌های PHPو JavaScript استفاده شده است.

## هشدار alert ها در DejaVu

هشدارها زمانی بوجود می‌آیند که حریف درگیر این طعمه‌ها شده باشد. وقتی که مجرم سایبری به قصد شناسایی و یا تلاش برای انجام احراز هویت وارد این طعمه‌ها می‌شود، این فریم‌ورک یک هشدار بوجود می‌آورد که این هشدار باید توسط بخش دفاع defense بررسی شود.

One of the major advantages of DejaVu is Using a single platform you can deploys decoys across different VLANS and manage, monitor them.

## چندین مثال برای عملکرد DejaVu

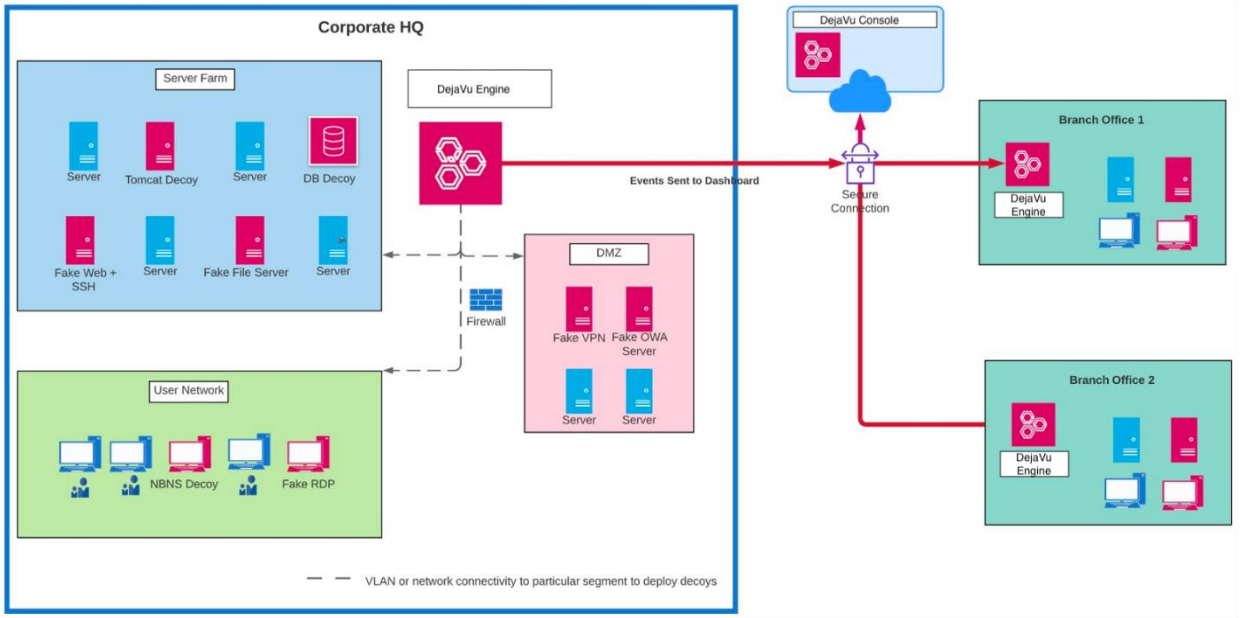
- Password Spray/ Brute Force Attack: منظور از password spray این است که مجرم سایبری بخواهد با استفاده از پسوردهایی که رایج نیستند به تعداد زیادی اکانت دسترسی داشته باشد.

- Attacker targeting low hanging fruits Tomcat/ MSSQL: منظور از low fruits hanging سیستم هایی است که نفوذ به آنها نسبت به بقیه سیستم‌ها آسانتر است

- LLMNR Poisoning: زمانی که هاست نتواند با استفاده از DNS به ای پی دسترسی پیدا کند، سراغ LLMNR یا به عبارتیLink Local Multicast Name Resolution می‌رود. حمله کنندگان در این میان چنان رفتار می‌کنند که درخواست هاست را شناسایی کرده‌اند و بدین گونه وارد سیستم می‌شوند.

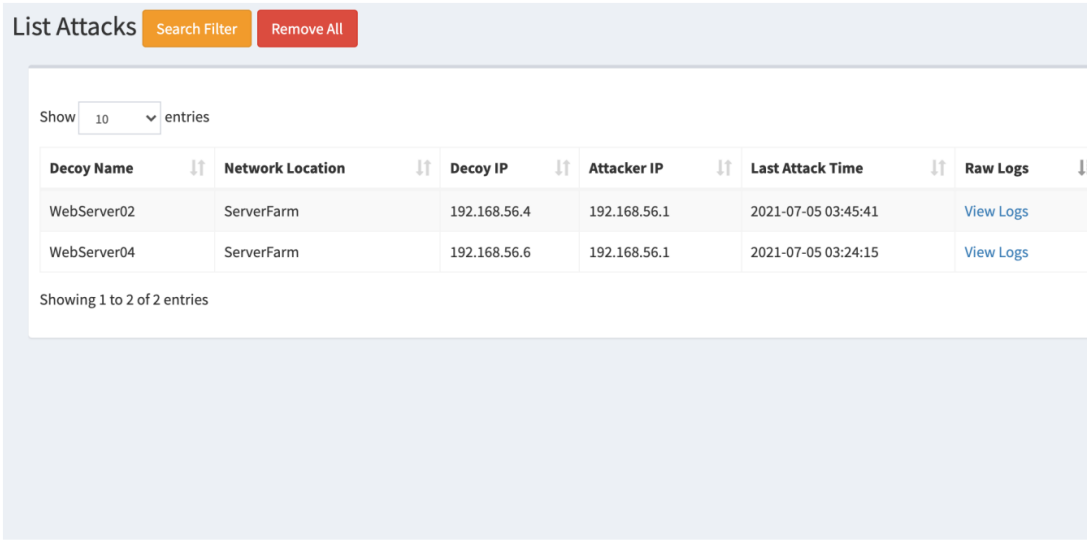
- Data Ex filtration: خارج کردن داده‌های حیاتی و مهم یک ســـازمان از داخل به خارج بدون اجازه

## معماری DejaVu

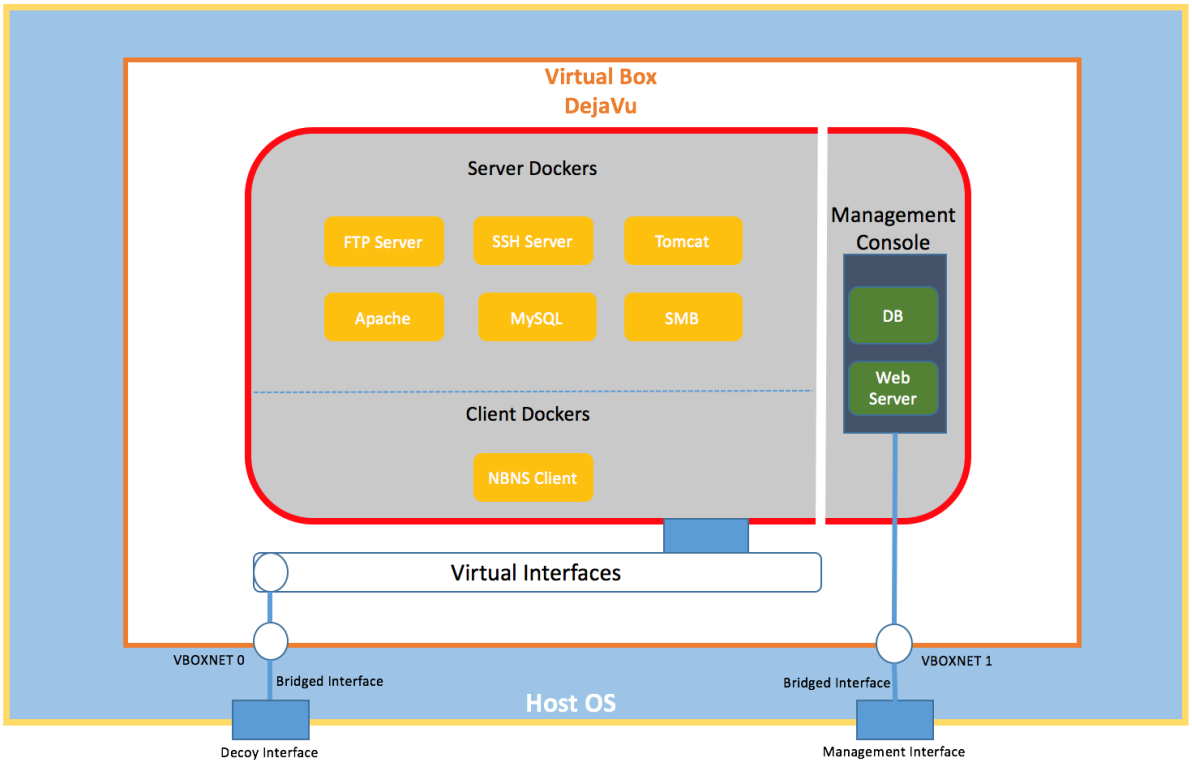



## توضیحاتی پیرامون معماری

- DejaVu Engine: برای پیاده‌سازی طعمه‌ها در سرتاسر زیرساخت استفاده می‌شود.
- DejaVu Console: برای نظارت و مدیریت هشدارهایی که توسط موتورهای DejaVu بوجود می‌آید.




## نگاهی دقیق تر به DejaVu




## توضیحات پیرامون اسلاید قبل

- Host OS: سیستم عاملی که Deja Vu را میزبانی م یکند.

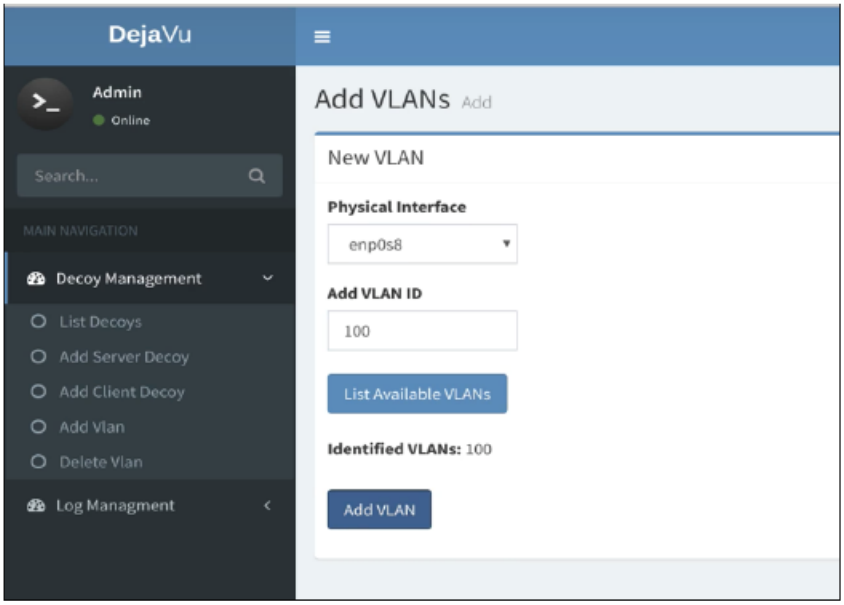
- Management Interface: برای دسترسی به web based management console
- Decoy Interface: رابط برای اتصالات ورودی از شبکه به طعمه ها
- Virtual Interface: با Decoy Interface پل شده اند تا ترافیک را به سمت طعمه هدایت کنند.
- Management Console: برای پیاده‌سازی، مدیریت و کانفیگ کردن طعمه‌ها.

### طعمه‌های سمت سرور

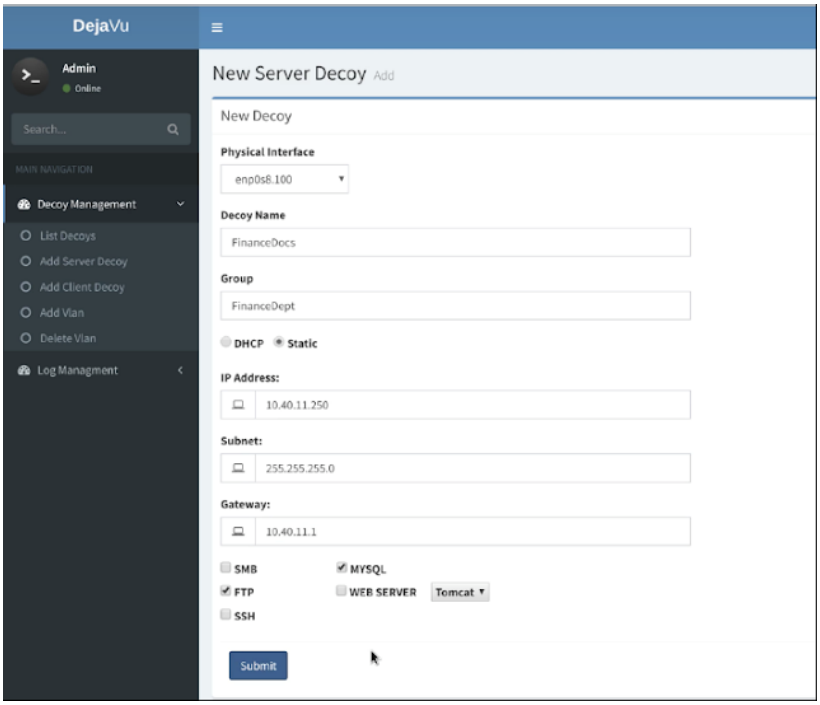
- MYSQL
- Custom HTTP Decoy You can configure this with a custom HTML template
- TELNET (allows users to test connectivity to remote machines and issue commands through the use of a keyboard)
- FTP (File Transfer Protocol)
- Web Server Tomcat, Apache, Basic Auth
- SMTP (Simple Mail Transfer
- RDP (Remote Desktop Protocol) Interactive and Non Interactive
- VNC (Virtual Network)

### پیاده‌سازی یک طعمه ی سرور

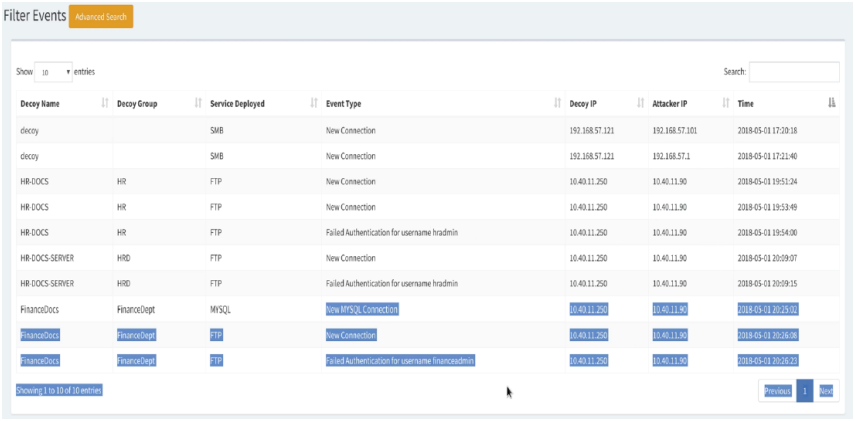
ابتدا باید یک VLAN اضافه کنیم. در منوی decoy management گزینه یaddVLAN آیدی را وارد کرده و List vailable VLANs را می‌زنیم.




در منوی decoy management گزینه ی add server decoy اطلاعات مربوط به طعمه موردنظر را وارد می‌کنیم.



## مشاهده ی هشدارها

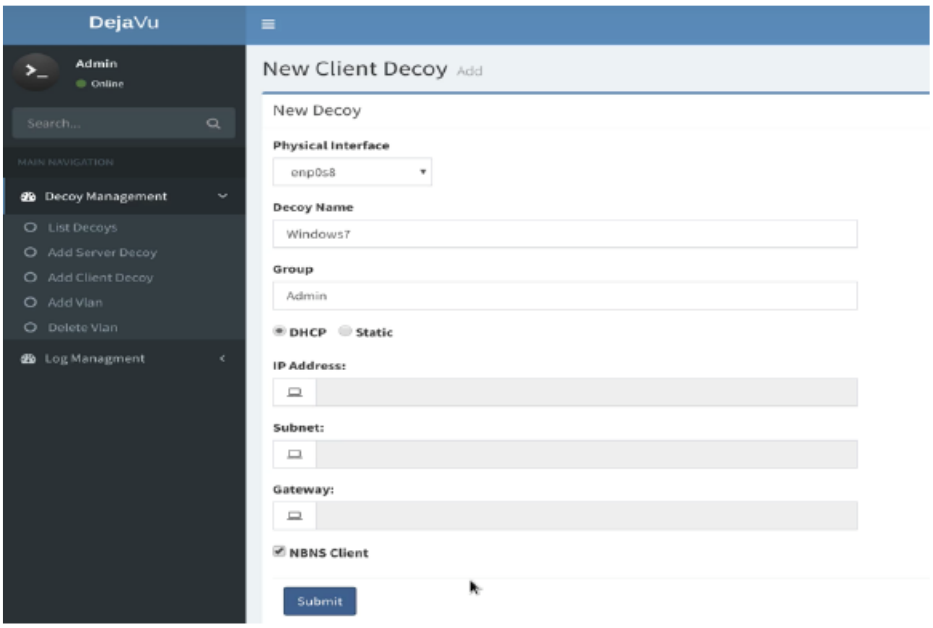



## طعمه‌های سمت کلاینت[۲]

- NBNS Decoy (NetBIOS Name Server)
- MITM Decoy (man in the middle)
- SSDP Client (Simple Service Discovery Protocol)
- Email Client

## پیاده‌سازی یک طعمه‌ی سمت کلاینت

در منوی decoy management گزینه‌ی add client decoy اطلاعات مربوط به طعمه موردنظر را وارد می‌کنیم.



## منابع


1. https://www.fortinet.com/resources/cyberglossary/what-is-deception-technology
2. https://github.com/bhdresh/Dejavu
3. https://www.kitploit.com/2018/06/dejavu-open-source-deception-framework.html
4. https://www.forcepoint.com/cyber-edu/deception-technology


سیستم‌های فریب فعلی : در دهه ۲۰۱۰ بسیاری از کمپانی‌ها به اهمیت سیستم‌های فریب پی برده و درصدد توسعه‌ی آنها برآمدند. همچنین سیستم‌های فریب توزیع شده، در این زمان پدید آمده است.

بنظر می‌رسد با رشد و تحقیق و توسعه‌ی علوم کامپیوتر، حوزه سیستم‌های فریب نیز پیشرفت زیادی کرده‌اند و ضعف‌های پیشین آنها برطرف گشته‌است.

### معرفی سیستم فریب متن باز DejaVu:

این سیستم که در سال ۲۰۱۸ در گیت‌هاب عرضه شد، پروژه متن‌بازی است که سازمان‌های زیادی از آن استفاده می‌کنند. از زمان عرضه تا به امروز، تله‌های بسیار بیشتری در آن قرار داده شده‌است.

فریم‌ورک متن باز DejaVu امکان پیاده‌سازی تله‌های بسیاری را به استفاده کننده می‌دهد. این تله‌ها می‌توانند با توجه به استراتژی وی، در سرتاسر شبکه در سمت سرور و همچنین سمت کلاینت قرار داده شوند. برای ساده‌کردن هرچه بیشتر استفاده از این فریم‌ورک، پلت‌فرم مبتنی بر وب نیز برای آن طراحی شده که می‌تواند پیاده‌سازی، مدیریت و پیکربندی شود. بدین معنی که تمام کارهای گفته‌شده، می‌تواند از طریق یک کنسول مرکزی انجام شده و سخت بودنِ مدیریت که یکی از مشکلات نسل‌های پیشین سیستم‌های فریب بوده، در این فریم‌ورک وجود ندارد.

در این فریم‌ورک داشبورد ورود به سیستم و همچنین اخطار به خوبی پیاده‌سازی شده و تمام logهای سیستم، برای استفاده کننده مشخص است. وی می‌تواند با توجه به نیازهای خود، آنها را شخصی‌سازی کرده و اخطارها را مدیریت کند. به عنوان مثال اگر IP مدیر سیستم برای بررسی نیاز به ایجاد حملات بود، می‌توان آنرا به سیستم اضافه کرد تا از وقوع false posetive جلوگیری شود و بار سیستم سبک گردد. همچنین همانطور که در بخش‌های پیشین توضیح داده شد، این مسئله می‌تواند به بالا بردن کارایی تیم امنیت شبکه و نرم‌افزار هم کمک کند. زیرا false posetive معمولا کارایی آنها را پایین می‌آورد.

در این سیستم، اخطار تنها وقتی ارسال می‌شود که فرد مهاجم به سیستم، واقعا با یکی از تله‌های پیاده‌سازی شده توسط ادمین، درگیر شود. مثلا وقتی هکر سعی میکند تایید هویت انجام دهد، سیستم می‌تواند با دقت بالا این اخطار (alert) را به تیم امنیت ارسال کرده و آنها را از وجود مشکل آگاه سازند.

تله‌های پیاده‌سازی شده می‌توانند در سمت کلاینت قرار داده‌شوند تا از حملات دسته‌ی LLMNR جلوگیری شود. همچنین در حملاتی که مهاجم سعی در دستکاری اطلاعات سرور دارد (مانند SQLi) ، تله‌های نام‌برده می‌توانند در شناسایی حملات و فعال‌سازی مکانیزم‌های دفاعی شبکه و سیستم، کارا باشند.

طبق گفته‌ی توسعه‌دهنده‌های این فریم‌ورک، مزیت اصلی و اساسی آن، ایجاد تله و مدیریت و مانیتور کردن آنها در یک پلت‌فرم واحد می‌باشد.

### تله‌های پیاده‌سازی شده در فریم‌ورک DejaVu:

همانطور که در بخش قبل نیز گفته شد، از زمان عرضه تا به امروز توسعه‌دهنده‌های این فریم‌ورک با بررسی نظرات سازمان‌هایی که از آن استفاده کرده‌اند، آن را بهبود و ارتقا داده‌اند. طبق آخرین آپدیت عرضه شده برای

آن در تاریخ ۲۰۲۱/۱۰/۴ لیست تله‌های موجود که هرکدام بسته به استراتژی می‌توانند پیاده‌سازی شوند، به شرح زیر می‌باشد:

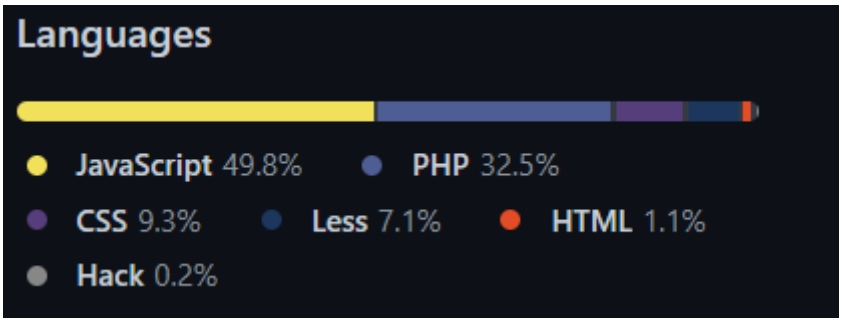
- Server Decoys
  - MYSQL
  - SNMP
  - Custom HTTP Decoy - You can configure this with a custom HTML template
  - TELNET
  - SMB Server with custom files
  - FTP
  - TFTP
  - Web Server - Tomcat, Apache, Basic Auth
  - SSH Interactive and Non-Interactive
  - SMTP
  - RDP Interactive and Non-Interactive
  - VNC
  - HONEYCOMB (To capture events from Honey Docs)
  - ICS/SCADA Decoys - Modbus and S7COMM
- Client Decoys
  - NBNS Decoy
  - MITM Decoy
  - SSDP Client
  - Email Client
- BreadCrumbs
  - Honey Docs
  - HoneyHash - Injects creds into memory
  - Kerberoast Honey Account

**بررسی معماری و کد فریم‌ورک:**

همانطور که گفته‌شد، این فریم‌ورک متن‌باز بوده و از طریق آدرس زیر در دسترس می‌باشد:

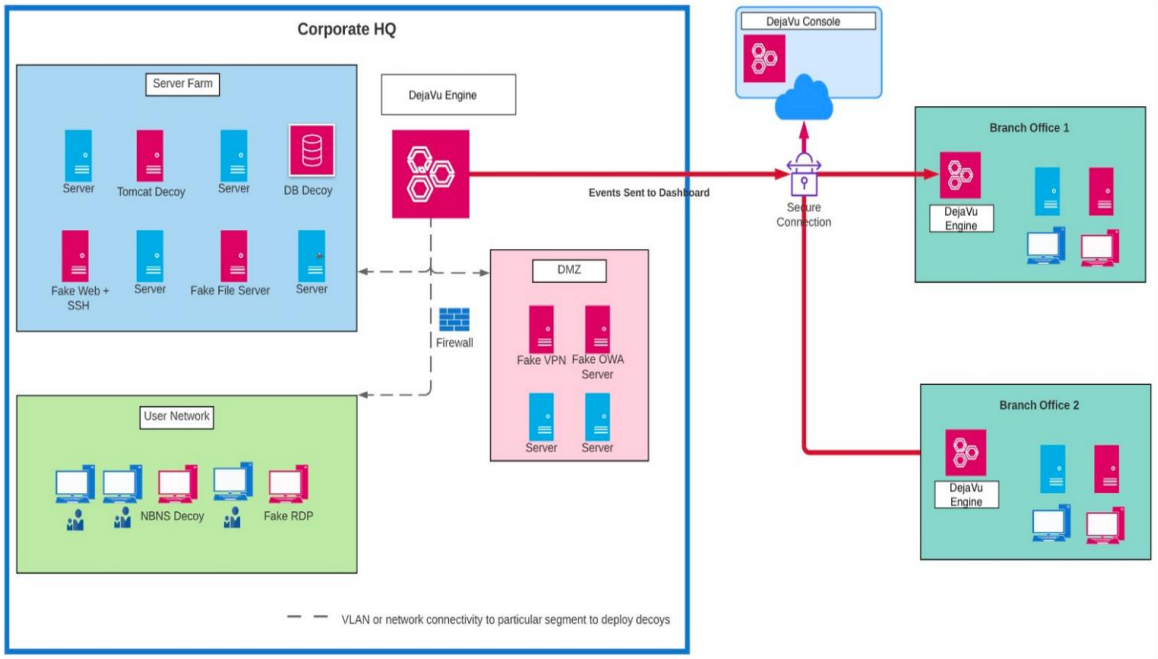
https://github.com/bhdresh/Dejavu

بخش اصلی توسعه‌ی این سیستم در وحله اول به javascript و پس از آن به php ، css و بقیه ابزارهای توسعه اختصاص دارد.




شکل ۱ - زبان‌های برنامه‌نویسی استفاده شده در توسعه فریم‌ورک DejaVu

این پروژه با 300 start و ۸۳ fork محبوبیت بالایی داشته و بنظر می‌رسد افراد زیادی از آن استفاده کرده و حتی ورژن شخصی‌سازی شده‌ی خود را بر مبنای آن توسعه داده‌اند.




شکل ۲ - معماری پیاده‌سازی شده سیستم فریب *DejaVu*

طبق معماری ساخته‌شده برای این سیستم، DejaVu Engine بخش اصلی آن می‌باشد. برای همین افراد استفاده کننده، می‌توانند چندین office را ساخته و با استفاده از DejaVu Engine تله‌های موردنظرشان را در آن پیاده‌سازی کنند. داشتن چندین office از این جهت اهمیت دارد که با گسترش شبکه می‌توانیم از طرفی سیستم خود را توزیع‌پذیرتر کرده و از طرفی دیگر، مدت زمان ساکن شدن مهاجم در سیستم را افزایش داده و با استفاده از این فریم‌ورک، رفتار وی را مانیتور کنیم.

بخش مهم دیگر این سیستم، DejaVu Console می‌باشد. طبق توضیحات توسعه‌دهندگان این پروژه، این بخش مانند داشبورد در سیستم‌های مدیریت عمل می‌کند. به بیان دیگر، می‌توانیم با استفاده از یک کنسول مرکزی، تمام بخش‌های این سیستم فریب را مدیریت و نظاره کنیم. وجود سیستم مرکزی، یکی از نقاط قوت اصلی این سیستم فریب می‌باشد.

منابع


1. https://www.wwt.com/article/deception-technology
2. https://www.countercraftsec.com/blog/post/whats-real-difference-between-cyber-deception-and-honeypots/
3. https://github.com/bhdresh/Dejavu
4. https://en.wikipedia.org/wiki/Honeypot_(computing)