XXI SBSeg 2021
Belém - PA

LaTARC Research Lab

Freki

Uma Ferramenta para Análise Automatizada de Malware

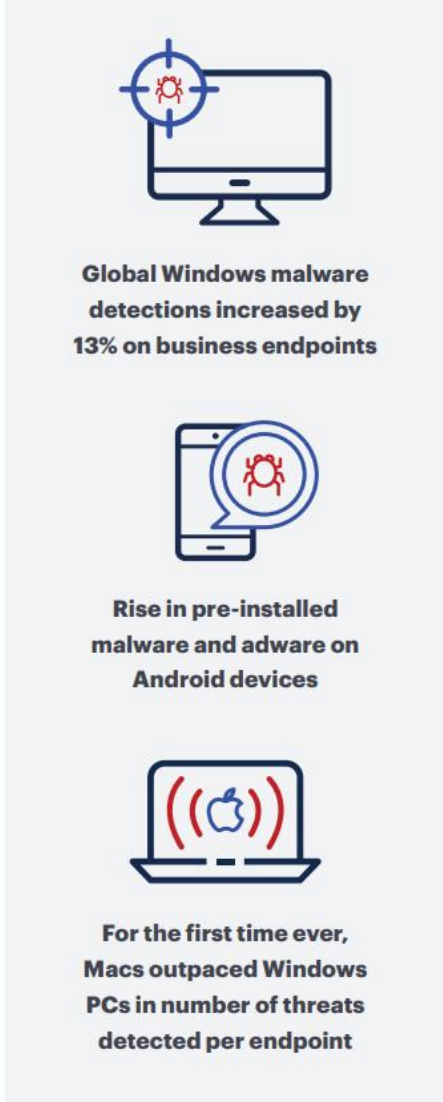
Cristian Souza - IFRN
Felipe Dantas - IFRN

Realização: SBC Sociedade Brasileira de Computação · UFPA

Agenda

1 – Introdução
2 – Objetivos
3 – O Freki
4 – Funcionalidades
5 – Demonstração
6 – Conclusão

Introdução



- Infecções por *malware* continuam sendo uma das principais ameaças aos sistemas computacionais.
- Em 2020 foi constatado um aumento de 13% nas detecções de *malware* focado no setor corporativo.
- Atividades de *ransomware* estão sendo cada vez mais utilizadas por cibercriminosos.
- Infecções por *adwares* em dispositivos móveis cresceram significativamente.

Fonte: https://www.malwarebytes.com/resources/files/2020/02/2020_state-of-malware-report-1.pdf

Introdução

- Visando entender o funcionamento do programa malicioso, são executadas análises estáticas e dinâmicas.

- Análise estática: coleta de informações sobre o binário sem executá-lo.

  Exemplos: Tipo de arquivo, *hashes, strings*, cabeçalhos, seções…

- Análise dinâmica: execução do artefato em um ambiente controlado.

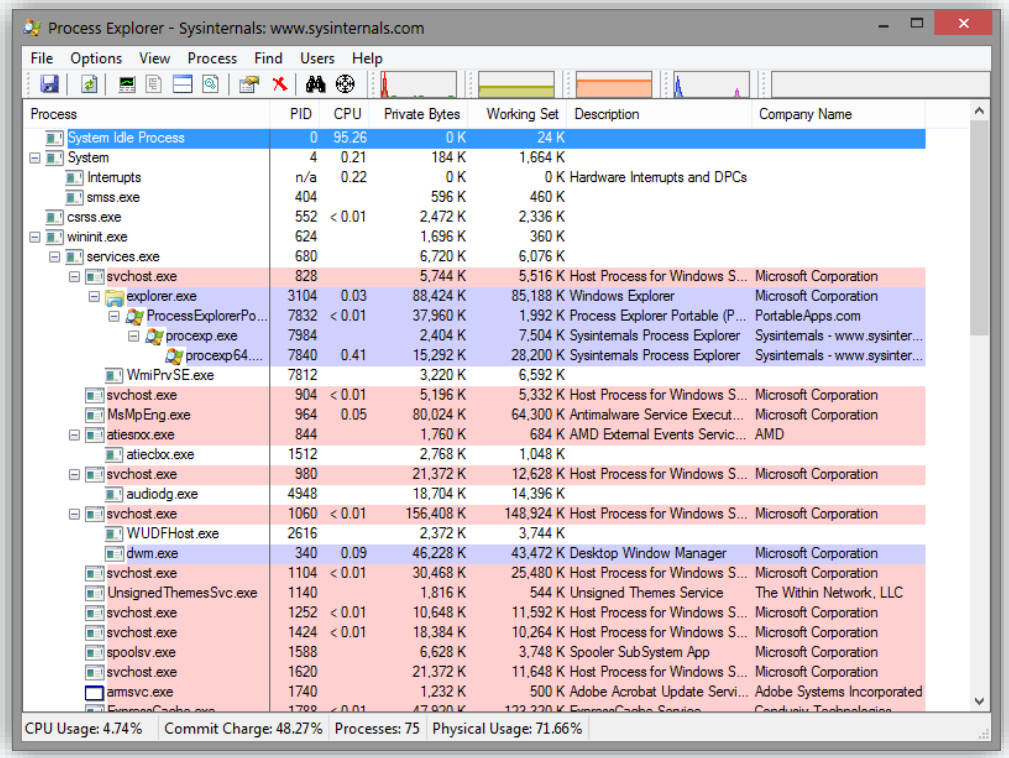
  Exemplos: Requisições, arquivos baixados, arquivos criados, chamadas de sistema, processos…

Introdução

- Análise estática:

```
$ sha256sum mimikatz.exe
92804faaab2175dc501d73e814663058c78c0a042675a8937266357bcfb96c50  mimikatz.exe
$ xxd mimikatz.exe | head -n 20
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ..............
00000010: b800 0000 0000 0000 4000 0000 0000 0000  ........@.......
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 2001 0000  ............ ...
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468  ........!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f  is program canno
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320  t be run in DOS 
00000070: 6d6f 6465 2e0d 0d0a 2400 0000 0000 0000  mode....$.......
00000080: 34c8 d822 70a9 b671 70a9 b671 70a9 b671  4.."p..qp..qp..q
00000090: 79d1 2371 72a9 b671 79d1 3571 4fa9 b671  y.#qr..qy.5qO..q
000000a0: 79d1 3271 60a9 b671 79d1 2571 72a9 b671  y.2q`..qy.%qr..q
000000b0: 6b34 2a71 72a9 b671 1647 7d71 74a9 b671  k4*qr..q.G}qt..q
000000c0: eb42 7d71 72a9 b671 0634 db71 72a9 b671  .B}qr..q.4.qr..q
000000d0: 6efb 3271 72a9 b671 0634 cd71 5fa9 b671  n.2qr..q.4.q_..q
000000e0: 70a9 b771 0aab b671 576f c871 71a9 b671  p..q...qWo.qq..q
000000f0: 79d1 3f71 13a9 b671 79d1 2271 71a9 b671  y.?q...qy."qq..q
00000100: 79d1 2771 71a9 b671 5269 6368 70a9 b671  y.'qq..qRichp..q
00000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000120: 5045 0000 6486 0600 6339 5a5e 0000 0000  PE..d...c9Z^....
00000130: 0000 0000 f000 2200 0b02 0900 002c 0c00  ......".......,..
```

- Análise dinâmica:



Introdução

- Algumas ferramentas automatizadas existentes:

VirusTotal

ANY RUN

MALWARE bazaar by ABUSE.ch

- Deficiências:

  Plataformas fechadas.

  Ausência de *deploy* local.

  Exposição do arquivo analisado.

  APIs limitadas.

Objetivos

- Facilitar a análise de malware e engenharia reversa.
- Prover uma API de fácil utilização em diferentes projetos.
- Fácil implantação: Via Docker.
- Desenvolvimento de uma ferramenta *open-source*.

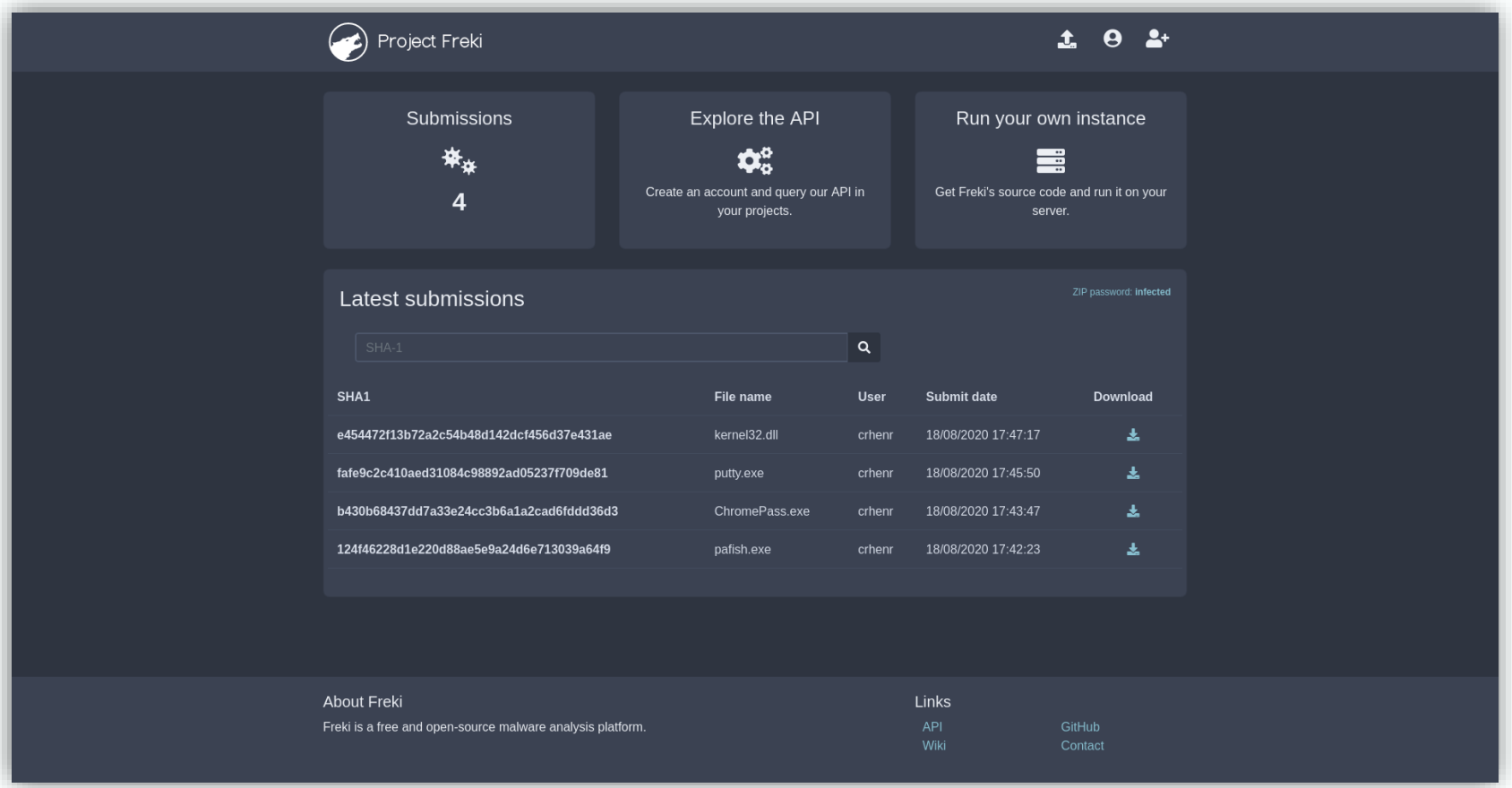
  Novas funcionalidades podem ser facilmente adicionadas pela comunidade.

  Licença AGPL.

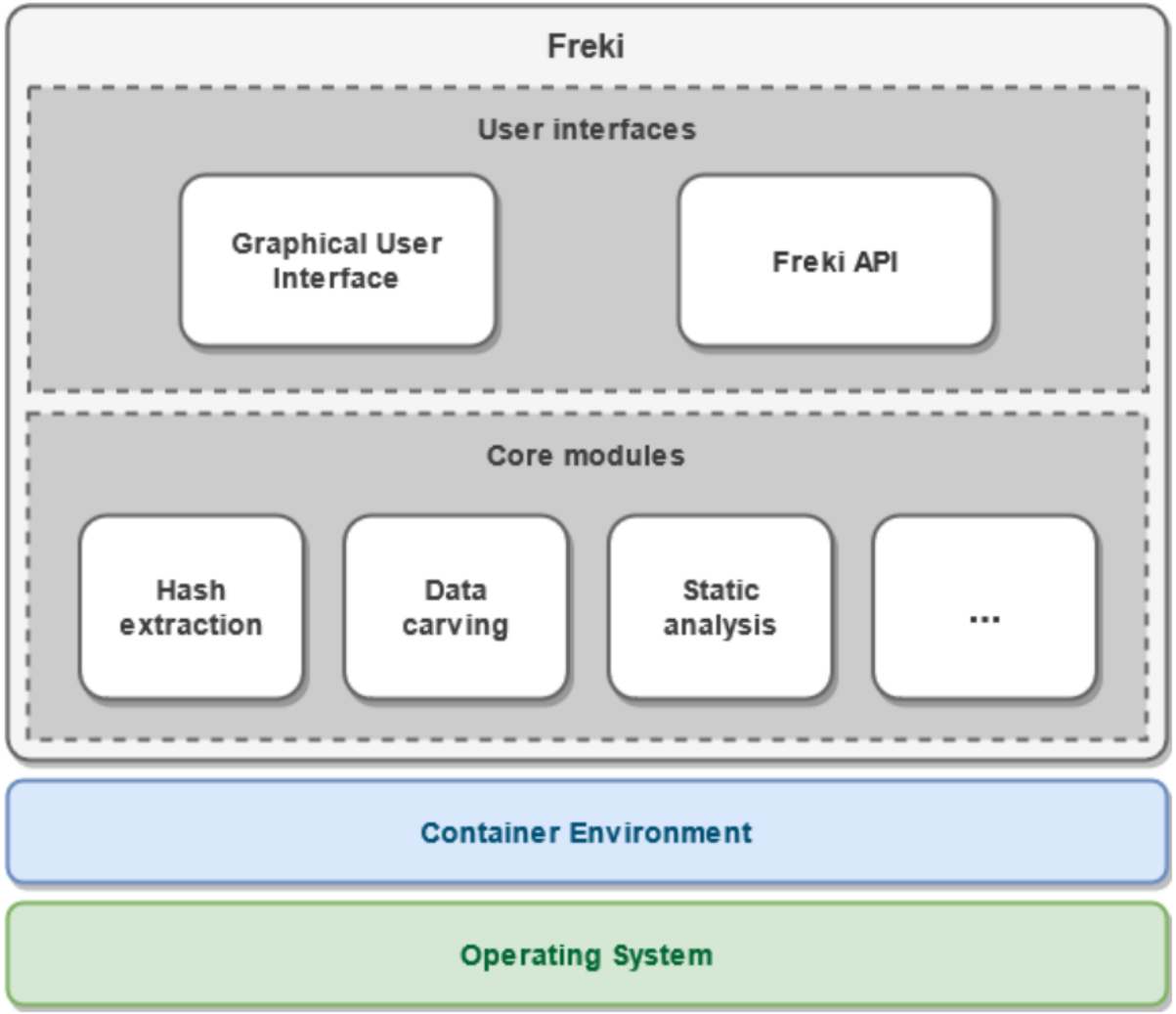
  Código fonte disponível em: https://github.com/crhenr/freki

O Freki

- Plataforma *open-source* para análise de malware e engenharia reversa.



O Freki: Arquitetura



- Principais componentes:

1 - User Interface (UI).

2 - REST API.

3 - Core Modules:

Extração de hashes.

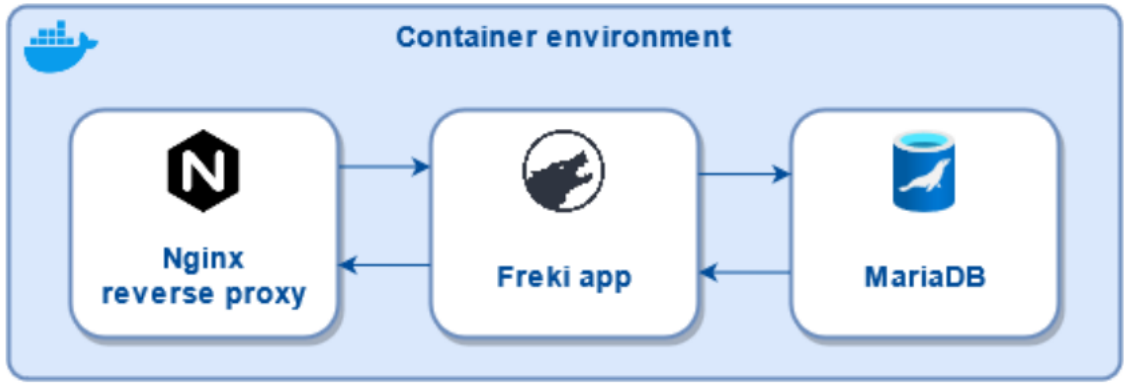
Consultas ao VirusTotal.

Análise estática.

Data carving.

Classificação do binário.

O Freki: Arquitetura



Funcionalidades

- Extração de *hashes*: MD5, SHA-1, SHA-256, SHA-384, SHA-512, CRC32 e SSDEEP.
- Consultas ao VirusTotal: Resultados das análises providas pela ferramenta.
- Análise estática de arquivos PE: Cabeçalhos, seções, importações, capacidades e *strings*.
- Identificação de padrões via Yara.
- Download de artefatos.
- Gerenciamento de usuários e comentários da comunidade.

Funcionalidades

- *Deploy* via Docker:

  1. git clone `https://github.com/crhenr/freki.git`

  2. Configure as credenciais no arquivo `.env` localizado na raiz do projeto

  3. É recomendável habilitar o HTTPS copiando o certificado e a chave privada no diretório `docker/nginx/certs`

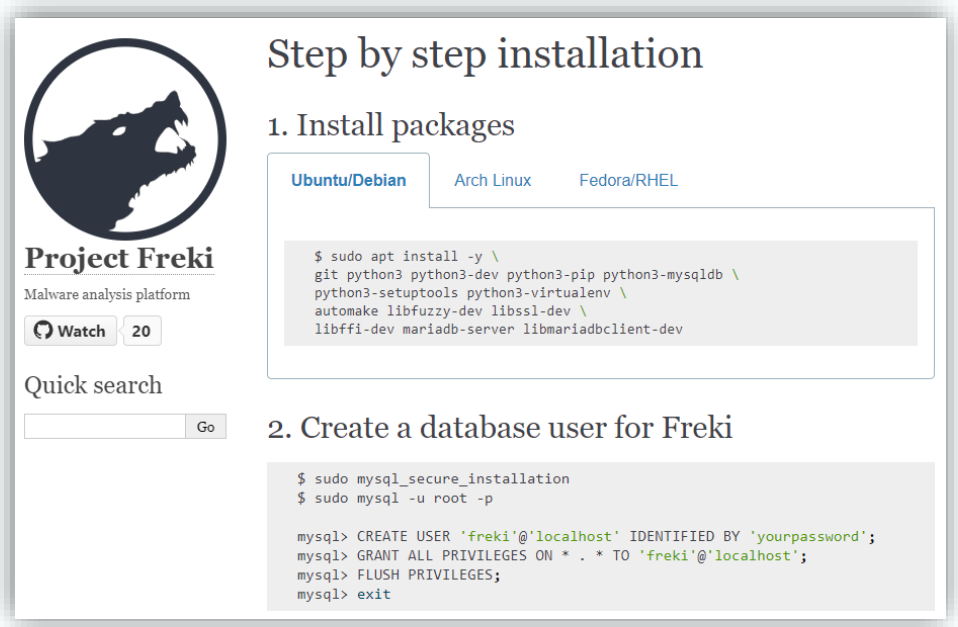
  4. Execute a ferramenta: `make run` ou `docker-compose up -d`

Funcionalidades

```
$ git clone https://github.com/crhenr/freki.git
Cloning into 'freki'...
remote: Enumerating objects: 939, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 939 (delta 7), reused 10 (delta 4), pack-reused 920
Receiving objects: 100% (939/939), 14.10 MiB | 2.96 MiB/s, done.
Resolving deltas: 100% (262/262), done.
$ cd freki
$ docker-compose up -d
Creating network "freki_frekinet" with driver "bridge"
Creating freki_db ... done
Creating freki    ... done
Creating freki_nginx ... done
$ docker container ls
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS          PORTS                                                                                   NAMES
bd0867025782   freki_nginx   "/docker-entrypoint.…"   10 seconds ago   Up 8 seconds    0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp   freki_nginx
3c9be16a638c   freki_freki   "bash -c '/freki/wai…"   12 seconds ago   Up 9 seconds    8000/tcp                                                                                freki
f7cd7cf3ac90   freki_db      "/scripts/startup.sh"    15 seconds ago   Up 12 seconds   3306/tcp                                                                                freki_db
```

Funcionalidades

- Também é possível realizar a instalação nativa da ferramenta.
- Mais detalhes em: https://crhenr.xyz/freki/admin/run-steps.html.
- Distribuições testadas:
  - Ubuntu/Debian
  - Arch Linux
  - Fedora/RHEL



Demonstração

https://www.youtube.com/watch?v=brvNUPgw7ho

Conclusão

- Freki é uma ferramenta livre para análise automatizada de malware e engenharia reversa.
- Sua arquitetura é bastante modular, permitindo a adição de novos recursos com grande facilidade.
- Seu *deploy* via Docker permite a criação de um ambiente para análise de forma simples e rápida.
- Trabalhos futuros:
  - Análise estática de outros tipos de arquivos: ELF, PDF, APK…
  - Criação de um ambiente para análise dinâmica.
  - Criação de um executável *standalone*.
  - Melhorias gerais no código fonte.