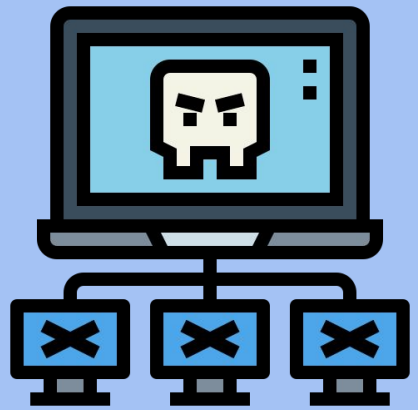
mtracker



co w botnecie piszczy

CERT.PL

$ whoami

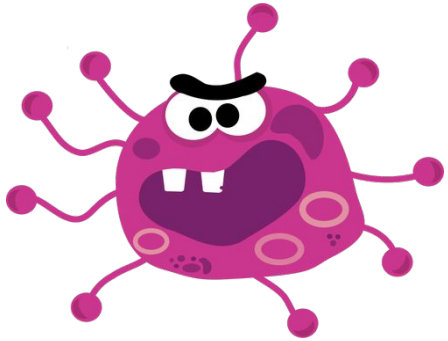
Jarosław Jedynak
msm@cert.pl
jaroslaw.jedynak@cert.pl

$ whoami

Jarosław Jedynak
msm@cert.pl
jaroslaw.jedynak@cert.pl

$ whoami

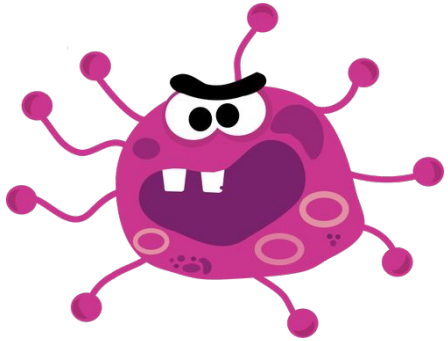
Jarosław Jedynak
msm@cert.pl
jaroslaw.jedynak@cert.pl



CERT.PL>_
CERT.PL>_
CERT.PL>_

$ whoami

Jarosław Jedynak
msm@cert.pl
jaroslaw.jedynak@cert.pl



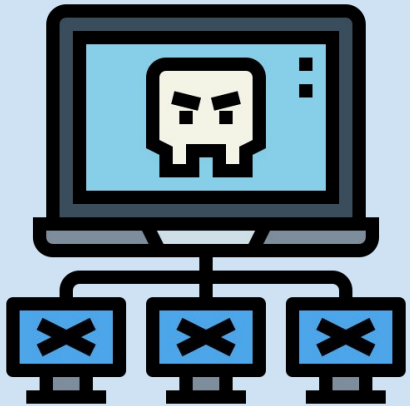
CERT.PL
CERT.PL
CERT.PL

mtracker

Agenda

- Po co komu mtracker
- Co to jest mtracker
- Jak działa mtracker
- Po co tobie mtracker

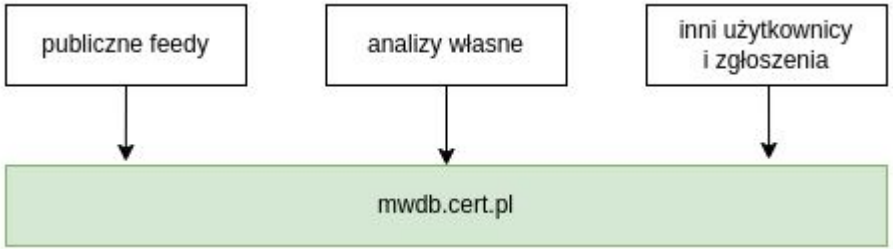
po co komu



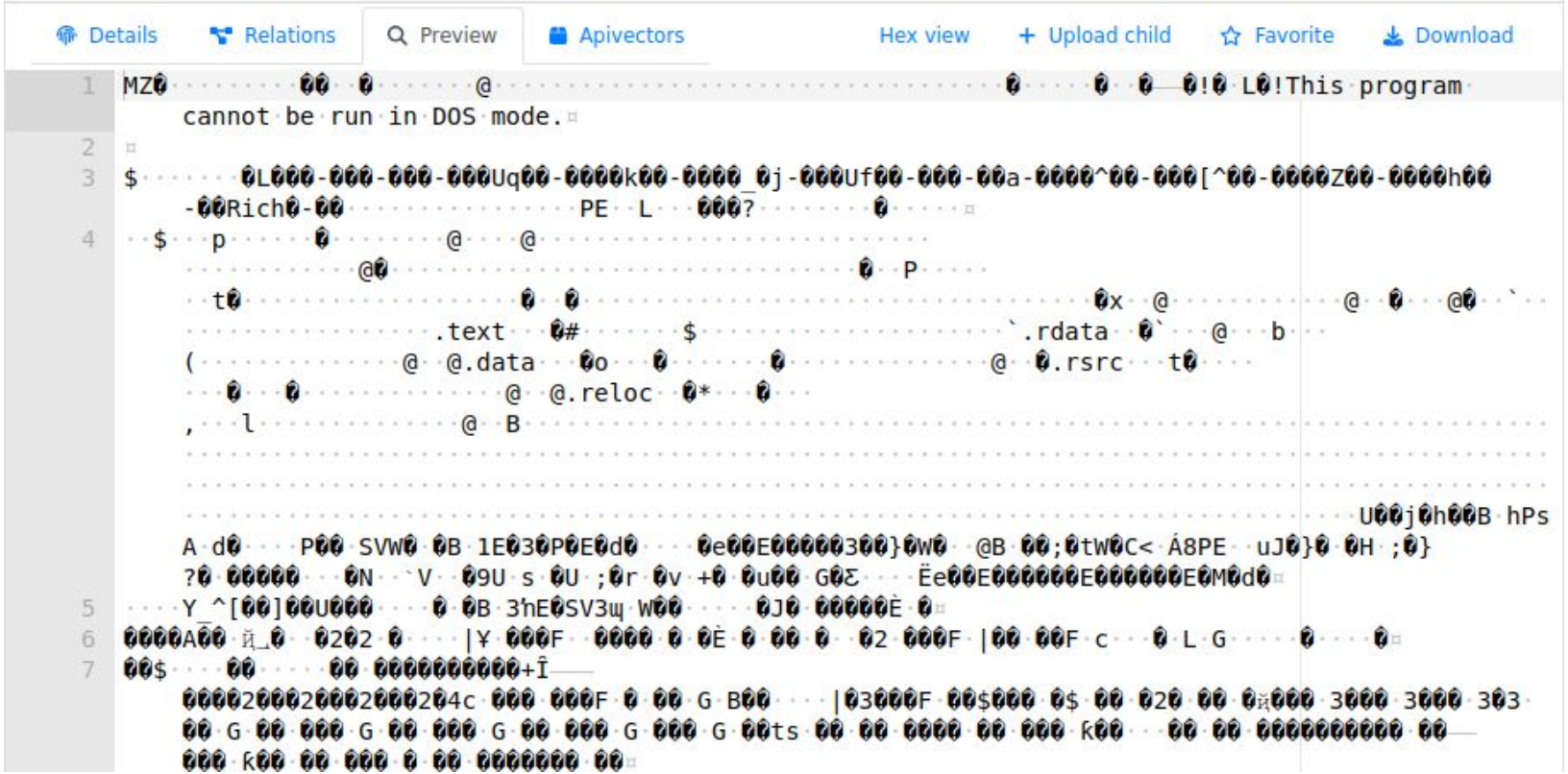
mtracker

CERT.PL

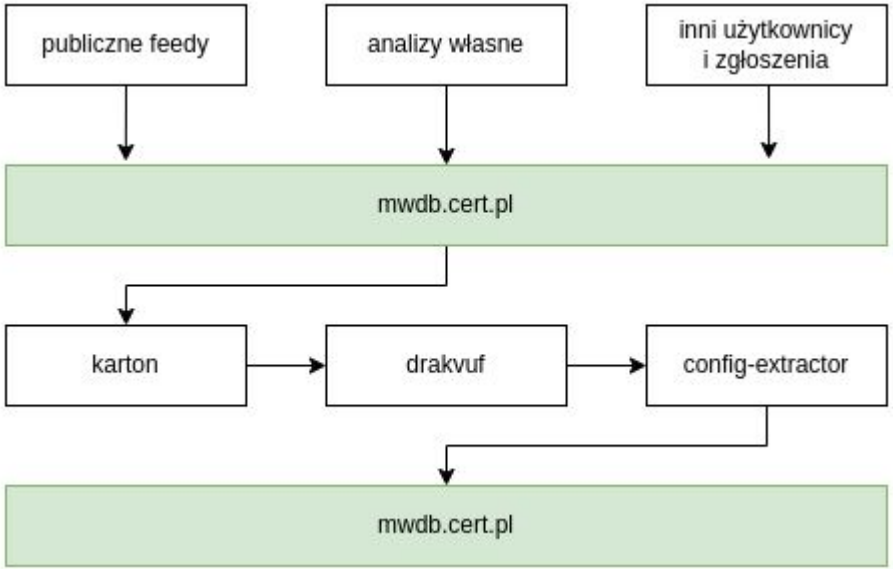
Śledzenie malware w CERT 101



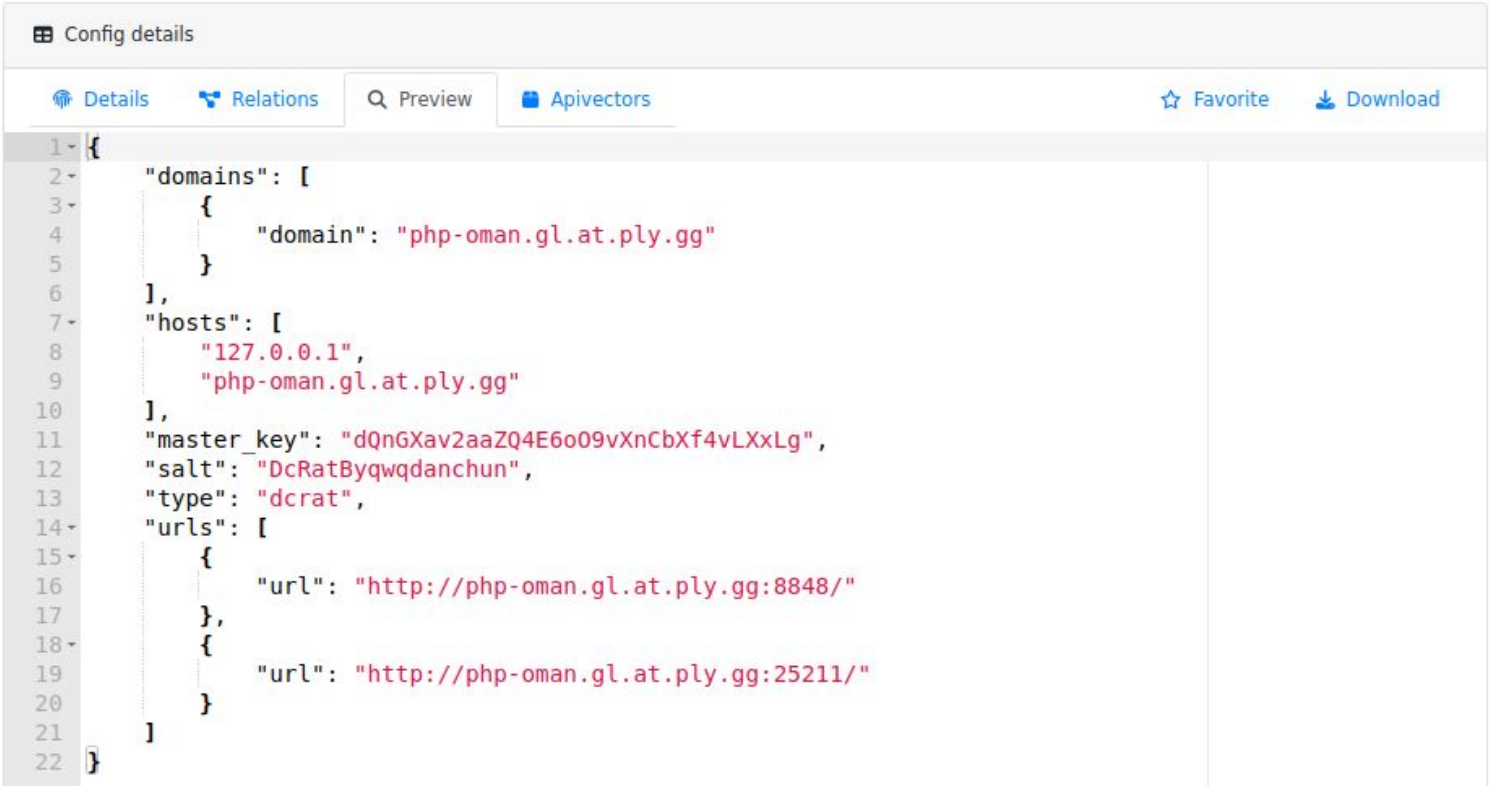
Śledzenie malware w CERT 101



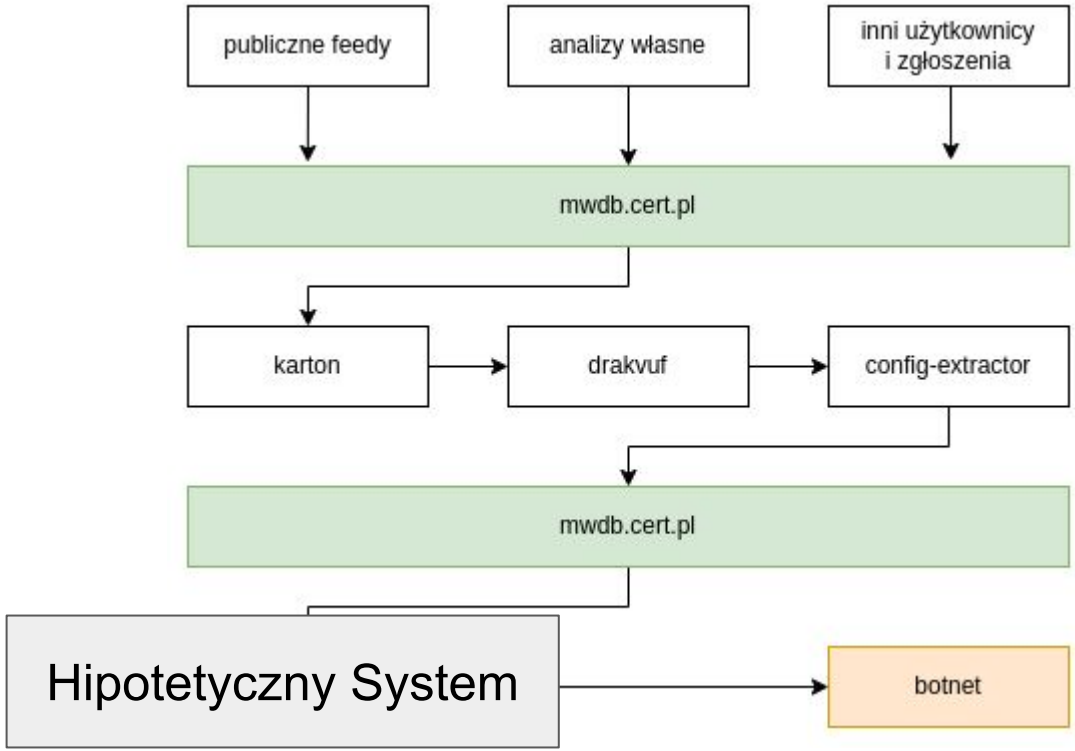
Śledzenie malware w CERT 101



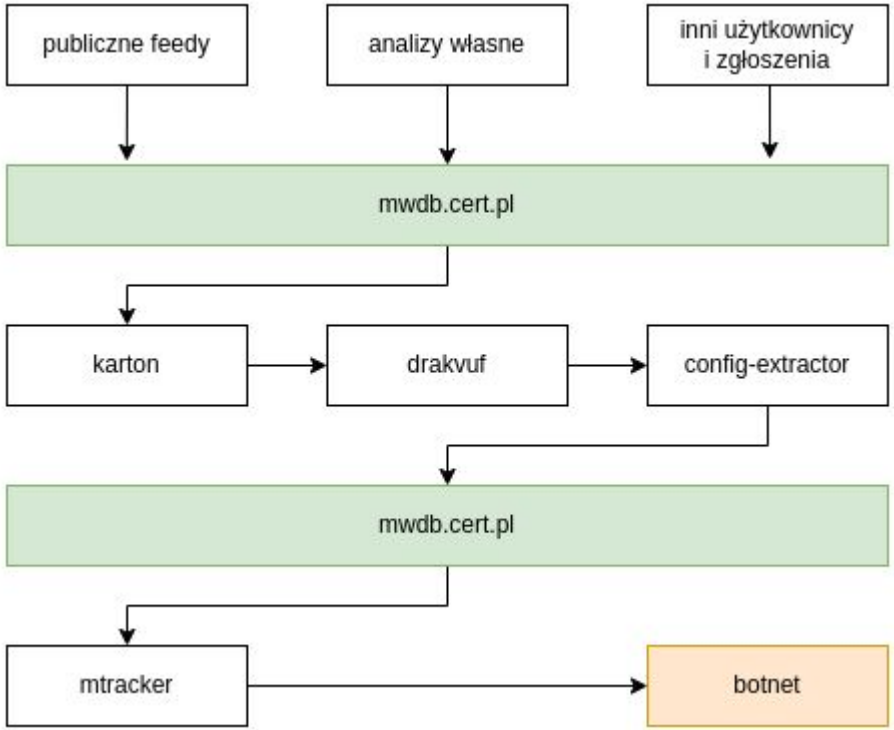
Śledzenie malware w CERT 101



Śledzenie malware w CERT 101

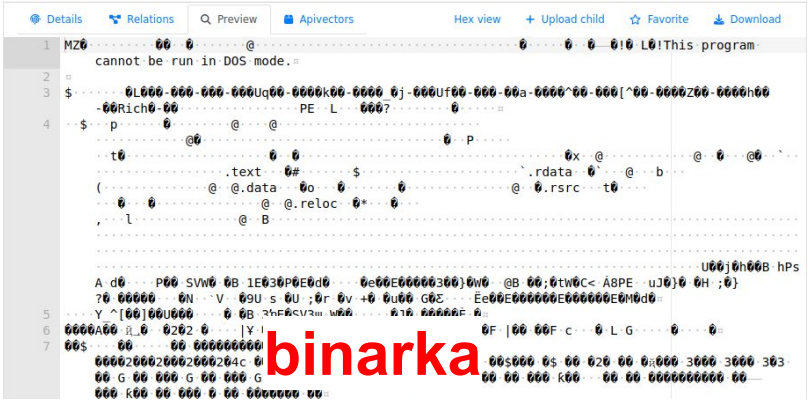


Śledzenie malware w CERT 101

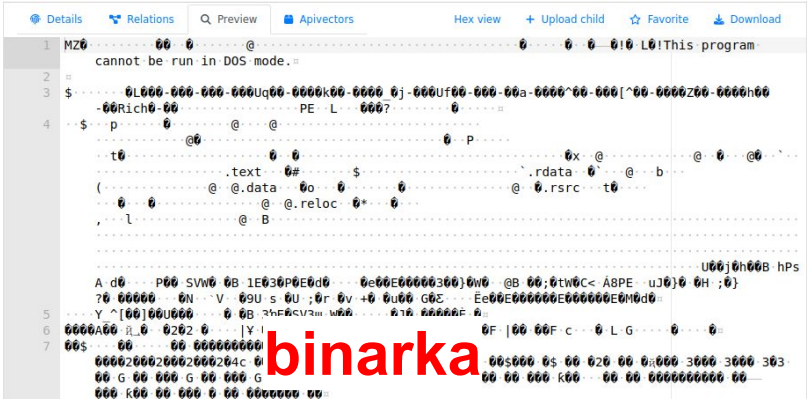


spoiler!

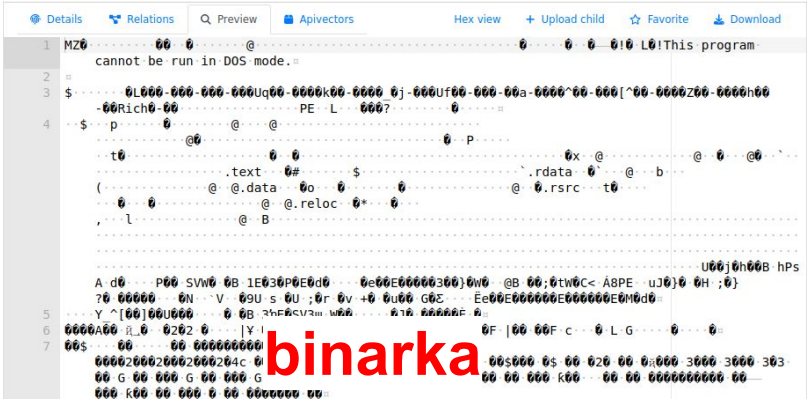
Śledzenie malware w CERT 101

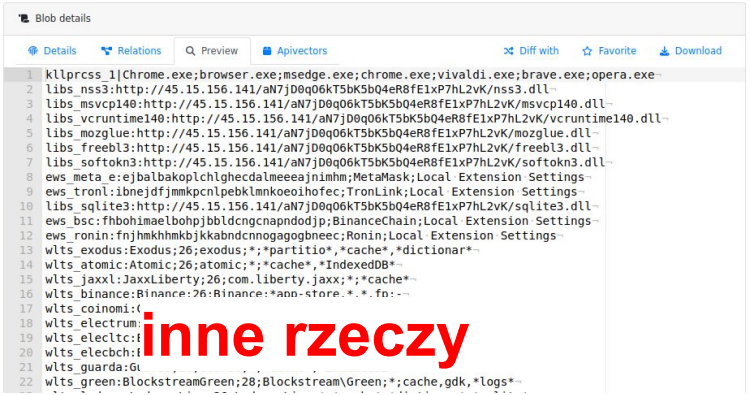


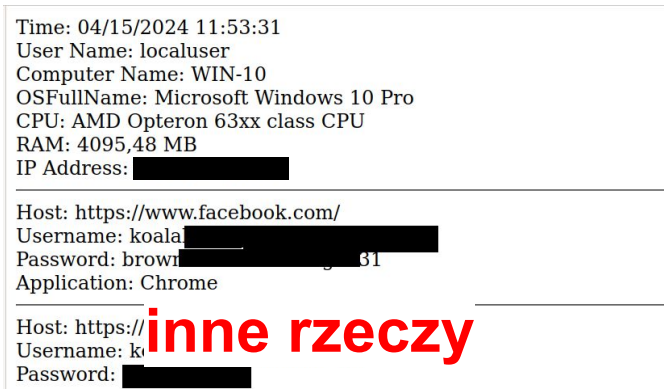
Śledzenie malware w CERT 101



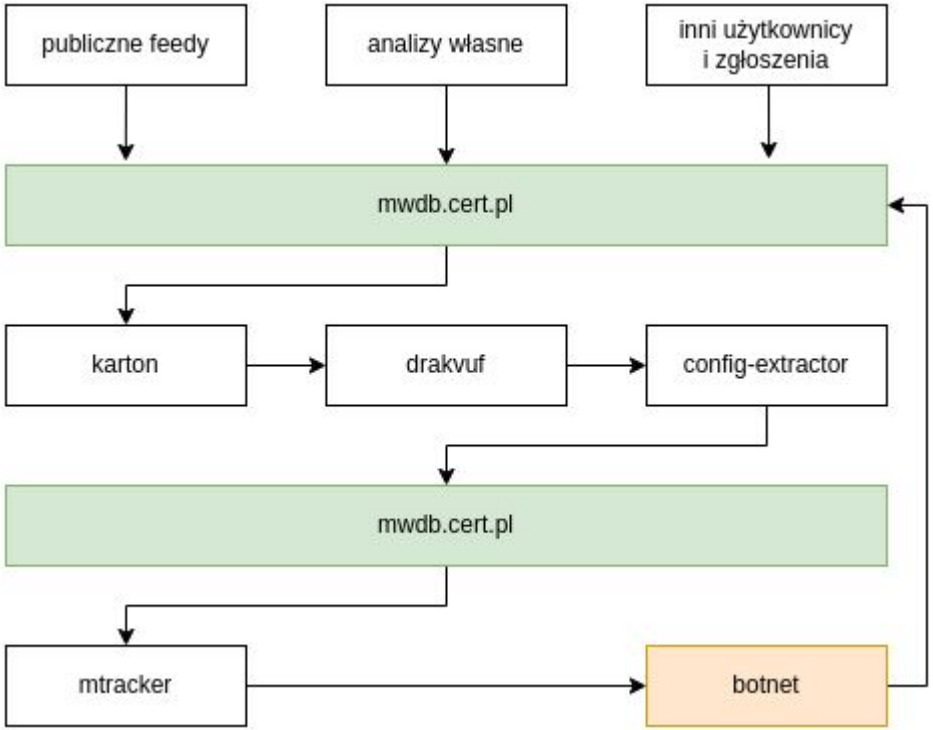
Śledzenie malware w CERT 101







Śledzenie malware w CERT 101



Co można pobrać z serwera C&C?

- Więcej plików wykonywalnych (droppery)

Co można pobrać z serwera C&C?

- Więcej plików wykonywalnych (droppery)
- Nowe komendy i polecenia (trojany)

Co można pobrać z serwera C&C?

- Więcej plików wykonywalnych (droppery)
- Nowe komendy i polecenia (trojany)
- Szablony spamu (spamboty)

Co można pobrać z serwera C&C?

- Więcej plików wykonywalnych (droppery)
- Nowe komendy i polecenia (trojany)
- Szablony spamu (spamboty)
- Wykradzione konta użytkowników (stealery)

Co można pobrać z serwera C&C?

- Więcej plików wykonywalnych (droppery)
- Nowe komendy i polecenia (trojany)
- Szablony spamu (spamboty)
- Wykradzione konta użytkowników (stealery)
- Szablony injectów (trojany bankowe)

Co obecnie pobieramy z serwerów C&C?

-
-
-

Problemy przy pobieraniu danych

- Reverse-engineering ⚠️

```
import requsts

def download_data(c2):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering ⚠️

```
import requsts

def download_data(c2):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- **Serwer proxy ⚠️**

```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- Serwer proxy
- **Maskowanie sie ⚠️**

```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- Serwer proxy
- Maskowanie sie


```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- Serwer proxy
- Maskowanie sie

- **Zrównoleglenie pobierania ⚠️**

```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- Serwer proxy
- Maskowanie sie

- **Zrównoleglenie pobierania ⚠️**
- **Geoblocking po IP ⚠️**

```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- Serwer proxy
- Maskowanie sie

- **Zrównoleglenie pobierania ⚠️**
- **Geoblocking po IP ⚠️**
- **Przechowywanie wyników ⚠️**

```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- Serwer proxy
- Maskowanie sie

- **Zrównoleglenie pobierania ⚠️**
- **Geoblocking po IP ⚠️**
- **Przechowywanie wyników ⚠️**
- **Przeglądanie wyników ⚠️**

```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- Serwer proxy
- Maskowanie sie

- **Zrównoleglenie pobierania ⚠️**
- **Geoblocking po IP ⚠️**
- **Przechowywanie wyników ⚠️**
- **Przeglądanie wyników ⚠️**
- **Długoterminowe utrzymanie ⚠️**

```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

Problemy przy pobieraniu danych

- Reverse-engineering
- Serwer proxy
- Maskowanie sie

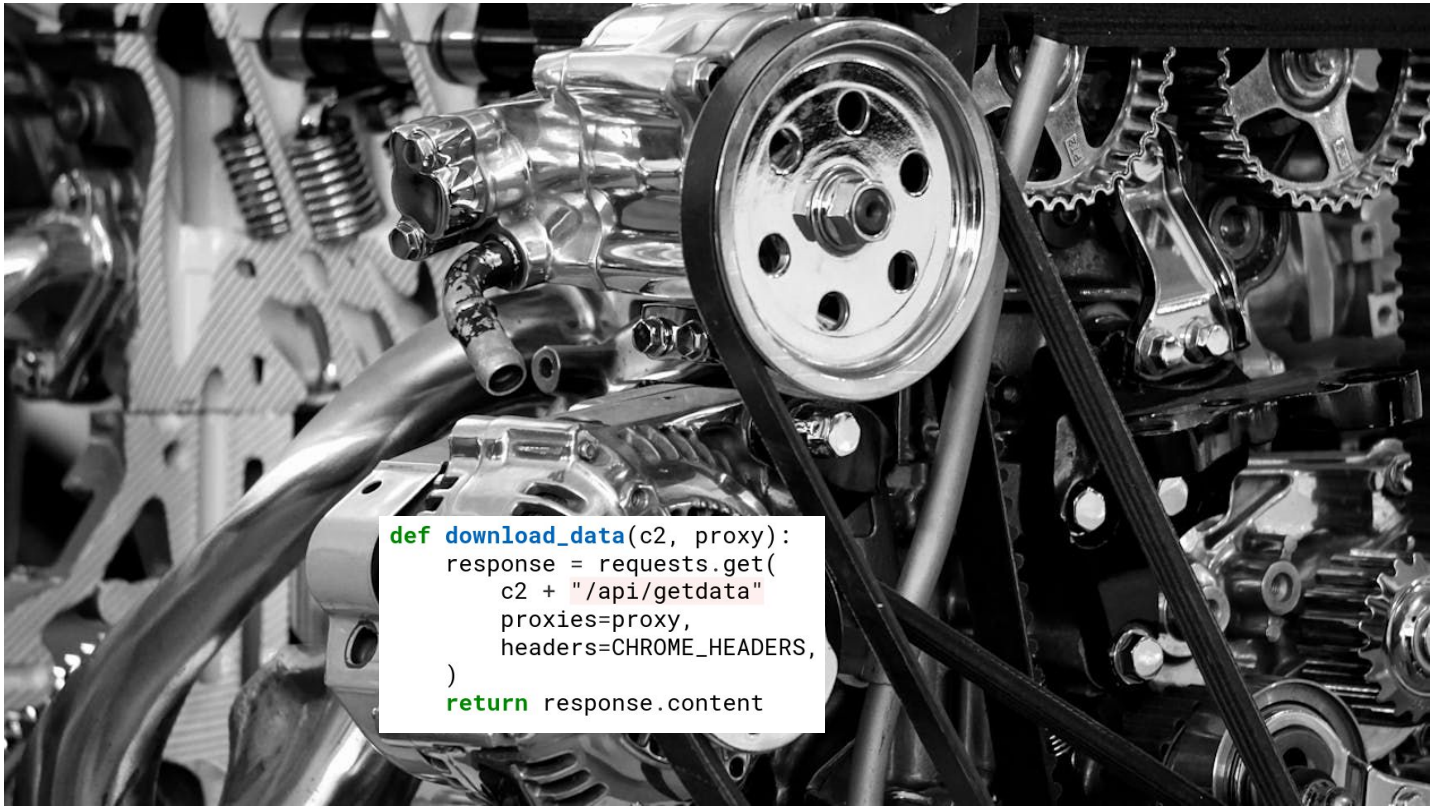
- **Zrównoleglenie pobierania ⚠️**
- **Geoblocking po IP ⚠️**
- **Przechowywanie wyników ⚠️**
- **Przeglądanie wyników ⚠️**
- **Długoterminowe utrzymanie ⚠️**
- **Zarządzanie stanem ⚠️**

```
import requsts

def download_data(c2, proxy):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

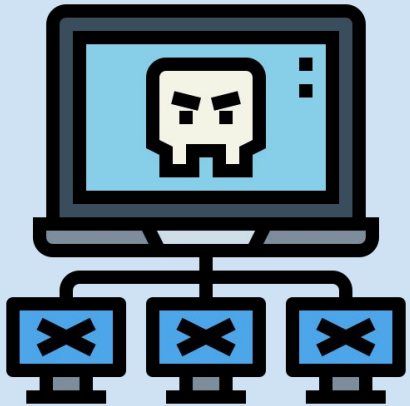
Po co komu mtracker

Po co komu mtracker



```python
def download_data(c2, proxy):
    response = requests.get(
        c2 + "/api/getdata"
        proxies=proxy,
        headers=CHROME_HEADERS,
    )
    return response.content
```

co to jest



mtracker

CERT.PL

Co to jest mtracker

- “System którego zadaniem jest automatyzacja i centralizacja aktywności związanych z **komunikacją ze złośliwym oprogramowaniem**”

Co to jest mtracker

Podczas tej prezentacji omówię rozwijany przez nas system zwany “mtracker” który automatyzuje wiele powtarzalnych kroków w tym procesie. Z dniem konferencji kod systemu

Co to jest mtracker

Podczas tej prezentacji omówię rozwijany przez nas system zwany “mtracker” który automatyzuje wiele powtarzalnych kroków w tym procesie. Z dniem konferencji kod systemu mtracker zostanie również udostępniony na licencji Open-Source na platformie Github. Mamy nadzieje że pozwoli to innym na

https://github.com/CERT-Polska/mtracker/

Co to jest mtracker

- “System którego zadaniem jest automatyzacja i centralizacja aktywności związanych z **komunikacją ze złośliwym oprogramowaniem**”

- Dostępny na licencji **Open-Source**

Co to jest mtracker

- “System którego zadaniem jest automatyzacja i centralizacja aktywności związanych z **komunikacją ze złośliwym oprogramowaniem**”

- Dostępny na licencji **Open-Source**

- Oparty na modułowej architekturze
  - …czyli trzeba napisać własne moduły żeby framework coś robił

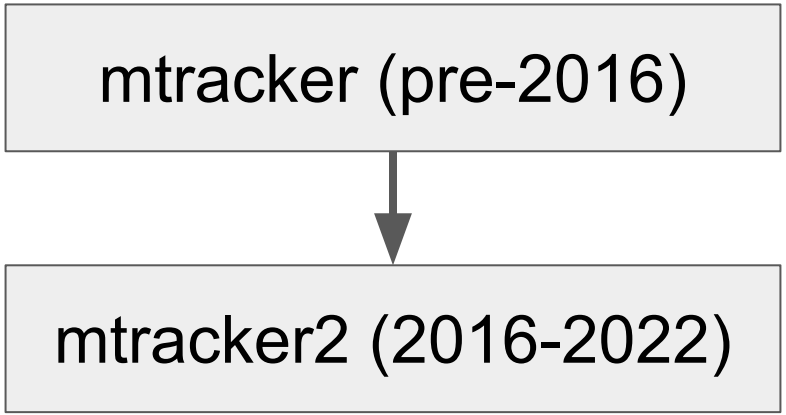
Krótka historia mtrackera

mtracker (pre-2016)

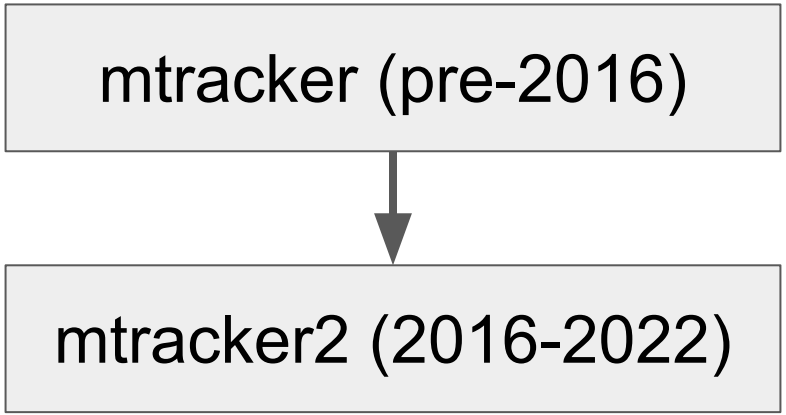
Krótka historia mtrackera

mtracker (pre-2016)

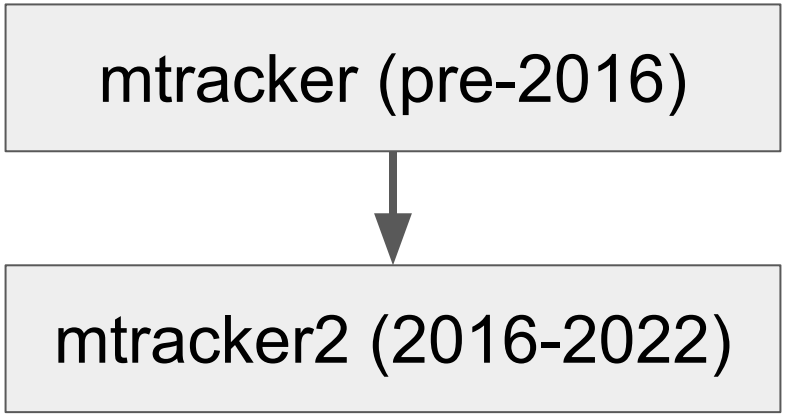
Krótka historia mtrackera



Krótka historia mtrackera

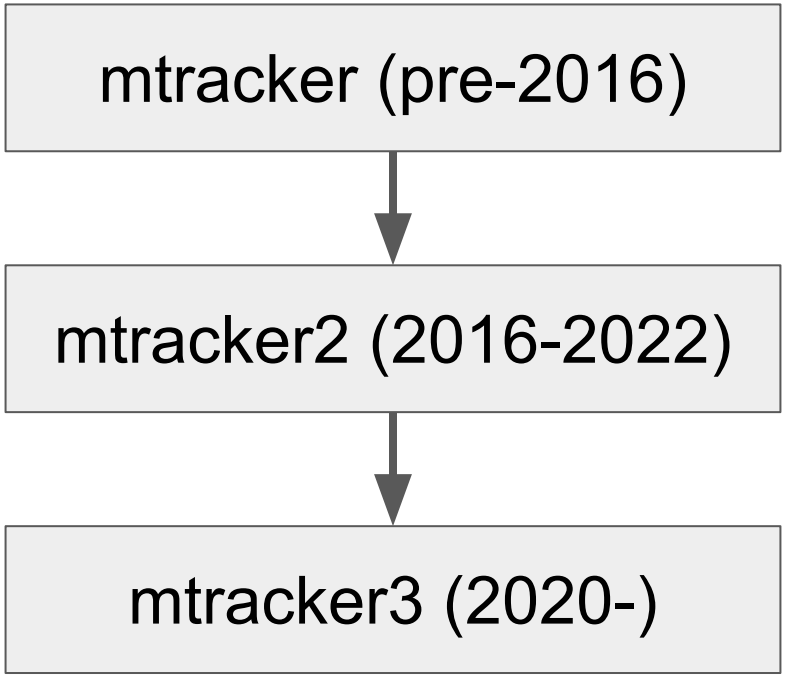


Krótka historia mtrackera

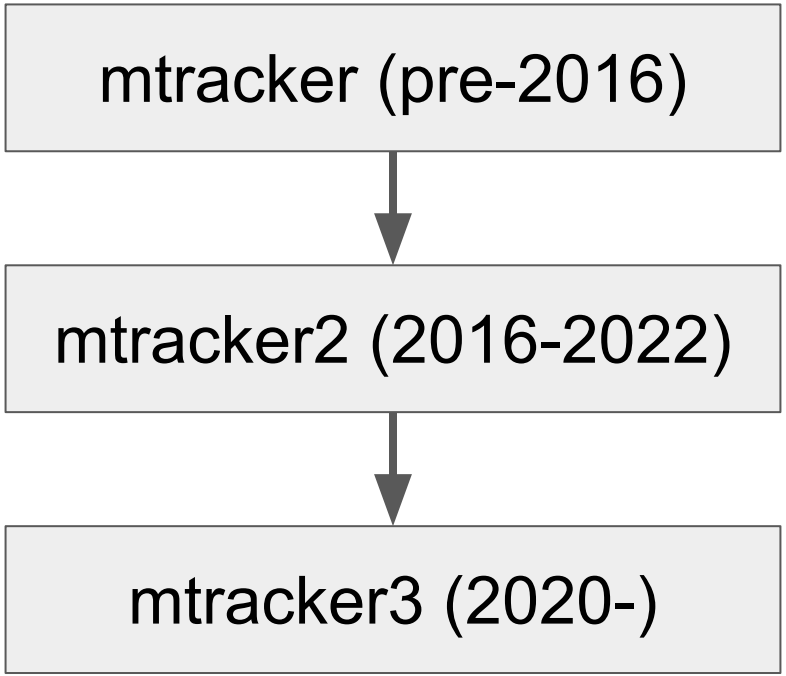


The **second-system effect** or **second-system syndrome** is the tendency of small, elegant, and successful systems to be succeeded by over-engineered, bloated systems, due to inflated expectations and overconfidence.[1]

Krótka historia mtrackera

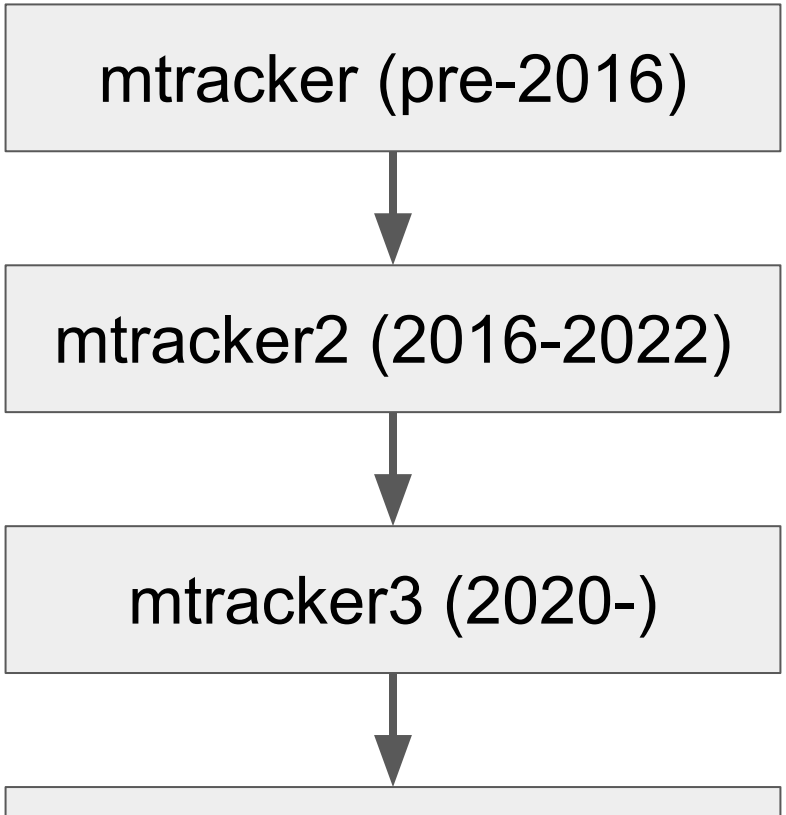


Krótka historia mtrackera



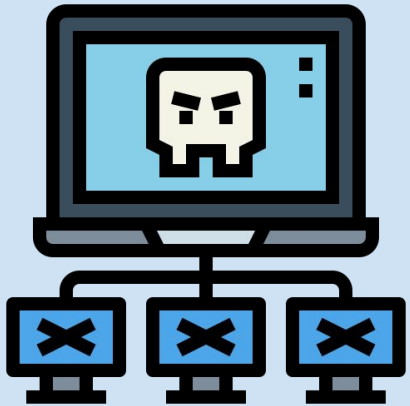
secure

Krótka historia mtrackera



secure

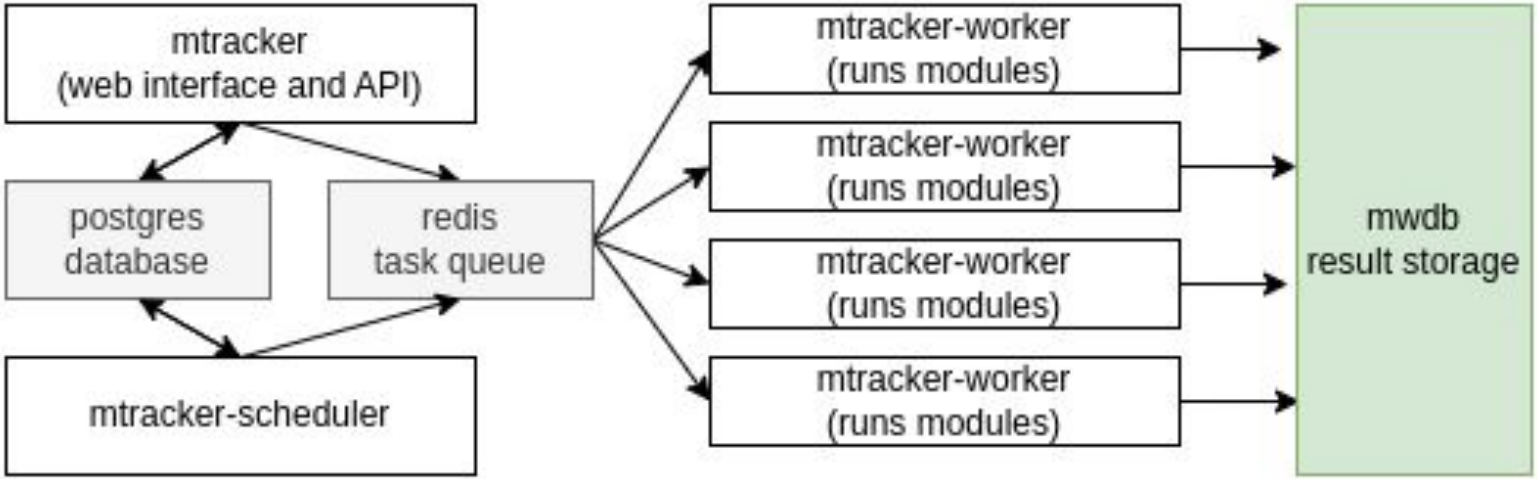
jak działa



mtracker

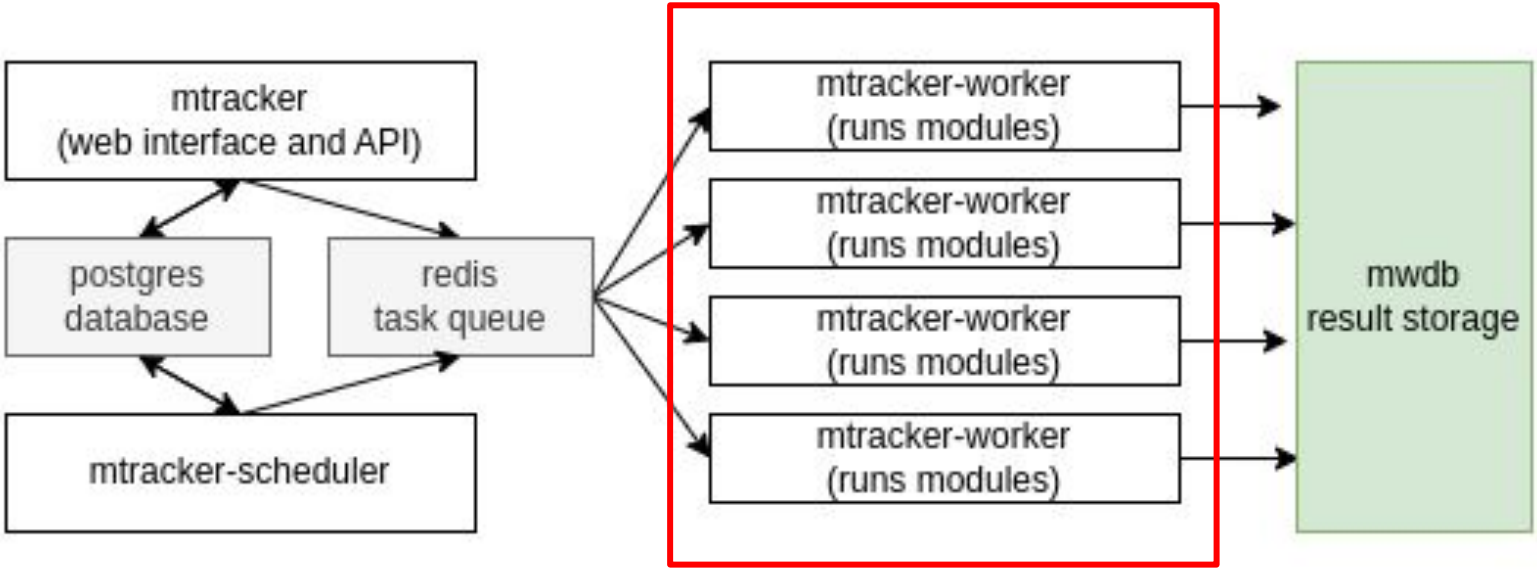
CERT.PL

Jak działa mtracker?



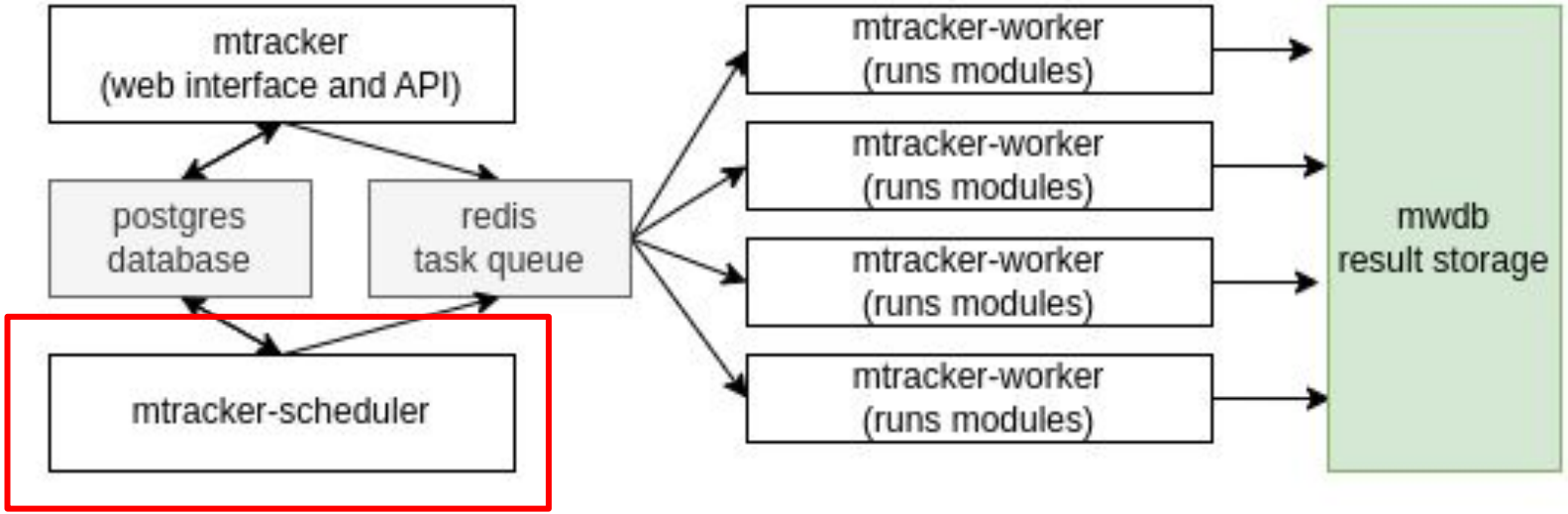
Jak działa mtracker?

Workery: wykonują właściwy kod



Jak działa mtracker?

Scheduler: cyklicznie uruchamia moduły



Jak działa mtracker?

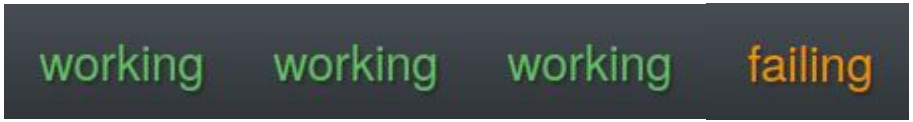
Scheduler: cyklicznie uruchamia moduły

Jak działa mtracker?

Scheduler: cyklicznie uruchamia moduły

Jak działa mtracker?

Scheduler: cyklicznie uruchamia moduły

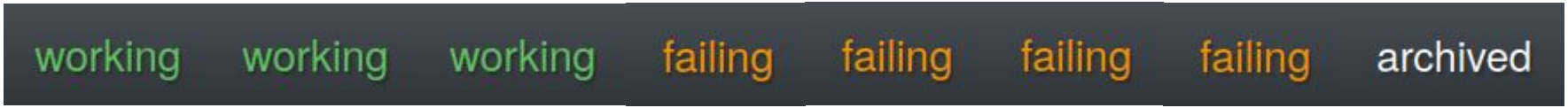


Jak działa mtracker?

Scheduler: cyklicznie uruchamia moduły

Jak działa mtracker?

Scheduler: cyklicznie uruchamia moduły

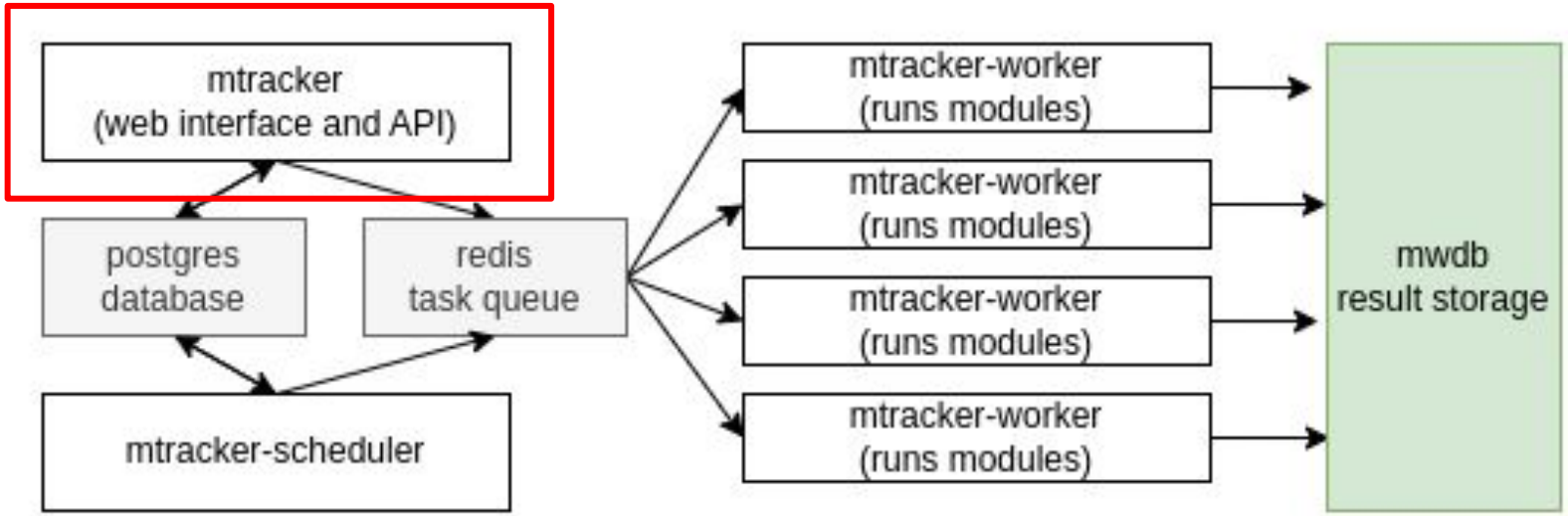


Jak działa mtracker?

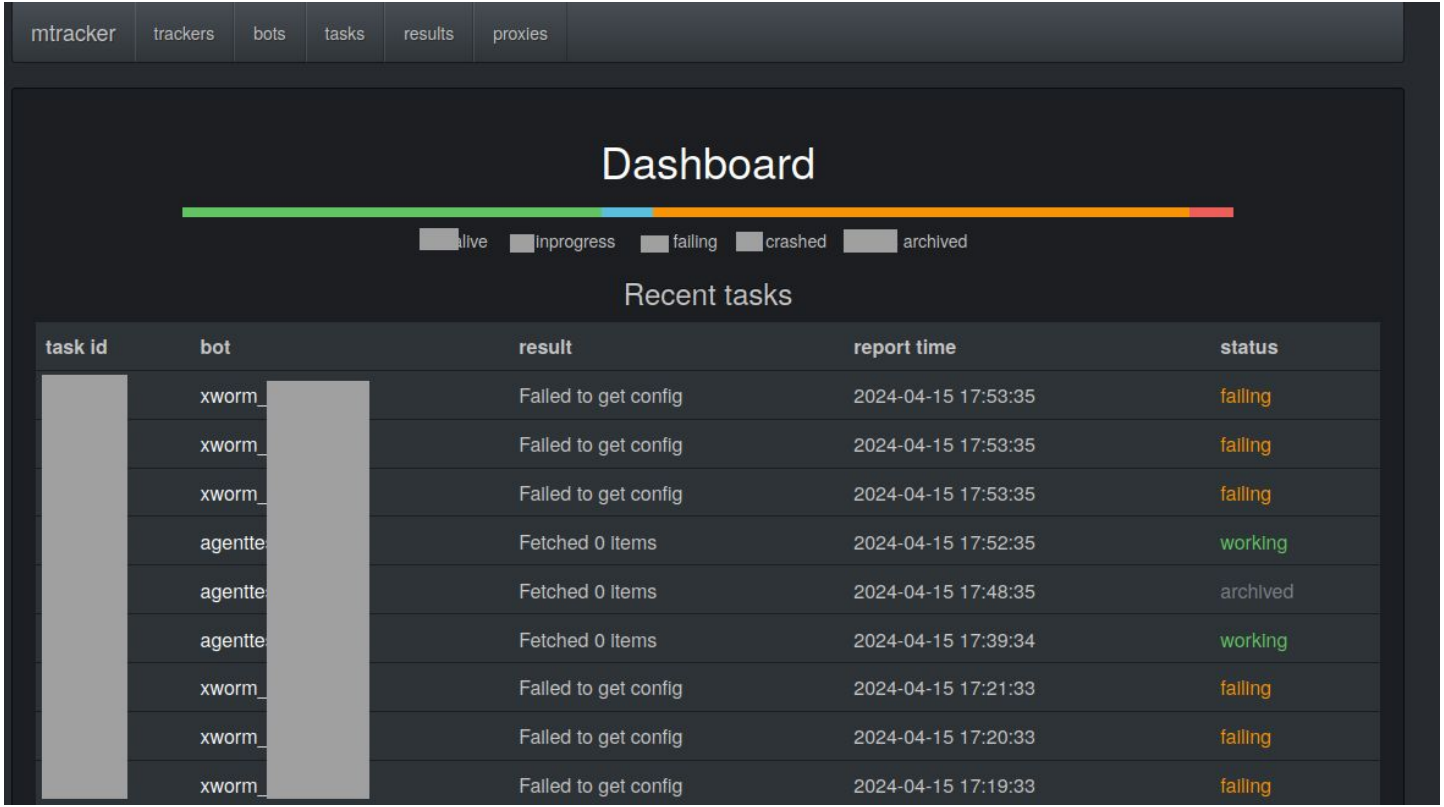
Scheduler: cyklicznie uruchamia moduły

Jak działa mtracker?

Web: API oraz panel admina

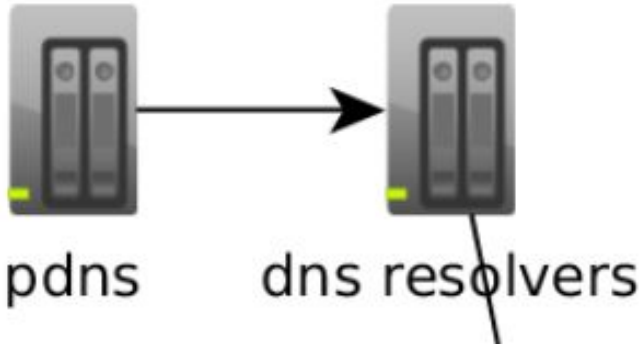


Jak działa mtracker?



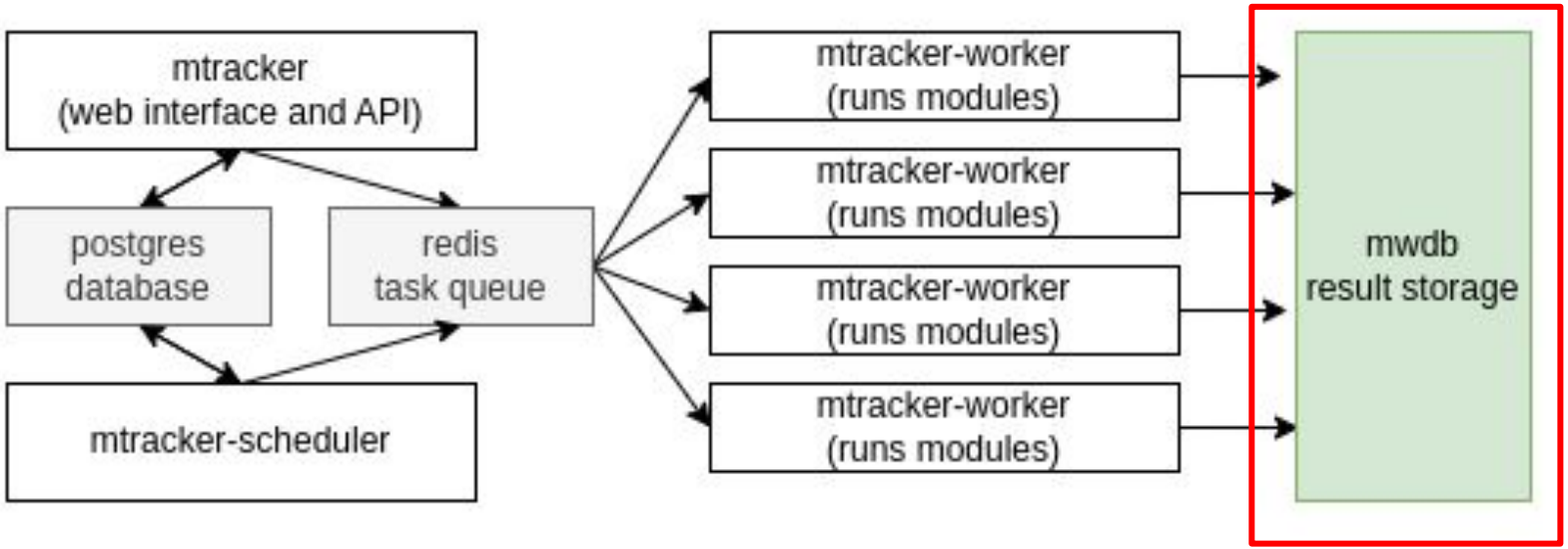
Jak działa mtracker?

Ciekawostka z mtracker2: pdns

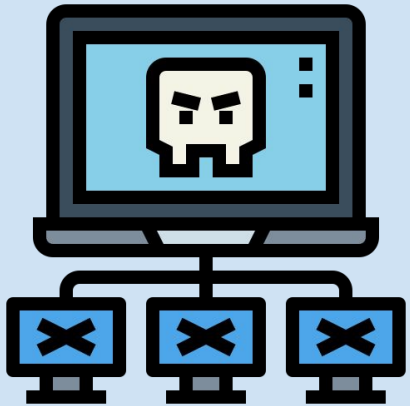


Jak działa mtracker?

Backend: obecnie wspierane tylko mwdb-core



po co tobie



mtracker

CERT.PL

Use case mtrackera

Założenie:

- Znamy protokół malware
- Mamy już konfigurację (adresy)
- Chcemy analizy w większej skali

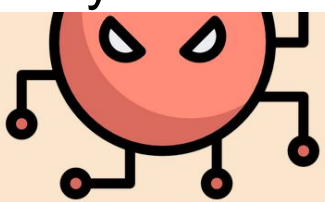
Use case mtrackera

Założenie:

- Znamy protokół malware
- Mamy już konfigurację (adresy)
- Chcemy analizy w większej skali

Rozmówki 🦠-🇵🇱

Więcej na temat RE komunikacji
(Oh My Hack 2023)



czyli o komunikacji ze stealerami

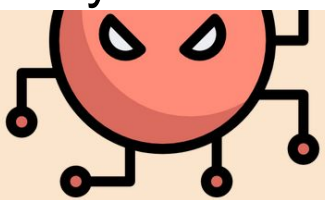
Use case mtrackera

Założenie:

- Znamy protokół malware
  - Jest nietrywialny
- Mamy już konfigurację (adresy)
- Chcemy analizy w większej skali

Rozmówki 🦠-🇵🇱

Więcej na temat RE komunikacji
(Oh My Hack 2023)



czyli o komunikacji ze stealerami

Use case mtrackera

Założenie:

- Znamy protokół malware
  - Jest nietrywialny
- Mamy już konfigurację (adresy)
- Chcemy analizy w większej skali
- Integracja z mwdb mile widziana

Use case mtrackera


```
{
    "drop": [
        {
            "url": "http://5.161.191.146/517c3c5d098ff0da/freebl3.dll"
        },
        {
            "url": "http://5.161.191.146/517c3c5d098ff0da/mozglue.dll"
        },
        {
            "url": "http://5.161.191.146/517c3c5d098ff0da/msvcp140.dll"
        }
    ]
}
```

Jak użyć mtrackera?

```
def download_data(c2):
    print("downloading: " + c2)
    response = requests.get(
        c2 + "/api/getdata"
        proxies=SOCKS_PROXY,
        headers=CHROME_HEADERS,
    )
    save(response.content)
    return True
```

Jak użyć mtrackera?

```
class Stealc(Dropper):
    FAMILY = "stealc"

    def download_data(self, c2):
        print("downloading: " + c2)
        response = requests.get(
            c2 + "/api/getdata"
            proxies=SOCKS_PROXY,
            headers=CHROME_HEADERS,
        )
        save(response.content)
        return True
```

Jak użyć mtrackera?

```
class Stealc(Dropper):
    FAMILY = "stealc"

    def download_data(self, c2):
        self.log.info("downloading: " + c2)
        response = requests.get(
            c2 + "/api/getdata"
            proxies=SOCKS_PROXY,
            headers=CHROME_HEADERS,
        )
        save(response.content)
        return True
```

Jak użyć mtrackera?

```
class Stealc(Dropper):
    FAMILY = "stealc"

    def download_data(self, c2):
        self.log.info("downloading: " + c2)
        data, filename = self.download_drop(
            c2 + "/api/getdata"
        )
        save(data)
        return True
```

Jak użyć mtrackera?

```
class Stealc(Dropper):
    FAMILY = "stealc"

    def download_data(self, c2):
        self.log.info("downloading: " + c2)
        data, filename = self.download_drop(
            c2 + "/api/getdata"
        )
        self.push_binary(data, filename, ["stealc_drop"])
        return True
```

Jak użyć mtrackera?

```
class Stealc(Dropper):
    FAMILY = "stealc"

    def download_data(self, c2):
        self.log.info("downloading: " + c2)
        data, filename = self.download_drop(
            c2 + "/api/getdata"
        )
        self.push_binary(data, filename, ["stealc_drop"])
        return BotResult.WORKING | BotResult.CONTINUE
```

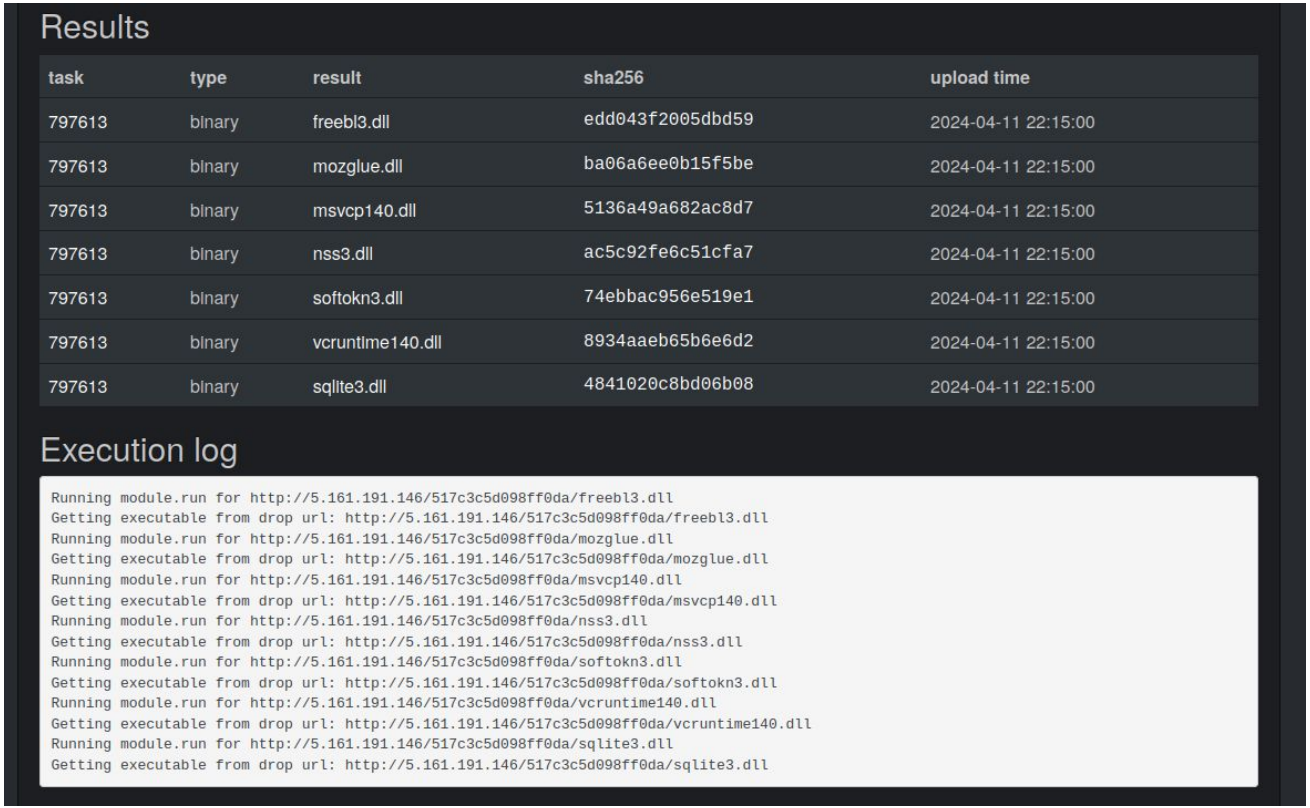
Jak użyć mtrackera?

```
class Stealc(Dropper):
    FAMILY = "stealc"

    @classmethod
    def get_cnc_servers(cls, config, state):
        for url in config.get("drop", []):
            yield url["url"]

    def download_data(self, c2):
        self.log.info("downloading: " + c2)
        data, filename = self.download_drop(
            c2 + "/api/getdata"
        )
        self.push_binary(data, filename, ["stealc_drop"])
        return BotResult.WORKING | BotResult.CONTINUE
```

Jak użyć mtrackera?



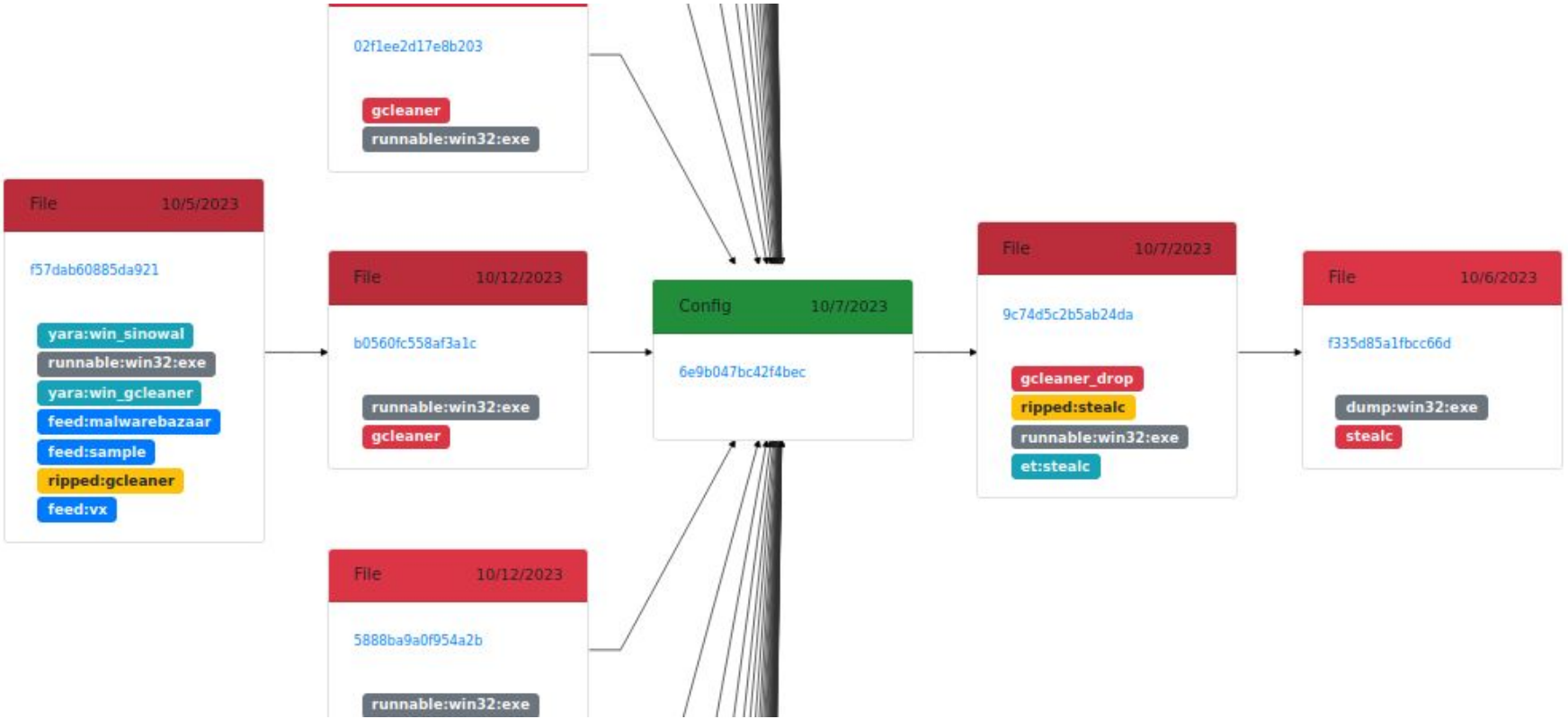
Results

task	type	result	sha256	upload time
797613	binary	freebl3.dll	edd043f2005dbd59	2024-04-11 22:15:00
797613	binary	mozglue.dll	ba06a6ee0b15f5be	2024-04-11 22:15:00
797613	binary	msvcp140.dll	5136a49a682ac8d7	2024-04-11 22:15:00
797613	binary	nss3.dll	ac5c92fe6c51cfa7	2024-04-11 22:15:00
797613	binary	softokn3.dll	74ebbac956e519e1	2024-04-11 22:15:00
797613	binary	vcruntime140.dll	8934aaeb65b6e6d2	2024-04-11 22:15:00
797613	binary	sqlite3.dll	4841020c8bd06b08	2024-04-11 22:15:00

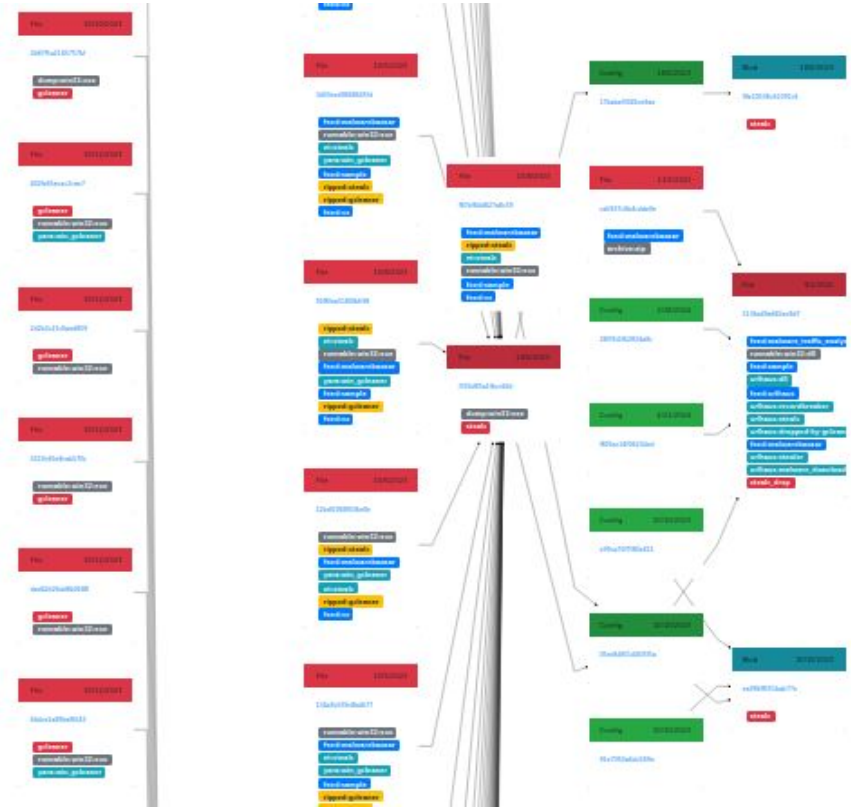
Execution log

```
Running module.run for http://5.161.191.146/517c3c5d098ff0da/freebl3.dll
Getting executable from drop url: http://5.161.191.146/517c3c5d098ff0da/freebl3.dll
Running module.run for http://5.161.191.146/517c3c5d098ff0da/mozglue.dll
Getting executable from drop url: http://5.161.191.146/517c3c5d098ff0da/mozglue.dll
Running module.run for http://5.161.191.146/517c3c5d098ff0da/msvcp140.dll
Getting executable from drop url: http://5.161.191.146/517c3c5d098ff0da/msvcp140.dll
Running module.run for http://5.161.191.146/517c3c5d098ff0da/nss3.dll
Getting executable from drop url: http://5.161.191.146/517c3c5d098ff0da/nss3.dll
Running module.run for http://5.161.191.146/517c3c5d098ff0da/softokn3.dll
Getting executable from drop url: http://5.161.191.146/517c3c5d098ff0da/softokn3.dll
Running module.run for http://5.161.191.146/517c3c5d098ff0da/vcruntime140.dll
Getting executable from drop url: http://5.161.191.146/517c3c5d098ff0da/vcruntime140.dll
Running module.run for http://5.161.191.146/517c3c5d098ff0da/sqlite3.dll
Getting executable from drop url: http://5.161.191.146/517c3c5d098ff0da/sqlite3.dll
```

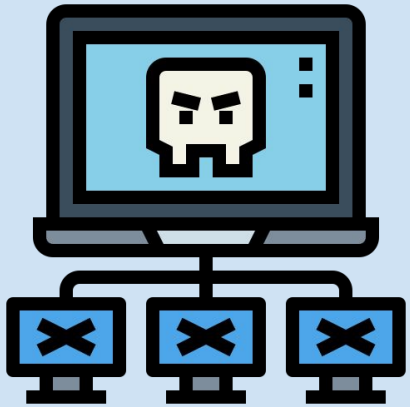
Jak użyć mtrackera?



Jak użyć mtrackera?



podsumowanie prezentacji o



mtracker

CERT.PL

Podsumowanie?

- Można zrobić dwie prezentacje o komunikacji z C&C

Podsumowanie?

- Można zrobić dwie prezentacje o komunikacji z C&C
- Cieszę się że projekt w końcu jest OS 🙂

Podsumowanie?

- Można zrobić dwie prezentacje o komunikacji z C&C
- Cieszę się że projekt w końcu jest OS 🙂
- Jak zwykle, współpraca w rozwoju modułów mile widziana.

Podsumowanie?

- Można zrobić dwie prezentacje o komunikacji z C&C
- Cieszę się że projekt w końcu jest OS 🙂
- Jak zwykle, współpraca w rozwoju modułów mile widziana.

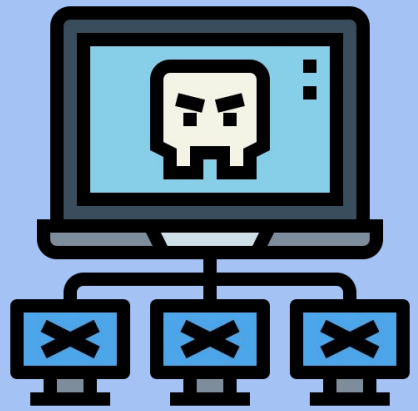
Szczególne **podziękowania** za wkład w rozwój:
@nazywam, @psrok1, @mak, Itay Cohen

Podsumowanie?

- Można zrobić dwie prezentacje o komunikacji z C&C
- Cieszę się że projekt w końcu jest OS 🙂
- Jak zwykle, współpraca w rozwoju modułów mile widziana.

Szczególne **podziękowania** za wkład w rozwój:
@nazywam, @psrok1, @mak, Itay Cohen
i ja (@msm)

to wszystko o

Q&  A?

mtracker

Botnet icon from flaticon.com
by smalllikeart - thanks!

CERT.PL