HASHR

v.2.0.0

CERT-AGID

30/09/2024

Indice

Strumento di ricerca file tramite hash

hashr è uno strumento progettato per cercare file all'interno di un filesystem confrontando il loro hash (ad esempio MD5, SHA1 o SHA256) con una lista di hash noti. Questo è particolarmente utile per indagini di sicurezza informatica, analisi forense e verifica dell'integrità dei file su filesystem di grandi dimensioni.

Funzionalità principali

- **Supporta le seguenti tipologie di hash**: MD5, SHA1 e SHA256.
- **Logging dei file scartati:** registra automaticamente i file che sono stati scartati a causa delle dimensioni o di problemi di accesso.
- **Output personalizzabile:** i risultati possono essere salvati in un file per analisi successive.
- **Per le Pubbliche Amministrazioni:** Le PA accreditate possono utilizzare gli Indicatori di Compromissione (IoC) basati su hash del Feed IoC di CERT-AGID in tempo reale.

Come funziona

hashr legge una lista di hash da un file fornito in input e li confronta con gli hash dei file presenti in una directory specificata o su un'intera unità. Supporta diversi algoritmi di hash e consente di filtrare i file in base alle dimensioni per ottimizzare le prestazioni.

Sintassi base

```
python3 hashr.py --hashlist <file_hash> --rootdir <directory>
```

- **`--hashlist`**: percorso o URL del file di testo contenente la lista di hash da cercare; in alternativa, è possibile specificare un token per prelevare gli hash dal feed IoC fornito dal CERT-AGID[1]. Se viene specificato un URL o un token, gli hash scaricati sono salvati nel file `hashlist.txt`, che può essere riutilizzato per successive scansioni.
- **`--rootdir`**: directory di partenza per la ricerca dei file.

Argomenti opzionali

- **`--hash`**: specifica il tipo di hash da utilizzare (`md5`, `sha1`, `sha256` o `all`). Il valore predefinito è `all`, con cui vengono presi in considerazione tutti e tre gli hash per ciascun file.
- **`--maxsize`**: imposta la dimensione massima dei file da elaborare. I file più grandi verranno saltati (predefinito: 1 GB).
- **`--output`**: salva i risultati del confronto in un file specificato.
- **`--csv, --tsv, --json`**: stampa o salva i risultati nel formato specificato.
- **`--log`**: imposta il file di log per registrare i file saltati (predefinito: `log_{timestamp}.log`).
- **`--ignore-cert`**: disabilita la verifica dei certificati TLS nelle richieste HTTPS.
- **`--live`**: stampa i risultati uno alla volta, invece che tutti insieme alla fine della ricerca.
- **`--version`**: mostra il numero di versione di hashr.

[1] Il token viene rilasciato dal CERT-AGID alle Pubbliche Amministrazioni che hanno fatto richiesta di accreditamento per l'utilizzo del feed IoC. Il token può essere utilizzato solo su macchine il cui indirizzo IP è stato abilitato all'accesso come indicato nel modulo di accreditamento. È possibile scaricare l'elenco di hash da un sistema abilitato all'accesso e successivamente utilizzarlo su altri PC.

Esempi di utilizzo

1. Ricerca di file tramite hash in una directory

Per scansionare file in `/percorso` confrontandoli con gli hash presenti in `hashlist.txt`:

```
python3 hashr.py --hashlist hashlist.txt --rootdir /percorso/
```

Per scansionare file in `/percorso` confrontandoli con gli hash accessibili all'URL `https://public-link-to-ioc/feed.txt`:

```
python3 hashr.py --hashlist https://public-link-to-ioc/feed.txt --rootdir /percorso/
```

Per scansionare file in `/percorso` confrontandoli con gli hash ottenibili dal feed IoC del CERT-AGID utilizzando il token `f81d4fae-7dec-11d0-a765-00a0c91e6bf6`[2]:

```
python3 hashr.py --hashlist f81d4fae-7dec-11d0-a765-00a0c91e6bf6 --rootdir /percorso/
```

2. Ricerca e salvataggio dei risultati in un file

Per salvare i risultati nel file `risultati.txt` usare il parametro `--output`:

```
python3 hashr.py --hashlist hashlist.txt --rootdir /percorso/ --output risultati.txt
```

3. Limitare la dimensione dei file da ricercare a 500MB

Per scartare i file più grandi di 500MB (di default `--maxsize` è 1GB, ossia 1000^3 byte):

```
python3 hashr.py --hashlist hashlist.txt --rootdir /percorso/ --maxsize 500MB
```

4. Personalizzare il file di log

Se si vuole specificare un nome file personalizzato è possibile specificarlo tramite il parametro `--log`:

```
python3 hashr.py --hashlist hashlist.txt --rootdir /percorso/ --log personalizzato.log
```

hashr genera un file di log (predefinito: `log_{timestamp}.log`) che documenta i file saltati durante il processo di scansione. Questo include:

- file più grandi della dimensione massima consentita;
- file non accessibili a causa di permessi o errori.

[2] Il token riportato è puramente a scopo esemplificativo e non è abilitato all'accesso al feed IoC.

Versione per Windows

Oltre alla versione Python di hashr, è disponibile anche una versione eseguibile per sistemi **Windows**. Questa versione consente di eseguire hashr senza dover installare Python o configurare dipendenze aggiuntive rendendo l'uso dello strumento ancora più semplice per gli utenti di Windows.

Esempio di utilizzo

```
hashr.exe --hashlist hashlist.txt --rootdir c:\percorso\
```

Tutte le opzioni della versione Python, come la selezione del tipo di hash, la dimensione massima dei file e il salvataggio dei risultati, sono disponibili anche nella versione Windows.

Note

- Valutare se eseguire il programma come utente amministratore per poter scansionare anche file non accessibili o leggibili da utenti standard.
- Se eseguito su directory contenenti molti file e cartelle, il programma potrebbe impiegare diversi minuti nella fase iniziale di scansione.
- Il programma, ed in particolare la versione eseguibile per Windows, potrebbe essere rilevato come malevolo da antivirus o altri strumenti anti-malware. Si consiglia pertanto di verificare la configurazione degli strumenti di sicurezza installati sulle macchine in cui hashr viene eseguito.

Progettato per le Pubbliche Amministrazioni

Le Pubbliche Amministrazioni accreditate al **Feed IoC del CERT-AGID** hanno accesso privilegiato a un flusso in tempo reale di Indicatori di Compromissione (IoC).

Questi IoC sono fondamentali per rilevare e mitigare le minacce informatiche. hashr consente alle PA accreditate di utilizzare il flusso testuale di IoC per ricercare, an che offline, file associati a hash di IoC collegati a campagne note o APT analizzati dal CERT-AGID.

Grazie a hashr, le PA accreditate possono identificare rapidamente i file compromessi nelle loro infrastrutture, conformandosi alle best practice del CERT-AGID per la protezione dei sistemi pubblici.

Per richiedere ulteriori informazioni, segnalare problemi o inviare suggerimenti, è possibile contattare tramite email il CERT-AGID all'indirizzo info@cert-agid.gov.it.

Link

https://cert-agid.gov.it/

https://cert-agid.gov.it/scarica-il-modulo-accreditamento-feed-ioc/