AmCache & ShimCache - Guide Pratique Cybersecurite

Catégorie : Forensics Lecture : 3 min Publié le : 07/12/2025 Auteur : Ayi NEDJIMI

exécution des. Expert en cybersécurité et intelligence artificielle. Guide technique complet avec recommandations pratiques et outils pour les...

🏢 **Avantages des solutions commerciales**

- **Visualisation avancée** : Timelines interactives, graphiques de relations
- **Corrélation automatique** : Intégration avec bases de données threat intelligence
- **Traitement par lots** : Analyse de centaines/milliers de systèmes
- **Reporting standardisé** : Génération automatique conforme aux standards légaux
- **Support et formation** : Assistance technique et certifications

8.3 Développement de scripts personnalisés

Le développement de scripts personnalisés reste souvent nécessaire pour adresser des besoins spécifiques ou automatiser des workflows complexes. PowerShell, Python, et même des langages comme Go ou Rust sont utilisés pour créer des outils sur mesure optimisés pour des environnements particuliers.

```
import pyregf
import pytsk3
import hashlib
from datetime import datetime
import pandas as pd

class ForensicCorrelator:
    def __init__(self):
        self.amcache_data = []
        self.shimcache_data = []
        self.mft_data = []
        self.prefetch_data = []

    def parse_amcache(self, hive_path):
        """Parse AmCache hive and extract relevant artifacts"""
        regf_file = pyregf.file()
        regf_file.open(hive_path)

        root_key = regf_file.get_root_key()
        file_key = root_key.get_sub_key_by_path("Root\\\\File")

        if file_key:
            for volume_key in file_key.sub_keys:
                for file_ref_key in volume_key.sub_keys:
                    entry = self._extract_amcache_entry(file_ref_key)
                    if entry:
                        self.amcache_data.append(entry)

        regf_file.close()
        return self.amcache_data

    def correlate_artifacts(self):
        """Correlate data from multiple sources"""
        df_amcache = pd.DataFrame(self.amcache_data)
        df_shimcache = pd.DataFrame(self.shimcache_data)

        # Perform correlation based on file paths
        correlated = pd.merge(
            df_amcache,
            df_shimcache,
            on='path',
            how='outer',
            suffixes=('_amcache', '_shimcache')
        )

        # Identify discrepancies
        correlated['time_diff'] = abs(
            (correlated['modified_amcache'] -
correlated['modified_shimcache']).dt.total_seconds()
        )

        anomalies = correlated[correlated['time_diff'] > 3600]

        return correlated, anomalies

    def generate_timeline(self):
        """Generate unified timeline from all sources"""
        timeline = []

        for entry in self.amcache_data:
            timeline.append({
                'timestamp': entry.get('modified'),
```

```
                'source': 'AmCache',
                'action': 'File Modified',
                'path': entry.get('path'),
                'details': f"SHA1: {entry.get('sha1', 'N/A')}"
            })

        timeline.sort(key=lambda x: x['timestamp'] if x['timestamp'] else
datetime.min)

        return timeline

# Utilisation
correlator = ForensicCorrelator()
correlator.parse_amcache("C:\\\\Evidence\\\\Amcache.hve")
correlated_data, anomalies = correlator.correlate_artifacts()
timeline = correlator.generate_timeline()
```

Questions frequentes

Comment mener une investigation forensique sur un systeme compromis ?

Une investigation forensique debute par la preservation des preuves via une image disque et un dump memoire, suivie de l'analyse des artefacts systeme (registres, journaux d'evenements, fichiers prefetch), la reconstruction de la timeline d'activite et la correlation des indicateurs de compromission pour identifier la source et l'etendue de l'attaque.

Quels sont les outils essentiels pour l'analyse forensique ?

Les outils essentiels pour l'analyse forensique incluent Volatility pour l'analyse memoire, Autopsy et FTK pour l'analyse disque, KAPE et Velociraptor pour la collecte automatisee, Plaso pour la creation de timelines, ainsi que des outils de triage comme Eric Zimmerman's tools pour l'analyse des artefacts Windows.

Pourquoi la chaine de custody est-elle importante en forensique ?

La chaine de custody garantit l'integrite et l'admissibilite des preuves numeriques en documentant chaque etape de manipulation, de la collecte a la presentation. Sans une chaine de custody rigoureuse, les preuves peuvent etre contestees juridiquement et perdre leur valeur probante.

Pour approfondir, consultez les ressources officielles : SANS White Papers, NVD - NIST et ANSSI.

Sources et références : SANS SIFT · MITRE ATT&CK

Articles connexes

- Windows Server 2025 - Guide Pratique Cybersecurite
- MacOS Forensics : Artifacts et Persistence : Guide Complet
- LNK & Jump Lists : Strategies de Detection et de Remediation

- Malware Reverse : Analyse de Cobalt Strike 5 : Guide Complet

Conclusion : Maîtrise et perspectives futures

L'analyse forensique d'AmCache et ShimCache représente bien plus qu'une simple extraction de données ; elle constitue un art nécessitant une compréhension profonde des mécanismes Windows, une maîtrise des outils d'analyse, et une capacité à synthétiser des informations provenant de sources multiples et parfois contradictoires. La complexité croissante des systèmes Windows modernes, avec leurs multiples couches d'abstraction et leurs mécanismes de compatibilité élaborés, rend cette expertise de plus en plus critique.

Les évolutions futures de Windows, notamment avec l'intégration croissante de la télémétrie cloud et des mécanismes de sécurité basés sur l'IA, promettent d'enrichir encore davantage ces sources d'artefacts. Windows 11 a déjà introduit de nouveaux champs dans AmCache et modifié certains comportements de ShimCache. Les analystes forensiques doivent maintenir une veille technologique constante et adapter continuellement leurs méthodologies et outils.

L'automatisation et l'intelligence artificielle promettent de changer l'analyse de ces artefacts dans les années à venir. Les modèles de machine learning entraînés sur des patterns d'exécution normaux pourront identifier automatiquement les anomalies subtiles que même un analyste expérimenté pourrait manquer. Cependant, cette automatisation ne remplacera pas le jugement expert et la compréhension contextuelle qu'apporte l'analyste humain.

🚀 **Perspectives d'évolution**

- **IA et ML** : Détection automatique de patterns d'attaque et d'anomalies comportementales
- **Télémétrie cloud** : Enrichissement des données AmCache via Microsoft Defender for Endpoint
- **Nouveaux formats** : Adaptation continue aux changements de structure Windows
- **Corrélation EDR** : Intégration avec solutions EDR/XDR pour contexte en temps réel
- **Standardisation** : Émergence de frameworks d'analyse unifiés (DFIR-ORC, Velociraptor)

En définitive, AmCache et ShimCache continuent d'évoluer en tant que sources forensiques critiques, et leur importance ne fera que croître avec la complexification des menaces et l'sophistication des techniques d'attaque. Les professionnels de la sécurité et les analystes forensiques qui investissent dans la maîtrise approfondie de ces artefacts se positionnent avantageusement pour relever les défis de cybersécurité de demain.

Ressources open source associées :

- awesome-cybersecurity-tools — Liste de 100+ outils de cybersécurité

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr