VolatileEve

Maximilian Li (2941769), Victor Schümmer (2930796)

# **1. Einleitung**

Im Rahmen des P4-Praktikums ‚Implementierung in Forensik und Mediensicherheit‘ sollte ein Tool zur automatischen Generierung von Arbeitsspeicher-Abbildern (memory dumps) zu Trainingszwecken implementiert werden. In den dumps sollen ähnlich wie beim Capture the Flag[[1]](#footnote-2) (CTF) randomisierte Geheimnisse (secrets) versteckt sein, die es zu finden gilt.

Ein derartiges secret könnte zum Beispiel folgendermaßen versteckt sein:

In einem Texteditor ist ein eine Datei geöffnet, die scheinbar willkürliche Zeichenfolgen enthält. In einem Browser ist eine Google-Anfrage zum Thema Verschlüsselung geöffnet. Durch Entschlüsselung der Zeichenfolge mit einem Passwort, welches in einer weiteren Textdatei geöffnet ist, ergibt sich das Geheimnis.

Die secrets sollen dabei randomisiert sein, damit Teilnehmer die Lösung nicht von anderen erfahren können und es muss außerdem eine (grobe) Lösung generiert werden, die auch den konkreten Wert des secrets enthält. Die dumps sollen auf einem Windows(-7)-System generiert werden und kompatibel mit dem Speicherforensik-Tool volatility sein, während das zu implementierende Tool auf Linux laufen soll.

# **2. Konzept**

Da Anwender in der Regel ungern einen dump ihres eigenen Systems verbreiten wollen, und damit mehr Kontrolle über den Speicherinhalt gewährleistet ist, bringt VolatileEve seine eigene Windows-7-Virtual-Machine mit. Diese wird von VolatileEve in einen Zustand mit geöffneten Programmen, Registry-Einträgen etc. gebracht und anschließend ein Speicherabbild erzeugt.

Um vielfältige und flexible Szenarien zu ermöglichen, haben wir uns für einen modularen Aufbau entschieden. Auf der Hostmaschine wird dafür aus Anweisungen ein Skript generiert, welches anschließend auf die VM verschoben und dort ausgeführt wird. Sobald das Skript fertig ausgeführt wurde meldet die VM dem Hostsystem, dass sie fertig ist, woraufhin das Hostsystem den Zustand der VM einfriert und ein Speicherabbild erstellt.

(Schematische Ablauf-Darstellung aus Powerpoint)

Um einen leichteren Einstieg in die Software zu ermöglichen, sind einige vorgefertigte Szenarien verfügbar:

(Liste der vorgefertigten Szenarien)

# **3. Implementierung**

## 3.1 VolatileEve.sh

Der Einstiegspunkt ist das Bash-Skript volatileEve.sh. Dieses setzt zunächst die VM mit virsh auf einen „sauberen“ Snapshot zurück, um einen problemlosen Durchlauf zu gewährleisten und zu verhindern, dass sich Restdaten im Speicher ansammeln. Anschließend wird ein Python-Skript aufgerufen, welches für die Erstellung der nötigen Dateien für das entsprechende Szenario verantwortlich ist. Die generierten Dateien werden dann über das SCP-Protokoll auf die VM geschoben und die VM neugestartet.

Nach dem Neustart versetzt sich die VM in den Zustand der gedumpt werden soll und signalisiert durch das Anlegen einer Datei, dass sie fertig ist. Das Bash-Skript prüft via SSH alle 2 Sekunden, ob die VM fertig ist, friert dann ihren Zustand ein und generiert mit virsh einen Speicherdump der VM.

## 3.2 Das python-Framework

Für die Generierung der benötigten Dateien existieren Wrapper-Klassen für verschiedene Windowsanwendungen, die eine aufs wesentliche beschränkte Konfigurationsmöglichkeit der zu startenden Anwendungen bieten. Aus dem so erstellten Modell wird anschließend automatisch ein AutoIt-Skript (\*.au3-File) generiert. Autoit ist eine Software, die Automatisierung von Abläufen auf Windows-Systemen ermöglicht und auf der mitgelieferten VM bereits vorinstalliert ist.

Zusätzlich zu dem AutoIt-Skript werden Bilder und weitere benötigte Dateien generiert und im Verzeichnis output abgelegt, welches dann auf die VM geschoben wird. In einem weiteren Verzeichnis liegen zudem Dateien, die nicht für die VM bestimmt sind, für manche Szenarien zum Beispiel ein verschlüsseltes Zip-Archiv, dessen Schlüssel im Dump versteckt ist, sowie eine Textdatei mit automatisch generierten Lösungshinweisen.

### 3.3 Die VM

Die Windows-7-VM liegt als kvm/qemu-Image vor. Auf der VM sind benötigte Programme wie Firefox, 7Zip und Notepad++ für die Szenarien, sowie Cygwin mit sshd und AutoIt vorinstalliert. Das automatische Update wurde deaktiviert, um zu verhindern, dass der Update-Prozess mit den Szenarien interferiert.

Nach dem Systemstart wird per Group-Policy automatisch eine Batchdatei gestartet, welche AutoIt mit der au3-Datei für das Szenario ausführt.

# **4. Installation**

Zur Verwendung von VolatileEve muss Python in Version 3.5 oder neuer, sowie kvm/qemu auf dem System installiert sein. Damit Flags auch in Passwortgeschützten Zip-Archiven gespeichert werden können sollte zusätzlich zip oder 7z installiert sein.

# **5. Verwendung**

Zur Verwendung sollte zunächst die lokale IP-Adresse der VM in Erfahrung gebracht werden.

VolatileEve kann folgendermaßen über ein Terminal ausgeführt werden:

volatileEve.sh <number of dumps> <local ip of vm> <path to challenge>

Ein Beispielaufruf aus dem Verzeichnis volatileEve könnte also so aussehen:

volatileEve.sh 10 192.168.0.1 ./challenges/autologin.py

1. https://de.wikipedia.org/wiki/Capture\_the\_Flag#Computersicherheit [↑](#footnote-ref-2)