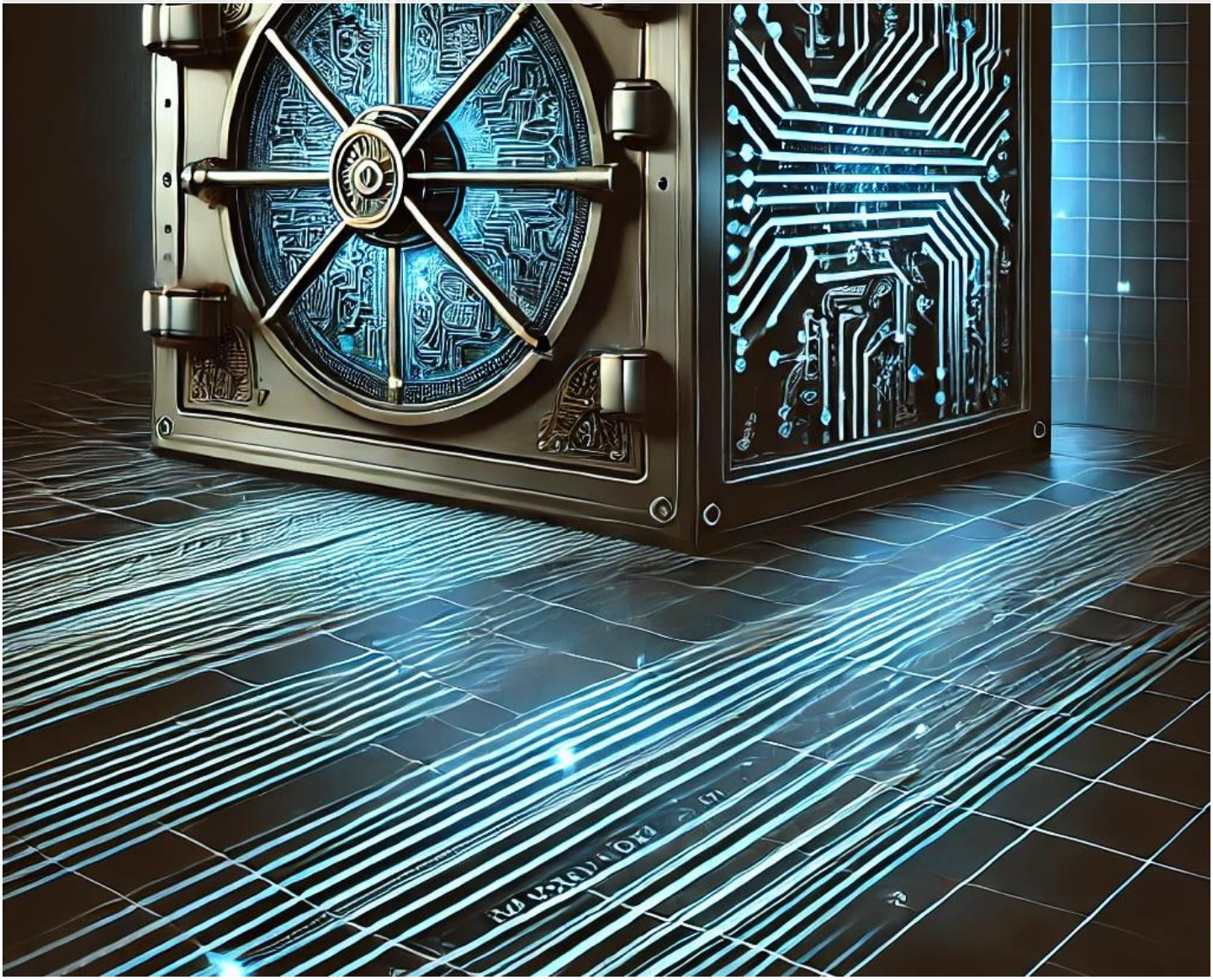
VAULT 7 – GRASSHOPPER

Analyse von CIA Leaks

21 November 2024

HSLU – Informatik
Alenka Isenring, Andrea Megan Sustic



I. Abstract

In dieser Arbeit wird das Grasshopper Leak untersucht, das als Teil des Vault 7 CIA Leaks auf der Whistleblower-Website WikiLeaks veröffentlicht wurde. Grasshopper ist ein Tool, das es ermöglicht, benutzerspezifische Malware für Windows-Systeme zu erstellen, indem verschiedene Komponenten, wie Installer, Persistenzmodule, Payloads, Builder und Post-Prozessoren kombiniert werden. Der Fokus der Analyse liegt dabei auf der Architektur und Funktionsweise des Tools.

Die Methodik der Arbeit basiert auf einer Kombination aus Literaturrecherche, Analyse offizieller Dokumentationen und zusätzlichen Quellen. Ein hypothetischer Use Case demonstriert, wie Grasshopper in der Praxis genutzt werden könnte. Hierbei wird der gesamte Ablauf von der Konfiguration der Module bis zur Ausführung der Malware schrittweise beschrieben.

Anschliessend analysiert die Arbeit die Konsequenzen des Leaks und zeigt die Reaktion der CIA sowie der betroffenen Unternehmen auf. Dabei werden potenzielle Risiken für die Cybersicherheit sowie die nachgebauten Tools aufgeführt. Obwohl mit dem Leak nur Benutzerhandbücher und keinen Quellcode veröffentlicht wurde, zeigt die Analyse, dass die geleakten Informationen ausreichen könnten, um neue Tools zu erstellen. Dadurch besteht das Risiko, dass solche Werkzeuge für potenzielle Angriffe eingesetzt werden.

II. Inhaltsverzeichnis

1 Fragestellung

Das Ziel dieser Arbeit ist es den Vault 7 Leak mit dem Schwerpunkt auf das Tool «Grasshopper» aufzuarbeiten und zu analysieren. Zur Konkretisierung dieses Ziels wurden drei Teilziele definiert:

1. **Überblick des Vault 7 Leaks**: Ein Überblick über den Inhalt des Vault 7 Leaks, mit besonderem Fokus auf Grasshopper, ist erarbeitet worden.
2. **Verstehen von Grasshopper**: Ein Verständnis der Architektur, Funktionalität und potenziellen Einsatzmöglichkeiten von Grasshopper wurde entwickelt.
3. **Bewusstsein über die Konsequenzen des Grasshopper-Leaks**: Die generellen sowie die sicherheitstechnischen Auswirkungen des Grasshopper-Leaks wurden erfasst.

Im ersten Teilziel geht es um die Untersuchung des Vault 7 Leaks mit einem speziellen Fokus auf die Rolle von Grasshopper. Das Ziel ist es, zu verstehen, welche Informationen durch das Leak offengelegt wurden.

Das zweite Teilziel bezieht sich auf die Auseinandersetzung mit dem Tool, um ein umfassendes Verständnis über das Grasshopper-Tool selbst zu erlangen. Dieses Verständnis bildet die Grundlage für die Analyse der sicherheitstechnischen Auswirkungen von Grasshopper.

Das dritte Teilziel befasst sich mit den Folgen des Grasshopper-Leaks. Hierbei liegt eine Analyse darüber vor, ob und wie Grasshopper in realen Cyberangriffen genutzt wurde und welche Konsequenzen das Vault 7 Leak mit sich zieht.

2 Stand der Technik

Im Rahmen der Literaturrecherche konnten keine veröffentlichten wissenschaftliche Studien oder Analysen über das Vault 7 Leak im Allgemeinen oder das Grasshopper-Leak im Speziellen gefunden werden.

Aufgrund der mangelnden Informationslage bieten die auf Wikileaks geleakten Originaldokumente die beste Grundlage für die Analyse der Arbeit und die Erreichung der Zielsetzung.

Es sind zusätzlich verschiedene Zeitungsartikel gefunden worden, die sich auf die Originaldokumente beziehen. Aufgrund der begrenzten verfügbaren Informationen wurden auch Beiträge aus Zeitungen, Blogs und Pressemitteilungen von Unternehmen als Sekundärquellen für die Arbeit herangezogen. Obwohl diese Quellen nicht den strengen wissenschaftlichen Kriterien entsprechen, ermöglichen sie eine differenzierte Analyse der Folgen des Grasshopper-Leaks.

2.1 Vault 7 – CIA Leak

Das Vault 7 Leak besteht aus einer Sammlung von Dokumenten der US-amerikanischen Central Intelligence Agency (CIA), die zwischen März und September 2017 auf der Enthüllungsplattform Wikileaks veröffentlicht wurde (WikiLeaks, 2017c). Diese Dokumente enthüllen Details über eine Vielzahl von Cyber-Tools, die von der CIA entwickelt wurden, und Schwachstellen in modernen Betriebssystemen und Netzwerkkomponenten ausnutzen, um Geräte wie Smartphones, Router und Computer zu infizieren (WikiLeaks, 2017a).

WikiLeaks ist eine Whistleblower-Plattform, die seit 2006 geheime Dokumente zu Themen wie Krieg, Spionage und Korruption veröffentlicht (WikiLeaks, o. J.). Die Vault 7 Dokumente sollen ursprünglich aus dem Center for Cyber Intelligence (CCI) der CIA in Langley, Virginia, stammen und waren zuvor unter ehemaligen Hackern und Regierungsauftragsnehmern im Umlauf, bevor sie von einer dieser Quellen an WikiLeaks weitergegeben wurden (Kovacs, 2017) (WikiLeaks, 2017a).

Das Vault 7 Leak besteht insgesamt aus 23 verschiedenen CIA Projekten:

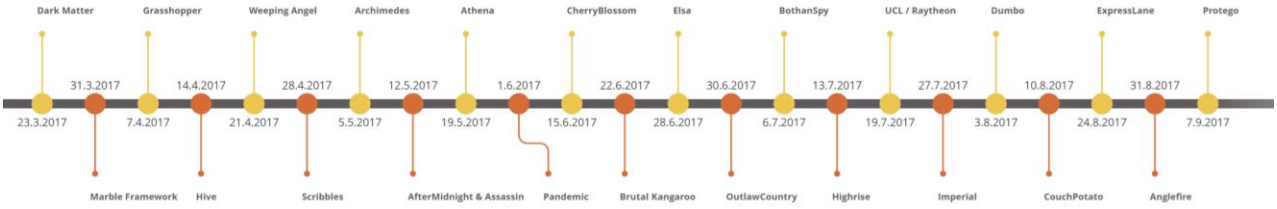


Abbildung 1 - Timeline der geleakten Vault 7-CIA Projekten auf Wikileaks (WikiLeaks, 2017c)

Im Anhang unter Kapitel 8.1 befindet sich eine Kurzbeschreibung zu jedem der 23 CIA Projekte, die mit dem Vault 7 Leak veröffentlicht wurden.

2.2 Grasshopper

In dieser Arbeit wird das Grasshopper-Tool analysiert, das eines der geleakten Projekte des Vault 7 Leaks ist. Insgesamt wurden 27 Dokumente im Rahmen des Vault 7 Grasshopper-Leaks veröffentlicht, wobei es sich um Benutzerhandbücher handelt, die ausführliche Informationen über die Funktionsweise des Tools für CIA-Agenten enthalten (Kovacs, 2017b). Eine genaue Liste der veröffentlichten Dokumente befindet sich im Anhang unter Kapitel 8.2.

2.2.1 Funktionsweise von Grasshopper

Grasshopper ist ein modulbasiertes Software-Tool, das speziell für Windows-Betriebssysteme konzipiert wurde und als Command Line Interface (CLI)-basiertes Tool fungiert. Es ermöglicht die Erstellung und Anpassung benutzerdefinierter Malware-Installationsprogramme, die in Microsoft-Umgebungen eindringen und Sicherheitsmechanismen, wie Antivirenprogramme, umgehen können (News, o. J.).

Das Tool wurde so entwickelt, dass es von herkömmlichen Antivirenlösungen nicht erkannt wird, was es zu einem effektiven Mittel für Cyber-Spionage macht (News, o. J.). Agenten können mit dem Grasshopper-Builder spezifische Komponenten auswählen, um massgeschneiderte Malware-Payloads zu erstellen, die an die technischen Details des Zielsystems, wie Betriebssystem und Antivirenprogramm, angepasst sind (Cimpanu, 2017).

2.2.2 Komponenten von Grasshopper

Das Grasshopper-Tool besteht aus fünf Komponenten: Installers, Persistenzmodule, Payloads, Builders, und Post Processors (CIA, 2013).

In dieser Arbeit werden die Komponenten basierend auf ihrer spezifischen Rolle in drei Hauptkategorien eingeordnet: Installer, einsetzbare (deployable) Module und funktionale Komponenten.

1. **Installer** sind einsatzfähige Komponenten, die als Vermittler für Grasshopper-Module fungieren. Sie sind dazu gedacht, andere Module auf dem Zielsystem zu laden, zu installieren oder auszuführen und löschen sich nach der Bereitstellung selbst (CIA, 2013).

2. **Deployable Modules** sind ausführbare Komponenten von Grasshopper, die direkt auf dem Zielsystem arbeiten. Diese Module kümmern sich um die Persistenz, die Bereitstellung von Payloads und die Aufgabenplanung und gewährleisten die Funktionalität und Langlebigkeit der Malware auf kompromittierten Geräten. Zu den Deployable-Modulen gehören:

    a. **Persistenzmodule:** Sie sind dafür verantwortlich, Payloads auf einem Zielsystem zu installieren. Jedes Modul repräsentiert einen anderen Peristenz oder Soft-Persistenzmechanismus (CIA, 2013).

    b. **Payloads:** Programme, die mithilfe von Grasshopper auf einem Zielsystem persistent gemacht werden können. Sie werden durch Katalogeinträge spezifiziert, die den Builder-Tools Informationen über die Payload liefern. Neue Payloads können zu Grasshopper hinzugefügt werden, indem ein Eintrag in den Katalog geschrieben wird. Der Benutzer muss zur Erstellungszeit eine konfigurierte Payload-Binärdatei bereitstellen (CIA, 2013).

3. **Funktionale Komponenten** sind das Fundament des modularen Designs von Grasshopper und ermöglichen die Anpassung und Analyse nach dem Einsatz. Im Gegensatz zu einsatzfähigen Modulen werden diese nicht an das Einsatzziel gesendet, sind aber für den Aufbau und die Überwachung der Fähigkeiten des Werkzeugs verantwortlich.

    a. **Der Grasshopper Builder** ist das zentrale Konfigurationswerkzeug von Grasshopper, mit dem Anwender Module, Persistenzmethoden und Payloads auswählen und konfigurieren können. Er dient als Mechanismus, der benutzerdefinierte Malware-Pakete auf der Grundlage der betrieblichen Anforderungen zusammenstellt. Der Builder ist für die Generierung von ausführbaren Dateien verantwortlich, die die ausgewählten Komponenten enthalten, so dass sich das Tool an verschiedene Zielumgebungen anpassen und eine Entdeckung verringern kann (CIA, 2013).

    b. **Der Postprozessor** dekodiert kodierte Binärprotokolle in lesbare XML-Berichte, die die auf dem Zielsystem durchgeführten Aktionen detailliert beschreiben und Metadaten wie Systemdetails und Installationsergebnisse enthalten. Zusätzlich entschlüsselt der Postprozessor die vom Payload gesammelten Daten mit Hilfe des vom Installer generierten Build-Skripts (CIA, 2013).

3 Ideen und Konzepte

In diesem Kapitel werden die Ideen und Konzepte dargestellt, die zur Erreichung der formulierten Ziele dieser Arbeit beitragen sollen. Aufgrund der grossen Menge an geleakten CIA-Tools mit dem Vault 7, wurde entschieden, dass sich diese Arbeit auf das Grasshopper-Tool fokussiert.

Fokus auf die Modularität von Grasshopper

Um Grasshopper zu verstehen, ist es wichtig die modulare Architektur des Tools und die verschiedenen Module zu analysieren. Dabei wurde beschlossen die verschiedenen Module in die Modultypen zu unterteilen und die Funktionsweise jedes Moduls kurz zu erläutern, um ein Verständnis für die Vielfalt und die Funktionsweise zu erhalten. Diese Modularität erlaubt eine flexible Anpassung an verschiedene Zielsysteme und erhöht die Effektivität von Cyber-Angriffen.

Hypothetischer Use Case

Da es sich beim Grasshopper-Leak nicht direkt um Quellcode oder ein fertig nutzbares Exploit handelt, wurde entschieden einen hypothetischen Use Case mithilfe der Informationen der geleakten Benutzerhandbücher zu entwickeln. Dieser soll aufzeigen wie die Module in der Praxis auf einem Computer aufgesetzt und in einem realen Angriff angewendet werden könnten.

Konsequenzen des Grasshopper-Leaks

Die Arbeit hat auch die Konsequenzen des Grasshopper-Leaks beleuchten. Dies umfasst sowohl die unmittelbaren Auswirkungen auf die CIA wie auch die Reaktionen von Unternehmen, deren Produktschwachstellen durch das Leak aufgedeckt wurden. Zudem wird geprüft, ob das Tool bereits für Angriffe eingesetzt wurde.

4 Methoden

Für diese Arbeit wurde eine qualitative Literaturanalyse als methodischer Ansatz gewählt (Balzert et al., o. J.). Diese Literaturrecherche ermöglichte es, detaillierte Informationen zu den technischen und operativen Aspekten von Grasshopper sowie zu den Auswirkungen und Folgen des Leaks systematisch zu sammeln und zu analysieren.

Die Literaturrecherche konzentrierte sich hauptsächlich auf die von WikiLeaks veröffentlichten Vault 7 Dokumente, die detaillierte Informationen über die Architektur, die Funktionsweise und die spezifischen Module des Grasshopper-Tools bieten. Diese Quellen wurden systematisch durchgearbeitet und durch Sekundärquellen ergänzt, um einen umfangreichen Überblick über Grasshopper zu erhalten.

4.1 Projektmethode

Für die Literaturanalyse wurde wie folgt vorgegangen:

1. **Analyse der veröffentlichten Dokumente**: Die Dokumente aus dem Vault 7 Leak wurden gründlich durchgelesen. Dies beinhaltete die Analyse der Benutzerhandbüchern von Grasshopper, um die Komponenten von Grasshopper besser zu verstehen. Hier wurde untersucht, wie das Tool laut den veröffentlichten Dokumenten in CIA-Operationen verwendet wurde, und wie die einzelnen Komponenten funktionieren.
2. **Use Case**: Zum Aufzeigen wie Grasshopper möglicherweise in der Praxis angewendet werden kann, wurde ein hypothetischer Use Case eines möglichen Einsatzszenarios erstellt.
3. **Überprüfung der Verwendung von Grasshopper in realen Angriffen**: In diesem Teil der Analyse wurde gezielt nach Fällen gesucht, in denen Grasshopper oder ähnliche Techniken, wie sie im Leak beschrieben wurden, in bekannten Cyberangriffen eingesetzt wurden.
4. **Konsequenzen**: Abschliessend wurde bewertet, welche Auswirkungen die Veröffentlichung der Grasshopper Dokumente hatte. Dazu wurden aufgrund mangelnder Informationen Sekundärquellen, wie Pressemitteilungen, Medienberichte und veröffentlichte Briefe herangezogen.

5 Realisierung

Das Grasshopper-Leak besteht insgesamt aus 27 Benutzerhandbücher die einen CLI-Basierten Malware-Builder beschreiben. In den Benutzerhandbüchern wird das Grasshopper-Tool, wie bereits im Kapitel 2.2.2 beschrieben, in verschiedene Modul Typen, mit eigener Funktionsweise unterteilt. In dieser Arbeit werden die Verschiedenen Module anhand ihrer Funktionsweise und Einsatzbereiche analysiert, sowie auf die Konsequenzen des Grasshopper-Leaks eingegangen.

5.1 Installer

Die **Installer** können als .EXE- oder .DLL-Dateien sowohl in 32- als auch 64-Bit-Versionen erstellt werden. Diese ausführbaren Dateien werden mithilfe eines installationsspezifischen Builders konfiguriert. Die konfigurierten Payloads und Persistenzmodule werden verschleiert und als Ressourcen in den Installer eingebettet. Beim Grasshopper-Tool wird hierbei von zwei verschiedenen Installers gesprochen.

Grasshopper-Installer

Der Grasshopper-Installer ist das primäre und empfohlene Installationswerkzeug in der Grasshopper-Tool-Suite. Mithilfe einer benutzerdefinierten regelbasierten Entscheidungslogik überprüft er die Bedingungen auf dem Zielrechner, um festzulegen, ob und wie eine Installation durchgeführt wird. Dabei wählt der Installer passende Payloads und Persistenzmodule und führt diese auf dem Zielsystem aus (CIA, 2013).

Grasshopper unterstützt die folgenden booleschen Operatoren:

	True if….	False if….	Invalid if….
And	all arguments True	any arguments False	No False and any arguments Invalid
Or	Any arguments True	All arguments False	No True and any arguments Invalid
Xor	One argument True	Zero, more than one True	Any arguments invalid
Not	Argument False	argument True	Argument Invalid
Assume_true	Argument True / Invalid	Argument False	
Assume_false	Argument True	Argument False / Invalid	

Abbildung 2 - Boolesche Operatoren von Grasshopper (CIA, 2013)

Sobald ein Payload und Persistenzmodul die gegebenen Bedingungen erfüllen, wird das ausgewählte Persistenzmodul in den Arbeitsspeicher geladen und mit den benutzerdefinierten Konfigurationen initialisiert. Die zur Installation nötige ausführbare Datei wird mit einem vom Benutzer bestimmten Namen und Speicherort im Dateisystem des Zieles platziert und ausgeführt. Nachdem die Installation erfolgreich ausgeführt wurde, löscht sich die Datei wieder selbst.

Grasshopper kann so konfiguriert werden, dass er eine Protokolldatei generiert, die alle Aktivitäten der Module zusammen mit den gesammelten Daten an einem vom Benutzer festgelegten Ort auf der Festplatte speichert (CIA, 2013).

Cricket-Installer

Der Cricket-Installer ist ein alternativer Installer und verwendet ein bestimmtes Persistenzmodul, um eine bestimmte Payload zu installieren. Im Gegensatz zum Grasshopper-Installer bestimmt der Benutzer direkt das geeignete Modul und die Payload für das Zielobjekt (CIA, 2013).

Feature	**Grasshopper-Installer**	**Cricket-Installer**
Zweck	Primärer Installer für Persistenzmodule	Alternativer Installer für Persistenzmodule
Ausführbare Payloads	EXE, DLL	EXE, DLL
Payload-Auswahl	Dynamische Auswahl durch den Operator	Vordefinierte Auswahl ohne Operator-Interaktion
Initialisierung	Benutzerdefinierte Konfiguration (Regelfestlegung)	Standardkonfiguration
Schritte	1. Auswahl von Payload und Modul 2. Laden 3. Ausführen 4. Bereinigung	1. Laden 2. Ausführen 3. Bereinigung
Fussabdruck	Minimal, EXE löscht sich selbst	Minimal, EXE löscht sich selbst
Geeignet für	Komplexe Angriffe, spezifische Anpassungen	Schnelle Implementierungen

Abbildung 3 - Grasshopper-Installer vs. Cricket-Installer

Verwendung der Installer

Der Grasshopper Installer eignet sich, wenn die Auswahl von Payloads und Persistenzmodulen gezielt an ein System angepasst werden muss, um komplexe Angriffe zu ermöglichen. Die flexible Konfiguration und die regelbasierte Auswahl erhöhen dabei die Effektivität des Angriffs.

Im Gegensatz dazu eignet sich der Cricket Installer für schnellere, unkomplizierte Implementierungen, bei denen das ausgewählte Persistenzmodul direkt ausgeführt wird, ohne dass spezifische Entscheidungen mithilfe des Zielsystems getroffen werden müssen.

5.2 Persistenzmodule in Grasshopper

Das Grasshopper-Tool enthält verschiedene Persistenzmodule und Soft-Persistenzmodule, um Schadcodes dauerhaft oder temporär in einem Zielsystem zu speichern.

Die Persistenzmodule von Grasshopper, integrieren sich tief in den Systemprozess und ermöglichen es Angreifern, sich dauerhaft auf einem kompromittierten System einzunisten. Diese Module verwenden verschiedene Techniken, um sicherzustellen, dass die Payloads auch nach einem Neustart des Systems oder nach dem Trennen der Verbindung zwischen Angreifer und Opfer weiterhin aktiv bleiben. Die Soft-Persistenzmodule sind weniger tief in das System integriert und lassen sich einfacher entfernen.

5.2.1 Persistenzmodule

Buffalo: Dieses Modul registriert sich als Windows-Dienst, indem es DLL-Dateien in den Systemeinstellungen speichert und die Payload mit Administratorrechten ausführt. Nach der Installation benötigte es in der Regel einen Neustart, um die Payload auszuführen. Buffalo erstellt in der Microsoft Management Console (MMC) einen Dienst mit einem benutzerdefinierten Namen und harmlos wirkenden Beschreibung, um die wahre Funktion des Diensts zu verschleiern (CIA, 2012d).

Bamboo: Bamboo arbeitet ähnlich wie Buffalo, verwendet jedoch zusätzlich eine Technik namens „Service Hijacking“, um Payloads direkt nach der Installation zu starten, ohne dass ein Neustart nötig ist. Dabei wird ein bestehender Windows-Dienst übernommen, um den Payload auszuführen. Sobald die Payload ausgeführt wurde, wird der Dienst mithilfe einer «Unhijack-DLL» wieder in den Originalzustand zurückversetzt um Spuren zu verwischen (CIA, 2012d).

Crab: Crab installiert Payload als Windows-Dienst in den Systemeinstellungen und startet den Schadcode bei jedem Systemstart mit Administratorrechten. Das Modul hinterlässt jedoch unverschlüsselte Dateien auf dem Dateisystem und erstellt einen Eintrag in den Systemeinstellungen, der den Dienst beschreibt, wodurch es weniger unaufällig ist (CIA, 2012a).

NetMan: NetMan integriert sich in den Netzwerkverbindungs-Manager von Windows. Dabei wird ein kleines Programm (Stub) in den Service integriert, das den Payload bei jedem Systemstart lädt und ausführt, wodurch die Payload unaufällig im Hintergrund aktiv bleibt. Falls die Payload nicht mehr verfügbar ist, entfernt sich NetMan selbst (CIA, 2012b).

Service DLL: Dieses Modul speichert Payload manuell als Windows-Dienst-DLL im Net Service Host (netsvcs), wodurch die Payload automatisch bei jedem Neustart des Systems ausgeführt wird. Die Payload kann dabei direkt aus dem Speicher geladen werden oder als Datei hinterlegt sein. Service-DLL bietet versteckte Optionen, wie verschiedene Dateiendungen, um die eigentliche Funktion des Moduls zu verbergen. Damit die Payload direkt nach der Installation gestartet werden kann, verwendet Service-DLL «Service-Hijacking» und eine «Unhijack-DLL» (CIA, o. J.-c).

Stolen Goods (SG2): Das Modul nutzt Treiber und DLLs, um die Payload dauerhaft auf dem Zielsystem zu verankern. SG2 installiert eine kleine DLL-Komponente, die den Payload automatisch lädt, sobald ein spezifischer System-Trigger ausgelöst wird, und startet diesen mit einer kurzen Verzögerung zur Stabilitätskontrolle. Um die Erkennung durch Sicherheitssoftware zu erschweren, ist der Treiber mithilfe von XOR-Verschlüsselung verschleiert, während die Payload-Komponenten selbst mit einem Host-Schlüssel gesichert sind, der auf BIOS-Informationen basiert. Sollte sich diese Systemumgebung ändern, schlägt die Payload-Entschlüsselung fehl und SG2 deinstalliert sich automatisch (CIA, 2014).

Wheat: Das Wheat-Modul registriert eine Payload als Windows-Treiber in den Systemeinstellungen, damit diese bei jedem Systemstart automatisch geladen wird. Nach der Installation hat das Modul keine weitere Interaktion mit die Payload (CIA, 2012f).

ServiceProxy: Das Modul nutzt Windows-Dienste, um Payloads auszuführen. Dazu wird eine kleine DLL-Komponente in den Systemeinstellungen ablegt, die einen Net Service Host (netsvcs) imitiert. Die DLL startet den Payload, sobald der zugehörige Net Services Host aktiviert wird. Mithilfe des „-hijack"-Flags kann die Payload direkt ausgeführt werden, indem das Modul einen gestoppten Dienst übernimmt. Nach einem Neustart übernimmt der Proxy-Dienst die Ausführung die Payload (CIA, o. J.-d).

5.2.2 Soft-Persistenzmodule

Bermuda: Bermuda erstellt geplante Aufgaben im Windows-Aufgabenplaner (Scheduled Task), die den Payload zu festgelegten Zeiten oder bei bestimmten Ereignissen, wie dem Systemstart, mit Administratorenrechten ausführt. Dabei speichert Bermuda unverschlüsselte Dateien und Konfigurationsparameter in den geplanten Aufgaben und im Dateisystem, die bei einer fehlerhaften Installation oder Ausführung entfernt werden (CIA, 2012c).

ScheduledTask: Ähnlich wie Bermuda verwendet auch das ScheduledTask-Modul den Windows-Aufgabenplaner, um Payloads bei bestimmten Ereignissen, wie dem Systemstart, oder der Benutzeranmeldung auszuführen. Allerdings bietet ScheduledTask erweiterte Konfigurations-möglichkeiten, wie das Ausführen an einem spezifischen Datum oder innerhalb eines festgelegten Zeitraums in regelmässigen Intervallen. Ein weiterer Unterschied ist die automatische Deinstallation des Moduls, bei der nach dem Entfernen des Payloads auch die zugehörige Aufgabe und alle Dateien sicher gelöscht werden (CIA, o. J.-b).

Scrub: Scrub verwendet einen erstellten Run-Schlüssel in der Windows- Systemeinstellungen, um Payload zu installieren. Bei jeder Benutzeranmeldung wird dabei der unter dem Run Schlüssel abgelegte Payload mit den Berechtigungen des Benutzers ausgeführt. Falls die Payload nicht erfolgreich installiert werden kann, entfernt Scrub alle Komponenten und löscht die Änderungen in den Systemeinstellungen (CIA, 2012e).

WUPS: WUPS verwendet den Windows-Update-Service, um Payload temporär zu speichern und zu starten. Durch einen Neustart des Windows-Update-Diensts kann die Payload direkt nach der Installation gestartet werden. Nach der Installation wird die Payload automatisch alle 22 Stunden, sowie bei jedem Systemneustart mit Administrationsrechten ausgeführt. Falls die Payload nicht gefunden werden kann, entfernt WUPS den erstellten Registrierungseintrag und den Stub (CIA, 2012g).

5.2.3 Vergleich der verschiedenen Module

In der Untenstehenden Grafik werden die Eigenschaften der verschiedenen Module angezeigt.

Modul	Verwendungsmethode	Neustart erforderlich?	Unterstützte Payloads	Ausführungs-rechte	Vorteile/Nachteile
Buffalo	DLL in Windows-Dienst	Ja	EXE, DLL, GH1	SYSTEM	Hohe Systemrechte, sichtbar im Task-Manager
Bamboo	Service Hijacking	Nein	EXE, DLL, GH1	SYSTEM	Sofortige Ausführung, verbirgt Spuren
Crab	Windows-Dienst	Nein	EXE, DLL, GH1	SYSTEM	Direkte Kontrolle über den Dienst
NetMan	Netzwerkverbindungs-Manager	Nein	EXE, DLL, GH1	SYSTEM	Lädt Payloads im Hintergrund, weniger sichtbar
Service DLL	Registry-Einträge im netsvcs-Host	Nein	EXE, DLL	SYSTEM	Versteckte Stub-Optionen
Stolen Goods	Kombination aus Treibern und DLLs auf der Festplatte	Nein	DLL, Windows Driver (.sys)	SYSTEM	Verschlüsselung und Obfuskation, schwere Erkennung durch Sicherheitssoftware
Wheat	Registrierung eines Windows-Treibers in der Registry	Nein	Windows Driver (.sys)	SYSTEM	Keine Interaktion nach Installation
ServiceProxy	Imitiert Net Services Host (netsvcs)	Nein	EXE, DLL	SYSTEM	Lädt Payloads bei Dienststart, optionale „Hijack“-Funktion für sofortige Ausführung
Bermuda	Geplante Aufgaben (Windows Task Scheduler)	Nein	EXE, DLL, GH1	SYSTEM	Zeit-/Ereignisgesteuert
ScheduledTask	Geplante Aufgaben (Windows Task Scheduler)	Nein	EXE, DLL, GH1	SYSTEM	Zeit-/Ereignisgesteuert, nach Ausführung weiter aktiv, erweiterte Konfigurationsmöglichkeiten
Scrub	Run-Registry-Schlüssel	Nein	EXE	USER	Verwendet Benutzerrechte
WUPS	Modifikation des Windows Update Service	Nein	EXE	SYSTEM	SYSTEM-Rechte, aber sichtbar im Task-Manager

Abbildung 4 - Überblick Persistenzmodule

Im Anhang unter Kapitel 8.3 und 8.4 befindet sich eine technischere Beschreibung zu jedem der oben erwähnten Module.

5.3 Payloads

Grasshopper erstellt oder entwirft selbst keine Payloads. Es ist ein Toolset, mit dem man vorgefertigte Payloads in bestimmten Konfigurationen und Umgebungen nutzen und ausführen kann. Der Hauptzweck der Grasshopper-Module besteht darin, Ausführungsmethoden für Payloads auf Zielsystemen bereitzustellen (CIA, o. J.-a).

5.4 Grasshopper Builder

Grasshopper Builder sind Python-Skripte, die zur Erstellung von ausführbaren Installationsprogrammen verwendet werden. Diese Skripte erhalten Eingaben vom Benutzer, um Nutzlasten und Persistenzmethoden auszuwählen, die Installationsprogramme und Module zu konfigurieren und die Payloads und Module in ein Installationsprogramm zu verpacken. Nach der Erstellung wird ein Beleg generiert, der die Konfiguration dokumentiert und Kopien der verwendeten Binärdateien speichert. Für jeden Installationstyp stehen separate Builder-Skripte zur Verfügung (CIA, o. J.-a).

5.5 Postprozessor

Der **Postprozessor** ist ein Python-Skript, das verwendet wird, um binäre Protokolldateien zu dekodieren, die von einem Grasshopper-Installer erstellt wurden. Er benötigt einen Grasshopper-Beleg und das codierte Protokoll der Installation. Der Postprozessor generiert eine dekodierte Protokolldatei, die die Aktionen des Installationsprogramms dokumentiert, sowie eine XML-Datei, die alle während der Installation gesammelten Daten und die Ergebnis-Codes jedes Installationsversuchs enthält (CIA, o. J.-a).

5.6 Use Case

Um die Funktionalitäten von Grasshopper und seine Arbeitsweise zu demonstrieren, wurde ein hypothetischer Anwendungsfall erstellt. Dies geschah auf der Grundlage der verschiedenen veröffentlichten Grasshopper-Benutzerhandbücher. In diesem Szenario wurde ein Sicherheitsforscher damit beauftragt, den Netzwerkverkehr auf einem Windows-Gerät aufzuzeichnen, um mögliche Schwachstellen über einen Zeitraum von einem Monat zu bewerten. Mithilfe des Grasshopper-Tools wird ein persistenter Payload erstellt, um Netzwerkverbindungen zur Analyse protokollieren und speichern

1. Vorbereitungsphase

1.1. Grasshopper lokal aufsetzen

Der Sicherheitsforscher beginnt mit der Einrichtung von Grasshopper auf seinem lokalen System. Die veröffentlichten Grasshopper-Dokumente enthalten keine Informationen darüber, wie Grasshopper auf dem Gerät eines CIA-Mitarbeiters installiert worden wäre. Daher wird in diesem Use Case davon ausgegangen, dass das Grasshopper-Tool bereits heruntergeladen und auf dem System des Forschers installiert wurde, was den Zugang zu den einsatzfähigen Modulen, Installationsprogrammen und funktionalen Komponenten ermöglicht, die für die Erstellung und Konfiguration der Logging-Payloads erforderlich sind.

1.2 Zugriff auf das Zielgerät

Um Grasshopper oder einen anderen Payload auf einem Zielgerät laufen zu lassen, muss sich ein Anwender zunächst Zugang zu dem Gerät verschaffen. Dieser Prozess übersteigt die Möglichkeiten von Grasshopper selbst, da Grasshopper in erster Linie ein Tool für die Bereitstellung und Persistenz ist und nicht für den Erstzugang.

Der Forscher könnte zum Beispiel eine Phishing-E-Mail mit einem trojanisierten Word-Dokument verschicken. Das Dokument würde ein Makro enthalten, das, wenn es aktiviert wird, einen Power Shell-Befehl ausführt, um eine Reverse Shell zurück zum Befehls- und Kontrollserver des Angreifers einzurichten.

2. Planung des Einsatzes

Der Forscher würde zunächst die Ziele der Operation festlegen, wie z. B. die Dauer der Persistenz, die Art des Payloads und den Grad der erforderlichen Tarnung. Auf der Grundlage dieser Ziele würde der Forscher über das geeignete Installationsprogramm für das Persistenzmodul entscheiden.

In diesem Anwendungsfall wird der Grasshopper-Installer verwendet, da er einen regelbasierte Ansatz zulässt und komplexe Payloads über längere Zeiträume bereitstellen und verwalten kann.

3. Einrichtung der Konfigurationsdatei (XML und Payload)

Die Payloads selbst würden in einer vordefinierten Datei in einer Grasshopper-Bibliothek gespeichert werden.

3.1 Nutzung der vom Grasshopper-Tool verwendeten Payload

Angenommen, der Forscher hat ein Payload-DLL in C++ geschrieben, die WinPcap verwendet. WinPcap ist ein Industriestandard-Tool für den Link-Layer-Netzwerkzugriff in Windows-Umgebungen, das Anwendungen ermöglicht Netzwerkpakete zu erfassen und zu übertragen, ohne dass das Protokoll umgangen werden muss (WinPcap, o. J.). Dieser Capture-Payload wird in einer Datei gespeichert, beispielsweise „Capture.dll“.

Nachdem Grasshopper nun installiert und funktionsfähig ist, richtet der Anwender das Netman-Persistenzmodul ein, um die Capture.dll- Payload für die Traffic-Erfassung auszuführen. Wenn der Zielcomputer neu gestartet wird, lädt das Netman-Modul automatisch die Capture-DLL und führt sie aus, um die Erfassung des Netzwerkverkehrs erneut zu starten. Da die DLL bei jedem Neustart neu ausgeführt wird, kann sie so programmiert werden, dass sie bei jedem Start Daten exfiltriert. So könnte die Payload beispielsweise prüfen, ob eine Protokolldatei vorhanden ist, und diese zu Beginn jeder neuen Sitzung an einen Remote-Server übertragen. Nach erfolgreichem Upload könnte die Payload die Protokolldatei löschen oder umbenennen, um eine redundante Datenerfassung zu verhindern.

3.2 Grasshopper XML-Datei konfigurieren

Um die zeitbegrenzte Ausführung von Capture.dll mit Grasshopper's Netman-Modul einzurichten, muss die Grasshopper-XML so konfiguriert werden, dass sie die Payload mit einer eingebauten Kill-Datei zur autonomen Bereinigung nach 30 Tagen enthält.

```
<Grasshopper version="2.0">
    <GlobalRules>
        <Rule name="OSCheck" type="boolean">
            <Condition>OSVersion == "Windows 10"</Condition>
            <Action>Allow</Action>
        </Rule>
        <Rule name="ArchitectureCheck" type="boolean">
            <Condition>Architecture == "64-bit"</Condition>
            <Action>Allow</Action>
        </Rule>
    </GlobalRules>
    <Catalog>
        <Payload>
            <Name>Capture DLL</Name>
            <Description>Network Capture DLL without overt persistence. </Description>
            <RuleData>
                <DefaultRule>default.rule</DefaultRule>
            </RuleData>
            <UUID>e5b6a123e8df4b72a9c6f8b19e6c890f</UUID>
            <Type bitness="64" format="dll" run_level="system"/>
            <Parameters prompt="no"/>
            <Obfuscate type="reorder">
                <MinBlockSize>25</MinBlockSize>
                <MaxBlockSize>75</MaxBlockSize>
            </Obfuscate>
        </Payload>

        <PersistenceModule>
            <Name>NetMan Capture DLL</Name>
            <Method>Network Connections Service</Method>
            <Description>Using NetMan for DLL Persistence</Description>
            <Interface>gh1</Interface>
            <Rule>NetMan.rule</Rule>
            <Handler>NetMan.py</Handler>
            <Binary64>..\common\PM-NetMan-64.dll</Binary64>
            <UUID>533eb9283e34414e8e1663d46af9d350</UUID>
            <Settings>
                <RunMode>memory</RunMode>
            </Settings>
            <SupportedTypes>
                <Type format="dll" bitness="64" run_level="system"/>
            </SupportedTypes>
            <KillFile>C:\Windows\Temp\kill_capture</KillFile>
        </PersistenceModule>
    </Catalog>
</Grasshopper>
```

Abbildung 5 - XML-Konfigurationsdatei

4. Konfigurierte Grasshopper-Datei auf Zielgerät anwenden

Mit Hilfe der Reverse Shell lädt der Forscher Grasshopper herunter und installiert es auf dem Zielgerät.

Nach dem Installieren führt der Forscher die ausführbare Datei von Grasshopper (gh.exe) oder das angegebene Modul direkt von der Befehlszeile der Remote-Shell aus.

```
gh.exe add component netman -p "C:\Path\To\Capture.dll"
```

Abbildung 6 - Grasshopper Ausführungsbefehl

Die geplante Aufgabe für die Kill-Datei wird mit einem Powershell-Befehl über die Reverse Shell eingerichtet. Er stellt sicher, dass kill_capture nach 30 Tagen erstellt wird und Grasshopper dazu auffordert, die Nutzlast zu deinstallieren und die Persistenz zu löschen.

```
$trigger = New-ScheduledTaskTrigger -Once -At (Get-Date).AddDays(30)

$action = New-ScheduledTaskAction -Execute "powershell.exe" -Argument "New-Item -Path
'C:\Windows\Temp\kill_capture' -ItemType File"

Register-ScheduledTask -Action $action -Trigger $trigger -TaskName "CaptureKillTask" -
Description "Creates kill file for capture cleanup"
```

Abbildung 7 - Befehl zur Kill-Datei Erstellung

Eine Beschreibung der in der XML-Datei verwendeten Befehle (Abbildung 5) sowie der Szenarien, in denen der Forscher die anderen Module verwenden würde, befindet sich im Anhang unter Kapitel 8.5.

5.7 Konsequenzen des Leaks

Es konnten keine genau detaillierten Angaben über die Konsequenzen des Grasshopper-Leaks im Speziellen gefunden werden.

5.7.1 Initiale Konsequenzen

Nach der Veröffentlichung der Vault 7 Leaks haben die CIA und das FBI eine interne Untersuchung eingeleitet, jedoch verweigerte die CIA einen öffentlichen Kommentar zur Echtheit der geleakten Dokumente oder zum Stand der Ermittlung (Kovacs, 2017). In einem späteren Brief des Senators Wyden im Jahr 2020 wurden jedoch mehrere Sicherheitsrisiken und fehlende Sicherheitsvorkehrungen der CIA erwähnt, die ein solches Leak ermöglichten und durch den Wikileaks Task Force Report belegt wurden. Die CIA wurde darin aufgefordert nötige Massnahmen wie eine obligatorische Benutzerüberwachung und Multifactor Authentication einzuführen. Zusätzlich mussten die Datenzugriffe stärker beschränkt sowie die interne Kultur der CIA verändert werden (Wyden, 2020).

Die CIA warnte, dass solche Enthüllungen von geheimen Dokumenten die Sicherheit der Nation gefährden könnten, da sie potenziellen Gegnern wertvolle Informationen und Werkzeuge zur Verfügung stellen, die gegen die USA eingesetzt werden könnten (Kovacs, 2017).

Für betroffene Technologieunternehmen wie Cisco, Apple und Google waren die Folgen erheblich. WikiLeaks bot ihnen technische Details zu den Schwachstellen an, was einige, darunter Cisco, dazu veranlasste, sofort Patches bereitzustellen. Andere Unternehmen zögerten jedoch, die geleakten Informationen zu nutzen, da rechtliche Risiken bestanden (Kovacs, 2017b; Bitdefender, 2018).

Apple reagierte mit einer Stellungnahme, in der das Unternehmen betonte, dass zahlreiche der Schwachstellen bereits durch vorherige Updates behoben worden seien und das Sicherheitsteam kontinuierlich daran arbeite, verbleibende Sicherheitslücken zu schliessen. Das Unternehmen unterstrich seine Verpflichtung, die Privatsphäre und Sicherheit der Nutzer zu gewährleisten, und riet allen iOS-Anwendern dazu, die neueste Version des Betriebssystems zu installieren, um optimal geschützt zu sein (Gadgets 360, 2017).

Andere Unternehmen wie Microsoft, Google, Samsung und WhatsApp äusserten sich ebenfalls und versicherten, die Enthüllungen eingehend zu prüfen, ohne jedoch konkrete Massnahmen oder Ergebnisse bekanntzugeben (Swartz, 2017).

5.7.2 Langfristige Konsequenzen

Obwohl es keine offiziellen oder bestätigten Angriffe gibt, die Grasshopper verwendet haben, liefern die detaillierten Anleitungen und modularen Komponenten des Leaks eine Grundlage, um ein eigenes Malware-Tool zu entwickeln (Cimpanu, 2017).

Ein Beispiel hierfür ist das von Wayne Ronaldson entwickelte Tool „Overwatch Offensive“. Ronaldson nutzte die Informationen aus den Vault 7 Dokumenten, um eine eigene Version eines Überwachungsprogramms zu entwickeln, das ähnliche Funktionen enthält, die auch in Grasshopper verwendet werden (Stone, 2019), (RSA Conference, o. J.). In ähnlicher Weise zeigt der hypothetische Use Case in Kapitel 5.6, wie Grasshopper genutzt werden könnte, um eine persistente Malware zu erstellen, die unauffällig Netzwerke überwacht und Informationen sammelt.

Dies zeigt, dass auf Grund des Leaks ein spezialisiertes Spionage-Tool entwickelt werden könnte, das in realen Angriffsszenarien eingesetzt werden könnte.

Eine weitere potenzielle langfristige Auswirkung von Vault 7 liegt weniger in der Veröffentlichung von spezifischem Quellcode und Tools als vielmehr in der Offenlegung der zugrunde liegenden Techniken und Methoden, die von der CIA für ihre Cyberoperationen entwickelt wurden. Während der offengelegte Code durch Sicherheitsupdates entschärft oder veraltet werden könnte, liegt der eigentliche Schaden in der Enthüllung der Strategien und Prinzipien, die diesen Tools zugrunde liegen. Ursprünglich für verdeckte Operationen entwickelt, könnten diese fortschrittlichen Techniken nun Teil des globalen Arsenals von Cyberkriminellen werden, was zu ausgefeilteren und schwerer zu entdeckenden Angriffen führt (Mimran, o. J.).

6 Evaluation und Validation

In der Arbeit wurde ein Überblick über das Vault 7 Leak im Allgemeinen gegeben und das Grasshopper-Tool wurde anhand der Funktionsweise und Einsatzmöglichkeiten dargestellt.

Die Konsequenzen des Grasshopper-Leaks konnten nicht detailliert aufgezeigt werden, da die Suche nach Angriffen, in denen Grasshopper verwendet wurden oder was die direkten Folgen dieses speziellen Leaks sind, aus mehreren Gründen unerfolgreich war. Einer dieser Gründe ist, dass Firmen oft keine Details zu Cybervorfällen oder gepatchten Schwachstellen veröffentlichen. Dies kann aus dem Grund sein, keine Panik unter den Anwendern auszulösen oder auch das Vertrauen der Stakeholder und Kunden zu wahren.

Aufgrund der wenigen Angaben ist es schwer herauszufinden, was für ein Angriff stattgefunden hat und welche Methoden dabei verwendet wurden. Es konnten keine bestätigten Fälle identifiziert werden, jedoch wurden Videos gefunden, in denen Personen die veröffentlichten Informationen von dem Vault 7 Leak nutzen, um eigene Hacking-Tools zu entwickeln, wie das Tool „Overwatch Offensive“.

Diese Videos, sowie der im Abschnitt 5.6 beschriebene Use Case deuten darauf hin, dass es möglich ist, die im Leak veröffentlichte Methodik und das Wissen, für eigene Zwecke wiederzuverwenden und neu zu entwickeln. Dies bedeutet, dass potenzielle Angreifer auch Informationen verwenden könnten, um entweder ihre eigene Version der Tools anzupassen oder die Methoden zu nutzen, um eigene Operationen durchzuführen.

Ein weiterer Grund, der die Suche nach verlässlichen Quellen erschwerte, ist, dass es weitere Gruppen und Organisationen gibt die ebenfalls den Namen Grasshopper tragen, wie beispielsweise die Operation «Supposed Grasshopper» (Khaitan, 2024).

Die Fragestellung „ob und wie Grasshopper in realen Cyberangriffen genutzt wurde“ konnte daher nicht zweifelsfrei beantwortet werden. Es konnte lediglich aufgezeigt werden, dass es möglich sein kann ein Malware-Tool aufgrund des Leaks zu erstellen.

7 Ausblick

Zukunft

Das Verwenden von Tools wie Grasshopper kann in Zukunft nicht verhindert werden, allerdings können mithilfe von präventiven und reaktiven Massnahmen die Risiken durch Daten Leaks minimiert werden. Sicherheitsunternehmen könnten beispielsweise ihre internen Prozesse und Richtlinien bezüglich Geheimhaltung und den Zugang zu sensible Informationen verstärken. Dabei ist es wichtig, die Bedrohungslandschaften durch ein Monitoring zu überwachen, um Sicherheitslücken schnell identifizieren und beheben zu können.

Um Leaks zu verhindern, könnten Unternehmen strengere Sicherheitsmassnahmen für den Umgang mit vertraulichen Dokumenten und Tools einführen. Beispielsweise könnten alle Dokumente und Benutzerhandbücher mit einem unveränderbaren Wasserzeichen versehen werden, das den jeweiligen Mitarbeiter oder die Abteilung identifiziert, die Zugriff auf das Dokument haben. Dadurch könnte im Falle eines Leaks nachvollzogen werden, woher die Daten stammen.

Weitere Ideen

Im Rahmen dieser Arbeit war es nicht möglich, eine umfassende Analyse des gesamten Grasshopper-Leaks durchzuführen, da der Scope und die Kapazitäten begrenzt waren. Daher wurde der Fokus auf die Untersuchung des Grasshopper-Tools gelegt. Für eine vertiefende Forschung könnte in einer zukünftigen Arbeit das Grasshopper-Tool weiter im Detail analysiert werden. Trotz der Tatsache, dass kein direkter Quellcode oder ausführbarer Exploit veröffentlicht wurde, könnten die detaillierten Informationen aus den Benutzerhandbüchern möglicherweise genutzt werden, um ein ähnliches Tool wie Grasshopper selbst zu erstellen.

Eine weitere Möglichkeit für zukünftige Arbeiten wäre eine Analyse von potenziellen Einsätzen des Grasshopper-Tools. Dazu könnten zum Beispiel Pressemitteilungen und Sicherheitsmeldungen von Unternehmen aus dem Zeitraum des Leaks gezielt analysiert werden. Auf diese Weise könnte geprüft werden, ob bestimmte Schwachstellen zeitnah nach dem Leak gepatcht wurden oder ob Angriffe stattfanden, die auf Grasshopper hindeuten könnten.

8 Anhang

8.1 Überblick Vault 7

Protego (7. September 2017)

Protego steuert Raketen, indem es mehrere Mikrocontroller nutzt, die über verschlüsselte Kanäle kommunizieren. Der "Master Processor" empfängt Signale von einem Beacon, um zu prüfen, ob das Flugzeug im richtigen Bereich ist, ein GPS-Signal empfangen wird und die Zeit im definierten Operationsfenster liegt. Wenn alle Signale "wahr" sind, erlaubt das System den Raketenstart. Falls das Flugzeug das Zielgebiet verlässt oder die Rakete verloren geht, zerstört Protego automatisch die Verschlüsselungsschlüssel, um einen unbefugten Start zu verhindern (WikiLeaks, 2017c).

Angelfire (31. August 2017)

Angelfire ist ein CIA-Tool für Windows-Computer (XP, Win7), das Schadcode lädt und ausführt. Solartime verändert den Startbereich, sodass Wolfcreek beim Hochfahren ausgeführt wird und weitere schadhafte Programme lädt. Keystone startet diese Programme. BadMFS ist ein verstecktes Dateisystem, das die schädlichen Dateien speichert und verschlüsselt. Das Windows Transitory File System erstellt temporäre Dateien für die Installation oder Entfernung von Programmen (WikiLeaks, 2017c).

ExpressLane (24. August 2017)

ExpressLane ist ein Tool, das biometrische Daten aus Systemen extrahiert, die von Partnerdiensten wie der NSA, dem FBI und dem DHS genutzt werden. Das Tool wird unter dem Vorwand eines Software-Updates installiert und verschleiert die Datenextraktion hinter einem Windows-Installationsbildschirm. Die verwendeten Systeme stammen von Cross Match, einem US-Unternehmen für biometrische Software (WikiLeaks, 2017c).

CouchPotato (10. August 2017)

CouchPotato ist ein CIA-Tool, das RTSP/H.264-Video-Streams aufnimmt. Es kann entweder ganze Videos als AVI-Dateien speichern oder JPG-Bilder von sich verändernden Frames machen. Das Tool nutzt ffmpeg für das Codieren und Decodieren von Videos und Bildern und wird in einem ICE v3 Fire and Collect-Loader gestartet (WikiLeaks, 2017c).

Dumbo (3. August 2017)

Dumbo ist ein CIA-Tool, das speziell für die Physical Access Group (PAG) entwickelt wurde. Es ermöglicht, Prozesse auf Zielcomputern mit Windows-Betriebssystemen zu unterbrechen, die mit Webcams oder Mikrofonen arbeiten, um Videoaufzeichnungen zu manipulieren oder zu löschen. Dadurch können Aufnahmen zerstört oder gefälscht werden, um Beweise für eine CIA-Operation zu eliminieren. Das Tool wird über ein USB-Laufwerk mit Administratorrechten ausgeführt und unterstützt Windows XP, Vista und neuere 32-Bit-Versionen (WikiLeaks, 2017c).

Imperial (27. Juli 2017)

Das Imperial-Projekt der CIA umfasst mehrere Malware-Tools: Achilles modifiziert OS X Disk-Images, um Schadsoftware auszuführen, Aeris ist ein Implantat für POSIX-Systeme, das Datei-Exfiltration und verschlüsselte Kommunikation bietet, und SeaPea ist ein OS X-Rootkit, das Prozesse und Dateien versteckt. Alle Tools ermöglichen eine verborgene Kontrolle über die betroffenen Systeme (WikiLeaks, 2017c).

UCL / Raytheon (19. Juli 2017)

Das UCL / Raytheon-Projekt bezieht sich auf die "UMBRAGE Component Library" (UCL), eine Initiative, die von Raytheon Blackbird Technologies in Zusammenarbeit mit der CIA durchgeführt wurde. Die veröffentlichten Dokumente, die zwischen 2014 und 2015 eingereicht wurden, enthalten Konzepte und Bewertungen von Malware-Angriffsvektoren. Raytheon Blackbird Technologies arbeitete als "Technologie-Scout" für die CIA, indem sie bestehende Malware-Angriffe untersuchten und Empfehlungen für neue Entwicklungen in der CIA-Malwareforschung abgaben (WikiLeaks, 2017c).

Highrise (13. Juli 2017)

Highrise ist eine Android-App, die für Geräte mit Android-Versionen 4.0 bis 4.3 entwickelt wurde und als SMS-Redirector fungiert. Sie wird in CIA-Operationen eingesetzt, die SMS-Nachrichten zur Kommunikation zwischen Implantaten und Listening Posts (LPs) nutzen. Highrise wirkt dabei als Proxy für eingehende und ausgehende SMS-Nachrichten, indem sie diese über das Internet zu einem LP weiterleitet. Auf diese Weise bietet Highrise eine sicherere Kommunikationsverbindung zwischen dem Feldoperator und dem LP, indem es eine TLS/SSL-verschlüsselte Internetverbindung verwendet, die die Trennung zwischen den Zielgeräten und dem Listening Post verstärkt (WikiLeaks, 2017c).

BothanSpy (6. Juli 2017)

BothanSpy und Gyrfalcon sind zwei ähnliche CIA-Malware-Tools, die entwickelt wurden, um SSH-Anmeldedaten abzufangen und zu exfiltrieren. Beide Implantate zielen auf die SSH-Client-Software ab, wobei BothanSpy für Windows (Xshell) und Gyrfalcon für Linux (OpenSSH) konzipiert sind. Beide Tools können Benutzernamen und Passwörter von aktiven SSH-Sitzungen stehlen, wobei BothanSpy dies über eine Erweiterung für das Windows-Programm Shellterm 3.x macht und Gyrfalcon ein Rootkit verwendet, um sich auf Linux-Systemen zu installieren. Die gesammelten Daten werden in beiden Fällen verschlüsselt gespeichert und später exfiltriert, entweder über CIA-Server oder durch andere Methoden, die eine spätere Datenübertragung ermöglichen. Die Hauptunterschiede liegen in den Zielplattformen und den spezifischen Techniken der Installation und Konfiguration (WikiLeaks, 2017c).

OutlawCountry (30. Juni 2017)

OutlawCountry ist ein CIA-Malware-Tool, das Linux-Computer angreift und den gesamten ausgehenden Netzwerkverkehr auf von der CIA kontrollierte Maschinen umleitet. Die Malware besteht aus einem Kernel-Modul, das eine versteckte Netfilter-Tabelle auf dem Zielsystem erstellt. Ein Operator kann Regeln hinzufügen, die bestehende Netfilter-/iptables-Regeln überschreiben und für Benutzer sowie Administratoren unsichtbar sind. Die Installation und Persistenz des Moduls erfolgt über CIA-Exploits und Backdoors. Die erste Version von OutlawCountry (v1.0) unterstützt 64-Bit CentOS/RHEL 6.x und fügt covert DNAT-Regeln zur PREROUTING-Kette hinzu (WikiLeaks, 2017c).

Elsa (28. Juni 2017)

ELSA ist ein CIA-Malware-Tool, das Geo-Location-Daten von WiFi-fähigen Geräten wie Laptops sammelt. Nach der Installation auf einem Zielgerät scannt die Malware regelmässig sichtbare WiFi-Zugangspunkte und zeichnet deren ESS-ID, MAC-Adresse und Signalstärke auf. Ist das Gerät mit dem Internet verbunden, nutzt ELSA öffentliche Geo-Location-Datenbanken von Google oder Microsoft, um den Standort des Geräts zu ermitteln und speichert diese Daten zusammen mit einem Zeitstempel. Die gesammelten Informationen werden verschlüsselt auf dem Gerät gespeichert und später extrahiert. Die Malware sendet die Daten nicht direkt an ein CIA-Backend, sondern erfordert, dass der Operator die Logdatei manuell abruft. ELSA kann an das Zielumfeld angepasst werden, um verschiedene Sampling-Intervalle, Logdateigrössen und Persistenzmethoden zu verwenden (WikiLeaks, 2017c).

Brutal Kangaroo (22. Juni 2017)

Brutal Kangaroo ist ein CIA-Tool, das Air-Gap-Netzwerke über USB-Sticks infiziert. Zunächst wird ein Internet-verbundener Computer mit Malware infiziert, die dann auf USB-Sticks übertragen wird. Wird der Stick auf einem geschützten System verwendet, infiziert er dieses ebenfalls. Das Tool schafft ein geheimes Netzwerk zwischen infizierten Geräten und ermöglicht die Koordination von Aufgaben und Datenaustausch. Zu den Hauptkomponenten gehören Drifting Deadline (USB-Infektion), Shattered Assurance (automatisierte Infektion) und Shadow (Permanenz- und Steuermechanismus) (WikiLeaks, 2017c).

Cherry Blossom (15. Juni 2017)

CherryBlossom ist ein CIA-Projekt, das WLAN-Geräte wie Router und Access Points kompromittiert, um Internetaktivitäten zu überwachen und Angriffe durchzuführen. Durch das Aufspielen einer speziellen Firmware wird das Gerät zu einem FlyTrap, das mit einem Command-and-Control-Server, dem CherryTree, kommuniziert. Der CherryTree verwaltet die FlyTraps und sendet Aufträge, wie das Abfangen von Netzwerkverkehr, das Durchführen von Exploits oder das Einrichten von VPN-Verbindungen für weitere Angriffe. FlyTraps können auch gezielt Browserverkehr umleiten und Daten exfiltrieren (WikiLeaks, 2017c).

Pandemic (1. Juni 2017)

Pandemic ist ein persistent implantiertes Malware-Tool der CIA für Microsoft Windows, das auf Maschinen in lokalen Netzwerken abzielt, die Dateien mit entfernten Nutzern teilen. Es ersetzt den Anwendungscode eines Programms durch eine trojanisierte Version, wenn dieses Programm von einem infizierten Rechner abgerufen wird. Dabei bleibt die Originaldatei auf dem Server unverändert, die Modifikation erfolgt nur während des Transports zum Zielsystem. Pandemic kann bis zu 20 Programme mit einer maximalen Grösse von 800 MB für ausgewählte Zielcomputer ersetzen. Das infizierte System fungiert als "Patient Zero" und verbreitet die Malware, wenn Nutzer Programme von diesem Server ausführen. Es ist auch technisch möglich, dass sich andere Computer im Netzwerk durch das Bereitstellen von Dateien zu neuen Pandemie-Servern entwickeln (WikiLeaks, 2017c).

Athena (19. Mai 2017)

Athena ist ein Malware-Tool der CIA, das für entfernte Beaconing- und Loader-Funktionen auf Windows-basierten Zielsystemen (Windows XP bis Windows 10) entwickelt wurde. Nach der Installation ermöglicht es die Kommunikation mit einem Command-and-Control-Server, das Laden und Entladen von Schadsoftware im Arbeitsspeicher für spezifische Aufgaben sowie den Austausch von Dateien in einem bestimmten Verzeichnis des Zielsystems. Der Operator kann die Einstellungen während des Einsatzes anpassen, um die Malware für die jeweilige Operation zu konfigurieren. Athena wurde in Zusammenarbeit mit Siege Technologies, einem Cybersecurity-Unternehmen, entwickelt (WikiLeaks, 2017c).

AfterMidnight (12. Mai 2017)

AfterMidnight ist ein Malware-Framework der CIA, das dynamisches Nachladen und Ausführen von Schadsoftware auf Zielsystemen ermöglicht. Es tarnt sich als Windows-Service-DLL und kommuniziert über ein HTTPS-basiertes System namens "Octopus" mit einem zentralen Steuerungsserver. Das Framework verwendet sogenannte "Gremlins" – kleine, versteckte Module –, um Software zu manipulieren, Daten zu sammeln oder interne Dienste bereitzustellen. Ein besonderes Modul, "AlphaGremlin", erlaubt durch eine eigene Skriptsprache die Erstellung massgeschneiderter Aufgaben (WikiLeaks, 2017c).

Archimedes (5. Mai 2017)

Archimedes ist ein Angriffstool der CIA, das dazu verwendet wird, Computer innerhalb eines lokalen Netzwerks (LAN), beispielsweise in Büros, zu kompromittieren. Es ermöglicht, den Datenverkehr eines Zielrechners über einen mit Malware infizierten und von der CIA kontrollierten Rechner umzuleiten. Dadurch wird der Webbrowser des Zielrechners unbemerkt auf einen Exploit-Server umgeleitet, während die Sitzung wie eine normale Nutzung erscheint. Das Tool nutzt bestehende Maschinen im Netzwerk, um Ziele unter Kontrolle zu bringen und weitere Angriffe zu ermöglichen (WikiLeaks, 2017c).

Scribbles (28. April 2017)

Scribbles ist ein Dokumenten-Watermarking-System der CIA, das sogenannte „Web Beacon“-Tags in Microsoft-Office-Dokumente einbettet. Es dient dazu, potenziell kopierte Dokumente von Insidern, Whistleblowern oder Journalisten zu verfolgen. Das Tool wurde offline entwickelt und erfordert, dass keine Dateien auf Zielsystemen zurückbleiben, um die operative Sicherheit zu gewährleisten. Scribbles funktioniert mit Office-Dokumenten von Version 97 bis 2016, weist jedoch Einschränkungen auf, da Wasserzeichen in alternativen Anwendungen wie OpenOffice oder LibreOffice sichtbar werden können. Sicherheitsforscher können durch den veröffentlichten Quellcode weitere Details zur Implementierung analysieren (WikiLeaks, 2017c).

Weeping Angel (21. April 2017)

Weeping Angel ist ein von der CIA entwickeltes Malware-Tool für Samsung F Series Smart-TVs, das auf einer Technologie des britischen MI5 basiert. Es ermöglicht die Aufzeichnung von Audiodaten über das eingebaute Mikrofon und speichert oder überträgt diese. Die Entwicklung erfolgte in Zusammenarbeit zwischen MI5 und CIA, die ihre Fortschritte in gemeinsamen Entwicklungsworkshops koordinierten (WikiLeaks, 2017c).

HIVE (14. April 2017)

HIVE ist eine von der CIA entwickelte Malware-Infrastruktur mit einem HTTPS-Interface, die zur Exfiltration von Daten und zum Senden von Befehlen an Zielsysteme verwendet wird. Die Kommunikation erfolgt über Tarn-Domains, um die Aktivitäten zu verschleiern. Sie dient als Back-End für verschiedene Malware-Implantate und Operationen der CIA und nutzt ein angepasstes kryptografisches Protokoll, um die Kommunikation zu schützen (WikiLeaks, 2017c).

Grasshopper (7. April 2017)

Grasshopper ist ein Tool, um benutzerspezifische Malware für Windows-Systeme zu erstellen. Es bietet verschiedene Module, wodurch es eine präzise Kontrolle über die Installation und Persistenz der Malware ermöglicht (WikiLeaks, 2017c).

Marble Framework (31. März 2017)

Marble ist ein Anti-Forensik-Framework der CIA, das aus 676 Quellcodedateien besteht. Das Marble Framework dient dazu, forensischen Ermittlern und Antiviren-Unternehmen die Zuordnung von CIA-Viren, Trojanern und Hackerangriffen zu erschweren, indem es Textfragmente innerhalb der Malware visuell verschleiert/obfuskiert. Der Quellcode enthält ebenfalls einen Deobfuscator, der die Textverschleierung rückgängig machen kann, was es Ermittlern ermöglicht, Muster in früheren Angriffen zu identifizieren. Marble unterstützt mehrere Sprachen, darunter Chinesisch, Russisch und Arabisch, was es der CIA erlaubt, eine falsche Herkunft der Malware vorzutäuschen und forensische Ermittler in die Irre zu führen (WikiLeaks, 2017c).

Dark Matter (23. März 2017)

Bei Dark Matter handelt es sich um Dokumentationen zu Techniken die Apple-Geräte auf Firmware-Ebene infizieren, sodass die Infektion auch nach einer Neuinstallation des Betriebssystem bestehen bleibt. Dabei werden Techniken wie „Sonic Screwdriver“ verwendet, um schädlichen Code beim Booten auszuführen, auch wenn Firmware-Passwörter aktiviert sind (WikiLeaks, 2017c).

8.2 Veröffentlichte Dokumente von Grasshopper

Die folgenden Dokumente wurden im Rahmen des Grasshopper-Leaks im Vault 7 Leak veröffentlicht (WikiLeaks, 2017b):

- Grasshopper-v1_1-AdminGuide
- Grasshopper-v2_0_2-UserGuide
- StolenGoods-2_1-UserGuide
- GH-Module-Null-v2_0-UserGuide
- GH-Module-Buffalo-Bamboo-v1_0-UserGuide
- GH-Module-Wheat-v1_0-UserGuide
- GH-Module-Crab-v1_0-UserGuide
- GH-Module-WUPS-v1_0-UserGuide
- GH-Module-Scrub-v1_0-UserGuide
- GH-Module-Bermuda-v1_0-UserGuide
- GH-Module-NetMan-v1_0-UserGuide
- GH-Run-v1_1-UserGuide
- GH-ServiceProxy-v1_1-UserGuide
- GH-ServiceDLL-v1_3-UserGuide
- GH-Drop-v1_0-UserGuide
- GH-ScheduledTask-v1_1-UserGuide
- IVVRR-Checklist-StolenGoods-2_0
- StolenGoods-2_0-UserGuide
- Grasshopper-v1_1-UserGuide
- GH-Run-v1_0-UserGuide
- Grasshopper-v2_0-UserGuide
- GH-ServiceDLL-v1_0-UserGuide
- GH-ServiceDLL-v1_1-UserGuide
- GH-ServiceDLL-v1_2-UserGuide
- GH-ScheduledTask-v1_0-UserGuide
- Grasshopper-v2_0_1-UserGuide
- GH-ServiceProxy-v1_0-UserGuide

8.3 Persistenzmodule

Buffalo
Buffalo lädt spezielle DLL-Dateien in einem Windows-Dienst, um Payloads dauerhaft auf dem Zielsystem zu speichern und auszuführen. Dabei registriert sich das Modul manuell als Dienst mit SYSTEM-Privilegien im Windows-Registry, um den Payload mit hohen Systemrechten auszuführen. Es erfordert in der Regel einen Neustart des Systems, um nach der Installation zu funktionieren. Im Falle eines Fehlers bei der Payload-Installation entfernt Buffalo alle bereitgestellten Komponenten sowie die Registry-Modifikationen, um Rückstände zu vermeiden (CIA, 2012d).

Bei der Ausführung einer EXE-Payload wird die Payload direkt gestartet, wobei der Prozess der Payload-Executable im Task-Manager sichtbar ist. Damit der Dienst nicht auffällt, erstellt Buffalo in der Microsoft Management Console (MMC) einen Dienst mit einem benutzerdefinierten Namen und einer harmloswirkenden Beschreibung, damit die wahre Funktion des Dienstes nicht sofort erkennbar ist (CIA, 2012d).

Bamboo
Das Bamboo-Modul funktioniert ähnlich wie Buffalo, jedoch wird zusätzlich eine Technik namens «Service Hijacking» verwendet. Dabei wird ein bereits bestehender Windows-Dienst übernommen, um den Payload auszuführen. Diese Technik erlaubt es Bamboo, die Payload direkt nach der Installation auszuführen, ohne dass ein Systemneustart erforderlich ist (CIA, 2012d).

Zusätzlich verwendet Bamboo eine sogenannte «Unhijack-DLL», die den Hijacking-Prozess steuert und den Dienst nach der Ausführung wieder in seinen ursprünglichen Zustand versetzt. Diese Unhijack-DLL wird an einem benutzerdefinierten Pfad im Dateisystem abgelegt und nach Abschluss des Installationsprozesses sofort gelöscht, um keine Spuren zu hinterlassen (CIA, 2012d).

Crab
Das Crab-Modul installiert Payloads auf einem Zielsystem, indem es einen Windows-Dienst verwendet. Bei der Installation registriert Crab sich über direkte Änderungen in der Windows-Registry als Dienst und konfiguriert Parameter wie Dienstname und Payload-Pfad. Bei jedem Systemstart wird der Crab-Dienst mit SYSTEM-Rechten ausgeführt, wobei das Verhalten je nach Payload-Typ variiert. Crab hinterlässt unverschlüsselte Binärdateien auf dem Dateisystem und erstellt einen Registry-Schlüssel, der den Dienst beschreibt und auf die Dienststub verweist (CIA, 2012a).

NetMan
Das NetMan-Modul verwendet den Windows Netzwerkverbindungs-Manager-Service, um Payloads auf einem Zielsystem dauerhaft zu halten. Es installiert einen Stub in den Service, der die Payload bei jedem Systemstart lädt und ausführt (CIA, 2012b).

Durch direkte Änderungen in der Registry registriert es eine Stub-DLL als Startup-DLL. Bei der Ausführung schleust sich der NetMan-Stub in den Prozess des Network Connection Managers ein, wodurch die Payload mit SYSTEM-Rechten gestartet wird. Das Modul kann so konfiguriert werden, dass die Payload sofort nach der Installation gestartet wird. Wenn die Payload nicht mehr verfügbar ist, entfernt sich NetMan selbst und löscht alle zugehörigen Komponenten (CIA, 2012b).

Service DLL
Das ServiceDLL-Modul hinterlegt und installiert Payloads als Windows-Service-DLL im netsvcs Service-Hosts, wodurch dieser bei jedem Start des Zielsystems ausgeführt wird. Der ServiceDLL-Stub wird manuell in den Registry-Einträgen des netsvcs Service-Hosts hinterlegt und installiert. Die Konfiguration ermöglicht es, die Payload entweder als Ressource des Stubs zu speichern und direkt aus dem Speicher zu laden (bei NOD-kompatiblen Payloads) oder als Datei im Dateisystem abzulegen (CIA, o. J.-c).

Die Stub-Varianten bieten unterschiedliche Implementierungen, die je nach Persistenzanforderungen angepasst werden können. So speichert beispielsweise Stub A die Payload als .tlb-Datei, während andere Stubs alternative Suffixe wie .hlp, .ext, .api, .lib oder .res verwenden, um die Dateibenennung zu verschleiern. Die Payload kann nicht direkt beim ersten Installieren gestartet werden, um dies zu ermöglichen muss ein bestehender Service-DLL-Einträg gehijackt werden. Dabei wird eine „Unhijack-DLL“ ins Dateisystem geschrieben, die vom Stub während der ersten Ausführung gelöscht wird (CIA, o. J.-c).

Service DLL kann automatisch oder manuell deinstalliert werden, wobei der Dienst gestoppt, alle zugehörigen Registry-Einträge entfernt und die Dateien sicher gelöscht werden.

Stolen Goods (SG2)
Stolen Goods 2.0 ist ein persistenz Modul für Grasshopper das auf componenten von einer veröffentlichten Malware namens Carberp, eines mutmasslich russischen rootkit, basiert. Dabei wurden nur wenige Code-Elemente von Carberp übernommen, die meisten Code-Elemente wurde stark angepasst oder im Stolen Goods Modul nicht verwendet (CIA, 2014).

Das Stolen Goods (SG2)-Modul nutzt eine Kombination aus Treibern und DLLs, um Payloads auf einem Zielsystem zu installieren. Es hinterlegt eine Stub-DLL im System, die die Payload automatisch lädt, sobald ein bestimmter Trigger im System erkannt wird. Die Payload wird dabei mit einer Verzögerung gestartet, um die Prozessstabilität und die Kontrolle über das kompromittierte System sicherzustellen.

Stolen Goods verwendet XOR um den auf der Festplatte installierten Stub-Treiber zu verschleiern, wodurch der Code für Sicherheitssoftware schwieriger erkennbar ist. Der Payload-Treiber und die Payload-DLL sind mit einem Host-Schlüssel verschlüsselt, der auf Informationen im BIOS Partitionsblock basiert. Sobald die BIOS Informationen geändert werden, kann die Payload nicht entschlüsselt werden und SG2 deinstalliert sich. Dies kann auftreten, wenn die Malware ausserhalb der vorgesehenen Systemumgebung ausgeführt wird oder wenn jemand versucht, das System zu verändern, um die Payload zu entschlüsseln und zu analysieren (CIA, 2014).

Durch die Verwendung von Techniken wie XOR-Verschlüsselung und der Integration in den Kernel bleibt SG2 auch in komplexen Umgebungen schwer erkennbar.

Wheat
Das Wheat-Modul installiert den Payloads als Windows-Treiber, um die Kontrolle über ein Zielsystem aufrechtzuerhalten. Bei der Installation wird der Payload-Treiber auf die Festplatte abgelegt und in das System integriert. Wheat registriert den Payload mit den benutzerdefinierten Konfigurationen als Treiber in der Windows-Registry. Die Payload wird nach der Installation bei jedem Systemstart jeweils direkt vom Betriebssystem als Windows-Treiber automatisch ausgeführt. Nach der Installation interagiert Wheat nicht mehr mit der Payload oder dem System, sondern übergibt die Verantwortung zur Ausführung dem Payload (CIA, 2012f).

Wheat verwendet keine zusätzlichen Mechanismen zur Verschleierung des Payloads oder der Treiberdatei, sondern legt die unverschlüsselte Payload-Binärdatei im Systemverzeichnis %SYSTEMROOT%\System32\drivers ab (CIA, 2012f).

ServiceProxy

Das ServiceProxy-Modul nutzt vorhandene Windows-Dienste, um eine Payload auf einem Zielsystem zu installieren und auszuführen. Dazu wird eine ServiceProxy-Stub-DLL manuell in der Windows-Registrierung abgelegt, die einen vorhandenen Net Services Host (netsvcs) imitiert. Sobald der Dienst gestartet wird, führt der Stub die Payload aus und startet den Proxy-Dienst. Die Payload wird dabei entweder als Ressource innerhalb des ServiceProxy-Stubs (bei NOD-kompatiblen Payloads) oder direkt im Dateisystem abgelegt. Damit die Payload direkt nach der Installation ausgeführt werden kann, bietet der ServiceProxy die Möglichkeit, einen gestoppter Dienst in der SCM-Datenbank mit dem "-hijack"-Flag zu übernommen. Nach einem Neustart wird der Proxy-Dienst verwendet (CIA, o. J.-d).

8.4 Soft-Persistenzmodule

Bermuda
Das Bermuda-Modul nutzt geplante Windows-Aufgaben (Scheduled Tasks), um Payloads auf einem Zielsystem zu installieren und auszuführen. Es erstellt eine geplante Aufgabe im Windows-Aufgabenplaner und konfiguriert diese so, dass eine Payload oder ein Stub-Executable-Programm zu festgelegten Zeitpunkten oder bei bestimmten Ereignissen wie Systemstart oder Benutzeranmeldung mit SYSTEM-Rechten ausgeführt wird (CIA, 2012c).

Bermuda speichert unverschlüsselte Dateien und Konfigurationsparameter im Taskplaner und im Dateisystem. Falls die Payload oder die Installation fehlschlägt, entfernt Bermuda alle damit verbundenen Komponenten, um Rückstände zu vermeiden (CIA, 2012c).

Das Modul bietet keine erweiterte Konfigurierbarkeit der Ausführungsintervalle und zielt hauptsächlich auf einmalige Aufgaben bei Systemstart oder Benutzeranmeldung ab.

ScheduledTask

Das ScheduledTask-Modul verwendet ebenfalls den Windows-Aufgabenplaner, jedoch mit erweiterten Konfigurationsmöglichkeiten. Es erstellt eine geplante Aufgabe, die eine Stub-Datei enthält, die den Payload bei bestimmten Triggern wie Systemstart oder Benutzeranmeldung ausführt. Im Gegensatz zu Bermuda ermöglicht es ScheduledTask, den Task zu einem spezifischen Datum oder innerhalb eines festgelegten Zeitraums zu aktivieren und bei Bedarf regelmässig zu wiederholen. Das Modul kann so konfiguriert werden, dass die Payload sofort ausgeführt wird oder auf eine Aktivierung durch den definierten Trigger wartet. Ein wesentlicher Unterschied zu Bermuda ist, dass ScheduledTask nach einem Neustart des Systems aktiv bleibt und eine dauerhafte Persistenz bietet. Falls die Aufgabe oder Payload nicht wie erwartet ausgeführt wird, entfernt das Modul automatisch alle zugehörigen Komponenten. Zusätzlich unterstützt ScheduledTask eine automatische Deinstallation, bei der nach dem Entfernen der Payload alle zugehörigen Dateien sicher aus dem Dateisystem gelöscht werden (CIA, o. J.-b).

Scrub
Scrub nutzt einen Windows-Registrierungs-Run-Schlüssel, um Payloads bei Benutzerlogin auszuführen. Während der Installation erstellt Scrub den Run-Schlüssel in der Windows-Registrierung und platziert den Payload sowie gegebenenfalls eine Stub-Datei am Zielort. Bei jeder Benutzeranmeldung wird das Windows-Betriebssystem alle ausführbaren Dateien die im Registrierungseintrag unter dem Run Schlüssel eingetragen sind mit den Berechtigung des Benutzers ausführen (CIA, 2012e).

Falls die Payload nicht erfolgreich installiert werden kann, entfernt Scrub alle Komponenten und löscht die vorgenommenen Änderungen in der Registrierung.

WUPS
WUPS nutzt Windows Update Service, um eine Payload auf deinem Zielsystem temporär zu speichern. Dabei wird direkt in der Windows-Registrierung ein WUPS-Stub in den Windows-Update-Service installiert und die Payload an dem festgelegten Zielort gespeichert. Falls die Installation der Payload fehlschlägt, werden alle bereitgestellten Komponenten entfernt und die Registrierungseinträge gelöscht.

Die Payload kann durch einen Neustart des Windows Update-Dienst‘s mit den Befehlen «sc stop» und *«sc start»* sofort gestartet werden. Nachdem die Payload installiert ist, wird bei jedem Systemstart und alle 22 Stunden die Payoad mit SYSTEM-Rechten ausgeführt, auch wenn der Benutzer die Update-Funktion deaktiviert hat (CIA, 2012g).

WUPS-Stub-DLL und Payload-EXE werden an benutzerspezifizierte Speicherorte geschrieben. Während dem die Payload ausgeführt wird, ist der Prozess im Task-Manager sichtbar. Falls die Payload vom Stub nicht gefunden werden kann, löscht sich WUPS selbst. Dabei wird der Registrierungseintrag und der Stub gelöscht.

8.5 Use Case Erklärung und Erweiterung

Unten stehend befindet sich die Erklärung des im Kapitel 5.6 verwendeten XML-Konfigurationsdatei des Use Cases.

```
<Grasshopper version="2.0"> Specifies the version of the Grasshopper configuration in use.
    <Catalog> Groups the payload and persistence module definitions.
        <Payload> Defines the specifics of the payload to be deployed.
            <Name> The name of the payload is "Capture DLL."
            <Description> Provides details on the payload.
            <RuleData> Contains information about the operational rules associated with the
             payload.
                <DefaultRule> Points to a pre-defined rule, which provides guidance on how
                the payload should behave or when it should activate. Rules can be customized
                for operational needs.
            </RuleData>
            <UUID> A unique identifier for this specific payload configuration.
            <Type> Specifies technical characteristics of the payload:
             bitness="64": Indicates that it is 64-bit.
             format="dll": Confirms that the payload format is a DLL.
             run_level="system": Specifies that the payload will run with system-level
             privileges.
            <Parameters prompt="no"/> Indicates that no prompts should appear when executing
             the payload, allowing for covert deployment.
            <Obfuscate> Contains instructions for obfuscating the payload.
                type="reorder": Specifies the obfuscation technique.
                <MinBlockSize> and <MaxBlockSize>: Define the minimum and maximum sizes for
                blocks when applying reordering. This helps reduce the risk of detection by
                restructuring parts of the payload into randomized segments of 25 to 75 units
                in size.

        <PersistenceModule> Defines the persistence mechanism for the payload.
            <Name> Identifies the persistence module.
            <Method> Specifies what method the persistence module uses.
            <Description> Provides additional details.
            <Interface> Defines the Grasshopper interface type.
            <Rule> A pre-defined rule, containing settings/conditions for handling this
             module.
            <Handler> Specifies the script responsible for managing the deployment and
             configuration of the chosen persistence module.
            <Binary64> Points to a binary file that the module uses to set up persistence in
             a 64-bit Windows environment. This binary likely includes code for interacting
             with the Windows registry or services.
            <UUID> A unique identifier for this persistence module.
            <Settings> Contains additional configuration options.
                <RunMode>memory Indicates that the persistence module will load the DLL into
                memory, avoiding any on-disk presence after initialization. This minimizes
                detection risk.
            <SupportedTypes> Specifies which types of payloads are compatible with this
             module.
                <Type> Specifies technical characteristics of the payload.
            <KillFile> Specifies a file path. When this file is detected, Grasshopper will
             automatically initiate cleanup procedures for the persistence setup, removing
             the payload and associated configuration. This ensures that the operation self-
             terminates if the kill file is present, minimizing potential exposure.
```

Einsatzmöglichkeiten der anderen Grasshopper Module im Use Case

1. **Drop Module**: Kann zusätzliche Konfigurationsdateien oder Protokollierungsskripte an das Ziel liefern und bei Bedarf Verzeichnisse erstellen. Nützlich für die Einrichtung von Hilfsdateien, die von anderen Modulen benötigt werden, oder zur Ergänzung der Nutzlastfunktionen.
2. **Bermuda Module**: Kann network_logger.exe so planen, dass es nur zu bestimmten Zeiten (z. B. nachts) ausgeführt wird, um Standard-Systemaufgaben zu imitieren und eine Entdeckung zu vermeiden.
3. **Buffalo Module**: Installiert die Nutzlast als DLL in einem Windows-Dienst, der einen Neustart erfordert. Dieses Modul ist ideal für Szenarien, die einen tiefen Zugriff auf die Systemebene erfordern, aber es ist auffälliger und kann die Betriebszeit stören, wenn häufige Neustarts nicht möglich sind.
4. **Bamboo Module**: Ähnlich wie Buffalo, ermöglicht jedoch die sofortige Ausführung durch die Entführung eines ruhenden Dienstes, ohne dass ein Neustart erforderlich ist, und eignet sich daher für Szenarien mit schneller Bereitstellung, bei denen die Nutzlast sofort nach der Installation aktiviert werden muss.
5. **Crab Module**: Registriert sich als Windows-Dienst und bietet dauerhaften Zugriff auf die Systemebene. Dies könnte zur Protokollierung der CPU- und Speichernutzung verwendet werden, um die Analyse des Netzwerkverkehrs durch Leistungskennzahlen des Servers zu ergänzen.
6. **NULL Module**: Führt einmalige Befehle aus, ohne Spuren zu hinterlassen. Ideal für die Erfassung eines ersten Schnappschusses offener Verbindungen oder aktiver Sitzungen, bevor die dauerhafte Protokollierung beginnt..
7. **Scrub Module**: Kann Anmeldungen überwachen und Anwendungsstarts nachverfolgen, wertvoll, wenn benutzerspezifische Interaktionen mit netzintensiven Anwendungen verfolgt werden müssen.
8. **Wheat Module**: Installiert einen benutzerdefinierten Treiber, der nützlich ist, wenn neben der Netzwerkprotokollierung auch die Überwachung von Hardware-Ereignissen wie USB-Verbindungen erforderlich ist und Einblicke in mögliche Datenübertragungen über Wechselmedien gewährt.
9. **WUPS Module**: Führt Nutzdaten im Einklang mit dem Windows Update Service aus (22-Stunden-Intervalle), ideal für die Erfassung regelmässiger Systemzustandsdaten, um Trends im Netzwerkverkehr zu kontextualisieren.
10. **Run Module**: Kann network_logger.exe sofort nach der Bereitstellung starten, um den Systemstatus und aktive Verbindungen einmalig zu erfassen und die dauerhafte Protokollierung zu ergänzen.
11. **ServiceDLL Module**: Lädt eine Nutzlast als DLL in den „netsvcs“-Prozess und bietet eine diskrete Methode zur Überwachung der Dienstaktivität. Könnte verwendet werden, um nur zu protokollieren, wenn bestimmte Netzwerkdienste aktiv sind, und so den Verkehrsmustern einen Kontext zu geben.

9 Abkürzungs-, Abbildungsverzeichnis

9.1 Abkürzungsverzeichnis

Abkürzung	Bedeutung
CIA	Central Intelligence Agency
CLI	Command Line Interface
DLL-Datei	Dynamic-Link-Library Datei
MMC	Microsoft Management Console
NOD	Network Object Database
WUPS	Windows Update Persistent Service

9.2 Abbildungsverzeichnis

10 Literaturverzeichnis

Balzert, H., Schröder, M., & Schäfer, C. (o. J.). *Wissenschafltiches Arbeiten* (2. Aufl.). Springer.

CIA. (o. J.-a). *Grasshopper v2.0.2 User Guide*. Abgerufen 26. September 2024, von https://wikileaks.org/vault7/document/#grasshopper

CIA. (o. J.-b). *Scheduled Task v1.1 Grasshopper Component User Guide*. WikiLeaks. Abgerufen 26. September 2024, von https://wikileaks.org/vault7/document/#grasshopper

CIA. (o. J.-c). *ServiceDLL v1.3 Grasshopper Component User Guide*. Abgerufen 26. September 2024, von https://wikileaks.org/vault7/document/#grasshopper

CIA. (o. J.-d). *ServiceProxy v1.1 Grasshopper Component User Guide*. Abgerufen 26. September 2024, von https://wikileaks.org/vault7/document/#grasshopper

CIA. (2012a, Juni). *Grasshopper Module Guide Crab v1.0 User Guide*. WikiLeaks. https://wikileaks.org/vault7/document/#grasshopper

CIA. (2012b, Juni). *Grasshopper Module Guide NetMan v1.0 User Guide*. https://wikileaks.org/vault7/document/#grasshopper

CIA. (2012c, Juni). *Grasshopper Module Guide Bermuda v1.0 User Guide*. https://wikileaks.org/vault7/document/#grasshopper

CIA. (2012d, Juni). *Grasshopper Module Guide Buffalo v1.0 and Bamboo v1.0 User Guide*. WikiLeaks. https://wikileaks.org/vault7/document/#grasshopper

CIA. (2012e, Juni). *Grasshopper Module Guide Scrub v1.0*. https://wikileaks.org/vault7/document/#grasshopper

CIA. (2012f, Juni). *Grasshopper Module Guide Wheat v1.0 User Guide*. https://wikileaks.org/vault7/document/#grasshopper

CIA. (2012g, Juni). *Grasshopper Module Guide WUPS v1.0 User Guide*. https://wikileaks.org/vault7/document/#grasshopper

CIA. (2013, Dezember). *Grasshopper v1.1 Users' Guide*. https://wikileaks.org/vault7/document/#grasshopper

CIA. (2014, Juli 14). *Engineering Development Group UMBRAGE: StolenGoods Version 2.1 User Manual*. https://wikileaks.org/vault7/document/#grasshopper

Cimpanu, C. *WikiLeaks Reveals Grasshopper, CIA's Builder for Windows Malware*. BleepingComputer. https://www.bleepingcomputer.com/news/security/wikileaks-reveals-grasshopper-cias-builder-for-windows-malware/

Gadgets 360. (2017, März 8). *Apple, Google, WhatsApp, and Others React to WikiLeaks CIA Dump*. Gadgets 360. https://www.gadgets360.com/mobiles/news/wikileaks-cia-vault-7-dump-apple-samsung-google-whatsapp-telegram-signal-react-1667508

Khaitan, A. (2024, Juli 3). *Supposed Grasshopper Campaign Targets Israeli Govt And Firms* [Ashish]. https://thecyberexpress.com/supposed-grasshopper-campaign/

Kovacs, E. (2017, März 9). *CIA Responds to WikiLeaks Hacking Tool Dump*. SecurityWeek. https://www.securityweek.com/cia-responds-wikileaks-hacking-tool-dump/

Mimran, M. (o. J.). *The Long-Term Threats Posed by the Vault 7 Leaks*. Abgerufen 21. Oktober 2024, von https://www.cybereason.com/blog/vault-7-leaks-long-term-threats

RSA Conference. (o. J.). *How Vault 7 Leaks Helped Develop My Own Cyberespionage Weapon—YouTube*. Abgerufen 16. November 2024, von https://www.youtube.com/watch?v=uhkUHHsKA5Y&ab_channel=RSAConference

Stone, J. (2019, Februar 27). A researcher made an elite hacking tool out of the info in the Vault 7 leak. *CyberScoop*. https://cyberscoop.com/vault-7-operation-overwatch-cia-hacking-tools-rsa-conference/

Swartz, J. (2017, März 7). *Apple, Google, Microsoft in crosshairs of WikiLeaks allegations*. USA TODAY. https://www.usatoday.com/story/tech/news/2017/03/07/apple-google-microsoft-crosshairs-wikileaks-allegations/98854320/

WikiLeaks. (o. J.). *WikiLeaks—What is WikiLeaks*. Abgerufen 26. Oktober 2024, von https://wikileaks.org/What-is-WikiLeaks.html

WikiLeaks. (2017a). *Vault7: CIA Hacking Tools Revealed*. WikiLeas. https://wikileaks.org/ciav7p1/

WikiLeaks. (2017b). *WikiLeaks—Documents*. https://wikileaks.org/vault7/document/#grasshopper

WikiLeaks. (2017c). *WikiLeaks—Vault 7: Projects*. https://wikileaks.org/vault7/#Grasshopper

WinPcap. (o. J.). *WinPcap—Home*. Abgerufen 16. November 2024, von https://www.winpcap.org/

Wyden, R. (2020, Juni 16). *United States Senate, Letter to Director of National Intelligence (DNI)*.