Hochschule Wismar
Fakultät für Ingenieurwissenschaften
Bereich Elektrotechnik und Informatik

Projektarbeit im Fern-Studiengang

„Bachelor IT-Forensik“

an der Hochschule Wismar

im Modul:

IT-Forensik-Projekt II

zum Thema:

Hotplug-Attack-Tools – Überblick und Vergleich gängiger Tools

eingereicht von: Florian Winterer

Betreuer: Prof. Dr.-Ing. Antje Raab-Düsterhöft

Wismar, den 09. Oktober 2023

Aufgabenstellung

Die Aufgabe der Projektarbeit besteht darin, einen Überblick über die Grundlagen von Hotplug-Attacks zu vermitteln. Zudem sollen gängige Hotplug-Attack-Tools vorgestellt werden. Die Tools sollen anhand bestimmter, für Angreifer relevanter Kriterien bewertet und verglichen werden. Hierdurch soll einerseits das Verständnis für Hotplug-Attacks und die damit einhergehenden Gefahren geschärft werden. Anderseits soll diese Ausarbeitung auch als Entscheidungshilfe für IT-Sicherheitsexperten bei der Auswahl solcher Tools für Penetration-Tests dienen. Abschließend soll darauf eingegangen werden, wie Hotplug-Attacks entdeckt und verhindert werden können.

Inhalt

1 Einführung und Motivation des Themas

1.1 Einführung und Definition der Forschungsfrage

Der Fokus der IT-Sicherheit liegt seit einigen Jahren auf Bedrohungen aus dem Internet. Unter den durch das Bundesamt für IT-Sicherheit (BSI) identifizierten Top-Bedrohungen für Wirtschaft bzw. Staat und Verwaltung fanden sich im Jahr 2022 fast ausschließlich Bedrohungen mit Internet-Bezug, wie Ransomware oder Schachstellen aus offenen oder falsch konfigurierten Online-Servern.[1] Dieser Fokus ist im Angesicht der zunehmenden Digitalisierung nur allzu verständlich.

Jedoch hat nicht nur die Gefahr aus dem Internet zugenommen, auch andere Angriffsmöglichkeiten haben sich weiterentwickelt. Durch immer kleiner werdende Prozessoren können mittlerweile leistungsstarke Computer in herkömmlichen USB-Sticks oder Kabeln versteckt werden. Diese harmlos aussehenden Geräte ermöglichen sogar Attacken auf Systeme, die nicht mit dem Internet verbunden sind.

Derartige Geräte, so genannte Hotplug-Attack-Tools, sind frei erhältlich und haben in den letzten Jahren einen gewissen Hype erfahren. Einzelne Tools haben mittlerweile eine regelrechte Fan-Base, welche für diese Geräte Payloads entwickelt.

Diese Arbeit soll einen Überblick über die Grundlagen von Hotplug-Attacks und gängigen Hotplug-Attack-Tools geben. Die Tools sollen anhand bestimmter, für Angreifer relevanter Kriterien bewertet und verglichen werden. Hierdurch soll einerseits das Verständnis für Hotplug-Attacks und die damit einhergehenden Gefahren geschärft werden. Anderseits soll diese Ausarbeitung auch als Entscheidungshilfe für IT-Sicherheitsexperten bei der Auswahl solcher Tools für Penetration-Tests dienen. Abschließend sollen mögliche Präventionsmaßnahmen gegen Hotplugs-Attacks vorgestellt werden.

[1] BSI-Lagebericht 2022.

1.2 Wissenschaftliches Konzept und Herangehensweise

Für einen aussagekräftigen Vergleich der verschiedenen Hotplug-Attack-Tools ist ein grundlegendes Verständnis von Hotplug-Attacks notwendig. Diese Grundlagen werden in Abschnitt 2 vermittelt. Für eine relevante Auswahl an Tools wird ferner ein Marktüberblick benötigt. Kapitel 3 stellt deshalb die gängigsten Anbieter von frei erhältlichen Hotplug-Attack-Tools vor. Da nicht alle am Markt erhältlichen Gadgets näher betrachtet werden können, wird eine Auswahl getroffen, mit dem Ziel, ein möglichst breites Spektrum von Einsatzzwecken und Anschlussmöglichkeiten abzudecken. In Abschnitt 4 werden für ein tiefergreifendes Verständnis dieser Geräte die technischen Merkmale vorgestellt und deren Merkmale anhand von Bildern und Grafiken illustriert. Die technischen Daten werden anhand von Herstellerangaben und aus Sekundärquellen zusammengetragen. Zudem werden die Tools, soweit zerstörungsfrei möglich, zerlegt und der Aufbau anhand von Fotos dokumentiert. Dies dient auch dem Abgleich mit den Herstellerangaben. Das eigentliche Kernstück der Arbeit, der Vergleich der ausgewählten Tools, anhand aus Sicht eines Angreifers relevanter Kriterien, erfolgt in Kapitel 5. Als relevante Kriterien wurden der mögliche Payload, Zugriffs- und Steuerungsmöglichkeiten, die Möglichkeiten zur Exfiltration von Daten, der Schutz vor Entdeckung sowie das Verhalten der Tools im praktischen Einsatz identifiziert. Für den Vergleich wird in einem ersten Schritt auf die technischen Beschreibungen der Tools zurückgegriffen. In einem zweiten Schritt werden die Tools anhand eines ausgewählten realitätsnahen Fallbeispiels auch praktisch verprobt. Hierdurch soll insbesondere die Eignung der Tools für den vorgesehenen Einsatz überprüft werden. Kapitel 6 und 7 widmen sich der Erkennung und Prävention von Hotplug-Attacks. Dabei werden gängige Verfahren anhand der einschlägigen Literatur vorgestellt. Die Arbeit schließt mit einem kurzen Fazit.

1.3 Motivation des Themas

Während in den letzten Jahren der Fokus der IT-Sicherheit überwiegend auf Gefahren aus dem Internet lag, hat die Gefahr, Opfer einer Hotplug-Attack zu werden, aufgrund des technischen Fortschritts deutlich zugenommen. Während

Prozessoren immer kleiner werden und damit die Möglichkeiten für Angreifer steigen, hat sich im Bereich der Sicherheit von USB-Schnittstellen sehr wenig getan. Die Grundproblematik besteht dabei darin, dass die Grundkonzeption von USB-Schnittstellen und Geräten der HID-Klasse darauf ausgelegt ist, möglichst performant zu sein. Sicherheitsaspekte wurden bei der Entwicklung hingegen unzureichend betrachtet.[2]

Aber nicht nur die technische Entwicklung hat zu einer veränderten Gefahrenlage in Bezug auf Hotplug-Attacks geführt, auch gesellschaftliche Entwicklungen haben hierzu beigetragen. Aufgrund der Corona-Pandemie arbeiten mittlerweile viele Menschen im Homeoffice, oftmals mit eigener Hardware. Hierdurch werden gängige Restriktionen hinsichtlich der eingesetzten Hardware aufgebrochen, wodurch die Gefahren für Hotplug-Attacks steigen. Eine Zuspitzung findet diese Entwicklung im Trend „bring your own device“.

Neben dem Wegfall sicherheitsrelevanter Restriktionen, bietet das verteilte Arbeiten Angreifern auch weitere Angriffsvektoren. So können gezielt Social-Engineering-Angriffe mit Hotplug-Attacks kombiniert werden, beispielsweise indem Mitarbeitern als USB-Sticks getarnte Hotplug-Attack-Tools, mit angeblich wichtigen Daten, ins Homeoffice geschickt werden.

Um die Gefahren von Hotplug-Attacks besser bewerten zu können und IT-Sicherheitsexperten eine Hilfestellung an die Hand zu geben, wurde dieses Thema gewählt.

[2] Lee, et. al., 2016, Seite 377.

2 Grundlagen von Hotplug-Attacks

2.1 Human Interface Devices (HID)

Hotplug-Attacks nutzen das systemimmanente Vertrauen von Computern gegenüber Eingabegeräten für Menschen (HIDs) aus. Sie basieren auf dem sogenannten Hot-Plugging von Human Interface Devices (HID).

Als Hot-Plugging wird das Hinzufügen von Komponenten zu Systemen im laufenden Betrieb, d.h. ohne wesentliche Unterbrechung des Systembetriebs, insbesondere ohne einen Neustart des Systems, verstanden.[3] Werden USB-Sticks, Festplatten, Monitore oder Tastauren im laufenden Betreib angeschlossen, muss das Betriebssystem auf die geänderte Hardware-Situation möglichst schnell und automatisch reagieren. Abhängig vom jeweiligen Betriebssystem sind hier verschiedene Arbeitsschritte notwendig. Beispielsweise sieht das Zusammenspiel der einzelnen Komponenten unter Linux wie folgt aus:[4]

1. Kernel: Der Kernel identifiziert Veränderungen an der Hardware, erzeugt neue Device-Dateien und startet geeignete Programme, um die Geräte zu verwalten bzw. um das Desktop-System zu benachrichtigen.
2. DeviceKit: Für manche Geräte bzw. Komponenten hilft das sogenannte DeviceKit bei der Verwaltung. Das DeviceKit besteht aus verschiedenen Bibliotheken, die üblicherweise in gleichnamige Pakete verpackt sind.
3. Desktop: Abhängig von der verwendeten Linux-Version sind verschiedene Frameworks für die Verarbeitung von D-Bus-Nachrichten zuständig.
4. D-Bus: Zur Kommunikation zwischen den verschiedenen Ebenen des Hotplug-Systems wird das D-Bus-Kommunikationssystem verwendet. Auf Basis einer Bibliothek kann die Kommunikation direkt zwischen zwei

[3] https://www.computerweekly.com/de/definition/Hot-Plug#:~:text=Jedes%20Ger%C3%A4t%2C%20das%20an%20einen,Ger%C3%A4t%20mit%20einem%20USB%2DAnschluss.

[4] Linux: Das umfassende Handbuch, Seite 585.

Programmen erfolgen. Das D-Bus-Kommunikationssystem ist außerhalb des Kernels implementiert.

Die Bezeichnung Human Interface Device (HID) definiert eine Geräteklasse, die es Benutzer erlaubt, auf Basis von generischen USB-Treibern mit einem Computer zu interagieren.[5]

Vor HID erfolgte der Anschluss von Geräten zur menschlichen Interaktion, wie Mäuse und Tastaturen, über PS/2-Anschlüsse, die streng definierte Protokolle nutzten. Diese hatten den Nachteil, dass Hardware-Innovationen entweder die Überlastung von Daten in einem vorhandenen Protokoll oder die Entwicklung nicht standardmäßiger Hardware mit einem eigenen speziellen Treiber erforderten.

HID setzte zunächst auf dem USB-Standard auf, wurde aber busunabhängig konzipiert. Die Spezifikation für HID über USB wurde 1996 vom USB Implementers Forum[6] ratifiziert.[7]

Die für die Kommunikation zwischen einem USB-Gerät und einem Computer notwendigen Komponenten sind in der nachfolgenden Grafik dargestellt:

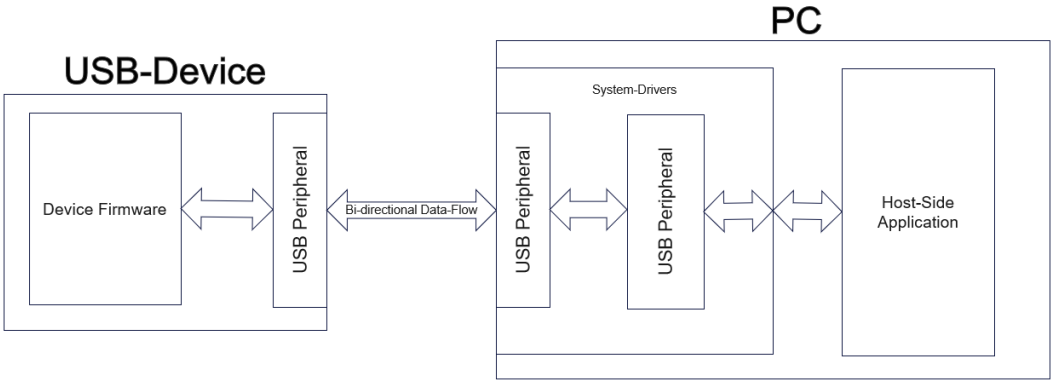


Bild 1: Kommunikation zwischen USB-Gerät und PC

Eine USB-Schnittstelle erfordert demnach drei Subsoftwaresysteme:

---

[5] https://learn.microsoft.com/en-us/windows-hardware/drivers/hid/.

[6] https://www.usb.org/about.

[7] https://learn.microsoft.com/en-us/windows-hardware/drivers/hid/.

1. Firmware des USB-Geräts
2. Betriebssystemtreiber des Hosts (PC)
3. Anwendung des Hosts

Geräte der HID-Klasse vereinfachen die Kommunikation zwischen dem USB-Gerät und Computer, indem sie einen standardisierten, flexiblen Treiber einsetzen, der in allen üblichen Betriebssystemen vorinstalliert ist.[8]

Damit dieser standardisierte Treiber genutzt werden kann, gelten folgende grundlegenden Anforderungen für die Klassifikation als HID-Gerät:

- HID-Geräte müssen über einen Control-Endpoint (Endpunkt 0) und einen Interrupt-IN-Endpoint verfügen. Viele Geräte nutzen zudem einen Interrupt-OUT-Endpoint. Die Geräte dürfen aber nicht mehr als einen IN- und einen OUT-Endpoint haben.
- Der gesamte Datenverkehr muss als standardisierter Bericht formatiert sein. Die Struktur des Berichts ist im report descriptor festgelegt.
- HID-Geräte müssen auf alle standardisierten HID- und USB-Anfragen reagieren.

Im Hinblick auf diese Anforderungen und zum besseren Verständnis von Hotplug-Attacks ist hervorzuheben, das HID-Geräte keine direkte menschliche Interaktion erfordern.[9]

Die dargestellten, der HID-Klassifikation zugrundeliegenden Anforderungen bilden den Ausgangspunkt für Hotplug-Attacks. Wird ein HID-Gerät an einen Computer angeschlossen, muss es bevor es in seinen normalen Betriebsmodus übergehen und Daten mit dem Host übertragen kann, ordnungsgemäß spezifiziert werden. Der Spezifikationsprozess besteht aus einer Reihe von Aufrufen des Hosts für Deskriptoren, die im Gerät (in der Firmware) gespeichert sind und die Fähigkeiten des Geräts beschreiben. Das Gerät muss mit Deskriptoren antworten, die einem Standardformat folgen. Den Aufbau der

[8] https://www.silabs.com/documents/public/application-notes/AN249.pdf, Seite 2.

[9] https://www.silabs.com/documents/public/application-notes/AN249.pdf, Seite 4.

Deskriptoren verdeutlicht die nachfolgende Grafik:

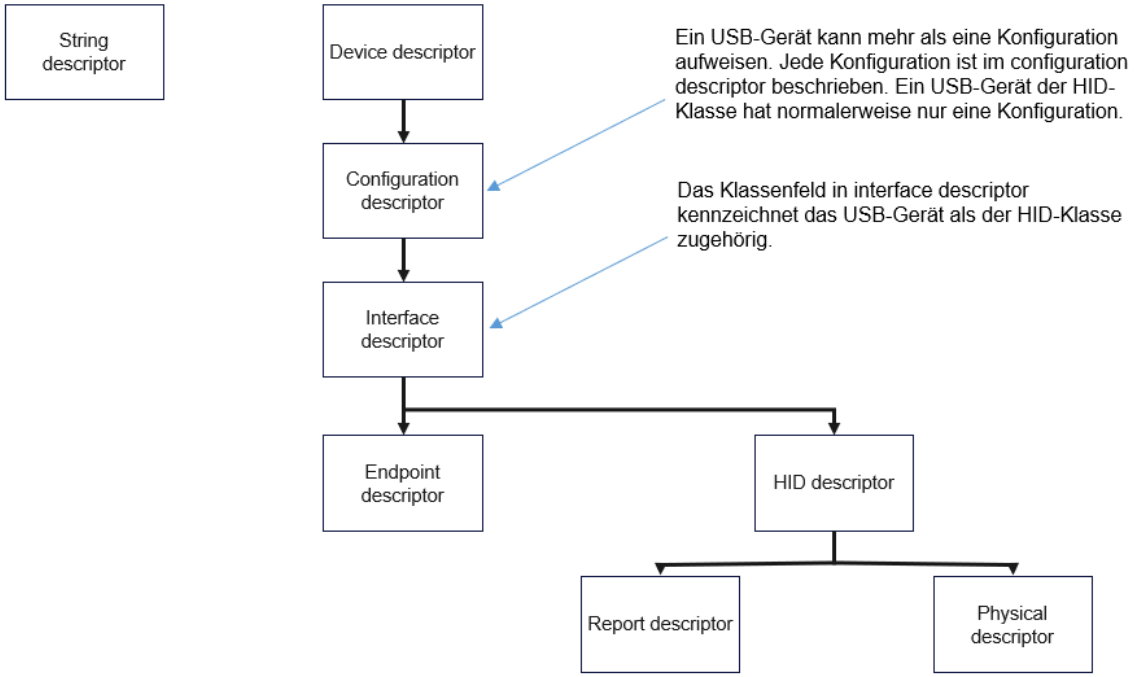


Bild 2: Device Descriptor Model

Auf der obersten Ebene enthält ein Deskriptor zwei Tabellen mit Informationen, den Geräte-Deskriptor (device descriptor) und den String-Deskriptor (string descriptor). Im Geräte-Deskriptor finden sich grundlegende Informationen über ein Gerät, wie die Produkt-ID (PID), die Anbieter-ID (VID) oder die Seriennummer.[10] [11]

Hotplug-Attacks machen sich diesen standardisierten Ablauf und Aufbau zu Nutze, indem Hotplug-Attack-Tools gefälschte Informationen, z.B. geänderte PIDs und VIPs verwenden und sich beispielsweise als USB-Tastatur tarnen. Nachdem der Computer das Hotplug-Attack-Tools fälschlicherweise als ein Gerät der HID-Klasse identifiziert hat, kann das Hotplug-Attack-Tool Payload ausführen, indem es beispielsweise Tastaturbefehle an den Computer sendet und eine sogenannte Keystroke-Injection ausführt.

[10] https://www.usb.org/sites/default/files/hid1_11.pdf, Seite 12.

[11] https://www.silabs.com/documents/public/application-notes/AN249.pdf, Seite 5.

2.2 Angriffsvektoren von Hotplug-Attacks

Die möglichen Angriffe mit Hotplug-Attack-Tools lassen sich in sechs Kategorien zusammenfassen:[12]

	Kategorie	Beschreibung
1	Malicious payload	Attacke, bei der Schadprogramm (z.B. Viren, Würmer, Trojaner) in das Zielsystem eingeschleust wird.
2	Keystroke-Injektion / Keyboard-Emmulation	Attacke, bei der sich das USB-Gerät als Tastatur ausgibt und Tastenbefehle übermittelt.[13]
3	Smartphone-based HID attacks	Angriffe auf ein Smartphone, bei denen sich das USB-Gerät als HID-Gerät ausgibt.
4	Cold boot attacks	Attacke, bei der ein Angreifer mit physischem Zugang zum Zielrechner Inhalte des Arbeitsspeichers ausliest, nachdem das System abgeschaltet wurde. Die Attacke basiert auf der Datenremanenz in gängigen RAM-Modulen, in denen sich Ladung unter bestimmten Bedingungen nicht innerhalb von Millisekunden, sondern nach und nach langsam über Sekunden bis Minuten verflüchtigt und die Dateninhalte aus den Speicherzellen eventuell nach einigen Minuten noch erfolgreich

[12] Nicho / Sabry, 2022, S. 245.

[13] https://it-forensik.fiw.hs-wismar.de/index.php/Keystroke_Injection_Attack_Tool.

		vollständig ausgelesen werden können.[14]
5	Default gateway override	Angriff, der einen Mikrocontroller verwendet, um einen USB-Ethernet-Adapter zu fälschen, um DHCP-Einstellungen außer Kraft zu setzen und lokalen Datenverkehr zu kapern.[15]
6	Hidden partition patch	Angriff, bei dem ein USB-Flash-Laufwerk so umprogrammiert wird, dass es sich wie ein normales Laufwerk verhält, wodurch eine versteckte Partition erstellt wird. Diese kann nicht formatiert werden, was eine verdeckte Datenexfiltration ermöglicht.[16]

Tabelle 1: Angriffsvektoren

In Kombination mit den vier Hauptschwachstellen

- schwache Verschlüsselung und schlechter Umgang mit Daten (poor encryption and misplacement of data),
- fehlerhafte Konfiguration von Systemen (misconfiguration of systems),
- fehlende Patches oder veraltete Software (unpatched or outdated software) und
- schwache Anmeldeinformationen (week credentials),

[14] https://de.wikipedia.org/wiki/Kaltstartattacke.

[15] Nissim, et. al. 2017, S 678.

[16] https://www.bleepingcomputer.com/news/security/heres-a-list-of-29-different-types-of-usb-attacks/#:~:text=9)%20Default%20Gateway%20Override%20%2D%20an,settings%20and%20hijack%20local%20traffic.

können folgende Angriffsvektoren identifiziert werden[17] [18]:

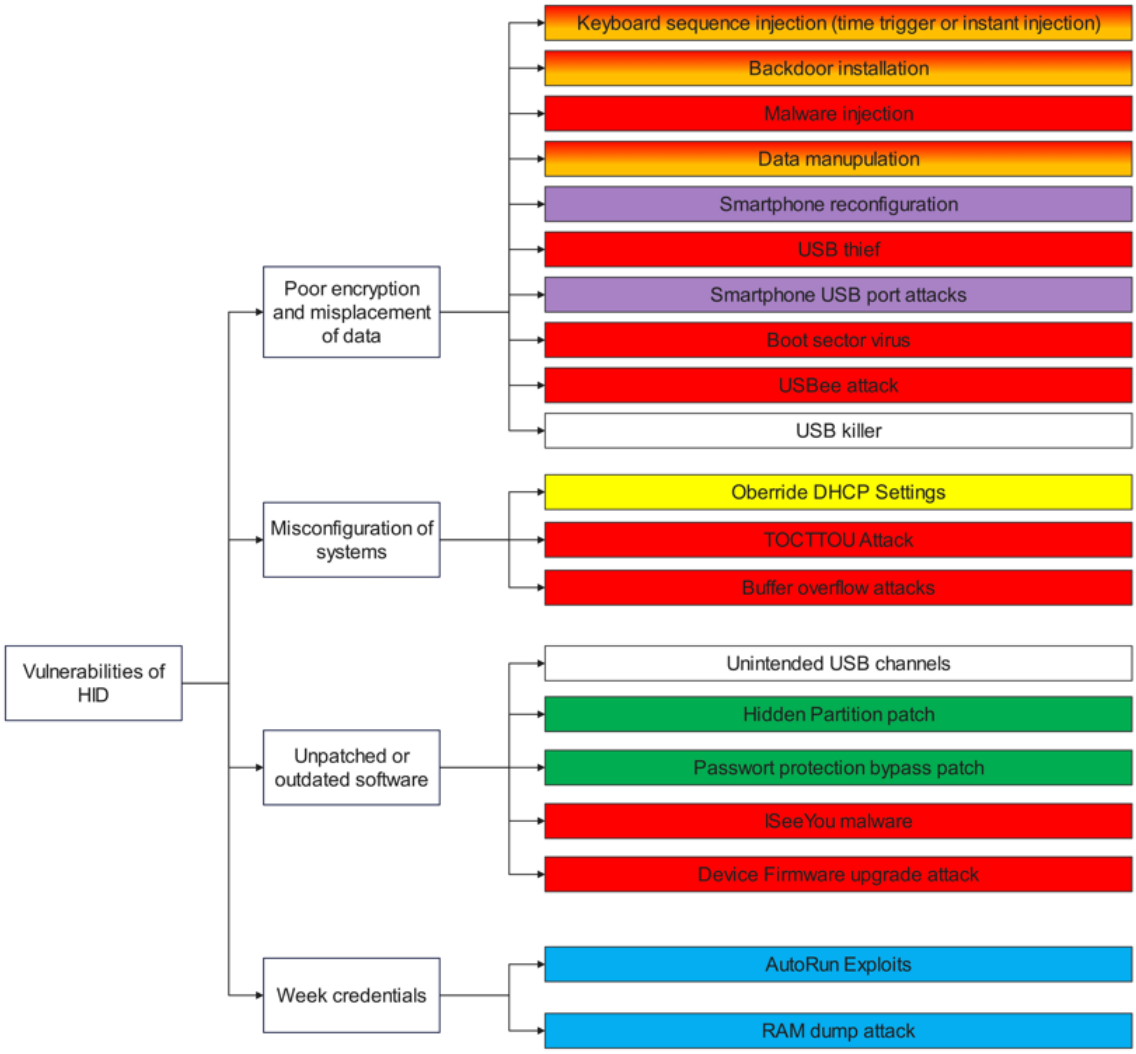


Bild 3: Angriffsvektoren für HID

2.3 Programmiersprache für Hotplug-Attack-Tools

Neben der Kenntnis der allgemeinen Funktionsweise von Hotplug-Attacks, ist für das Verständnis der nachfolgenden Ausführungen auch das Wissen um die gängige Programmiersprache für Hotplug-Attack-Tools notwendig.

Grundsätzlich kommen für die Programmierung von Hotplug-Attack-Tools

[17] Die Zuordnung der Angriffe zu den übergeordneten Kategorien erfolgt anhand der Farbe.

[18] Nicho / Sabry, 2022, S. 244.

verschiedene Programmiersprachen, wie JavaScript oder Python, in Frage. Oftmals wird als Programmiersprache allerdings DuckyScript verwendet. DuckyScript 1.0 ist eine Makro-Skriptsprache, die von Hak5 im Jahr 2010 für das Hotplug-Attack-Tool USB Rubber Ducky entwickelt wurde. DuckySkript 1.0 kennt zwei grundlegende Aktionen:

- keystroke-injection (Eingabe einer Reihe von Tastaturbefehlen) und
- delay (kurzes Anhalten).

Diese Kombination dieser Aktionen bildet die so genannte Payload, d.h. die Anweisungen an den USB Rubber Ducky.

Im Laufe der Jahre wurde die DuckyScript-Sprache um gerätespezifische Befehle ergänzt. Mit der Einführung eines weiteren Hotplug-Attack-Tools im Jahr 2017 durch Hak5, dem Bash Bunny, wurde DuckyScript mit der Shell-Skriptsprache BASH gekoppelt. Durch die Nutzung einer Linux-Basis wurden Multi-Vektor-USB-Angriffe ermöglicht.

In ähnlicher Weise war DuckyScript in verschiedenen anderen Tools von Hak5 enthalten. Aufgrund des Erfolgs von DuckyScript wurde die Skriptsprache sogar für Tools von Drittanbietern, die in Zusammenarbeit mit Hak5 entwickelt wurden, lizenziert. Hierzu gehören beispielsweise die Hotplug-Attack-Tools der O.MG-Plattform von Mischief Gadgets.

Zeitgleich mit der Veröffentlichung des neuen USB Rubber Ducky im Jahr 2022 wurde DuckyScript 3.0 eingeführt. DuckyScript 3.0 ist eine funktionsreiche, strukturierte Programmiersprache. Sie enthält alle bisher verfügbaren Befehle und Funktionen des ursprünglichen DuckyScript, ergänzt um Kontrollflusskonstrukte (if/then/else), Wiederholungen (while-Schleifen), Funktionen und Erweiterungen. Außerdem enthält DuckyScript 3.0 viele Funktionen, die speziell für Keystroke-Injection-Angriffe/Automatisierung geeignet sind, wie z.B. HID- und Storage-Angriffsmodi, Keystroke Reflection, Jitter und Randomisierung.[19]

[19] https://shop.hak5.org/pages/duckyscript-3-0.

Ein klassisches Beispiel für DuckSkript-Code sieht wie folgt aus:

```
REM Example Numeric and Punctuation Keystroke Injection
ATTACKMODE HID STORAGE
DELAY 2000
STRING 1+1=2
```

Bild 4: Beispiel für DuckySkriptCode

Die erste Zeile des Beispiels enthält einen Kommentar. Im Gegensatz zu vielen anderen Programmiersprachen werden Kommentare nicht mit „/“, „*“ oder „#“ gekennzeichnet, sondern beginnen stets mit einem „REM“. Die zweite Zeile legt drei verschiedene Einstellungen fest. Zuerst wird festgelegt, dass der USB Rubber Ducky sich im Angriffsmodus befindet (ATTACKMODE). Das Kürzel HID bestimmt, dass der USB Rubber Ducky als Tastatur erkannt wird. Das Kürzel STORAGE sorgt dafür, dass der USB Rubber Ducky zusätzlich als Massenspeichergerät angezeigt wird. Die dritte Zeile sorgt mit dem Befehl DELAY für eine Verzögerung von 2000 Millisekunden. Die letzte Zeile sorgt schließlich dafür, dass die Zeichenfolgen 1+1=2 übermittelt werden. Wichtig für das Verständnis ist hierbei, dass durch die Kennzeichnung STRING keine Berechnung vorgenommen wird, sondern lediglich die Zeichenfolge übermittelt wird.

3 Überblick über frei erhältliche Hotplug-Attack-Tools

Die Nachfrage nach Hacking-Tools hat in den letzten Jahren stark zugenommen. Bei neuen Tools, wie dem Flipper Zero[20], überstieg die Nachfrage das Angebot deutlich. So wurde auf der Crowdfunding-Plattform Kickstarter der geplante Zielbetrag von 60.000 Dollar um das 81-fache übertroffen.[21] Vor diesem Hintergrund verwundert es nicht, dass immer mehr Anbieter am Markt aktiv sind. Nachfolgend wird ein Überblick der gängigsten Anbietern von Hotplug-Attack-Tools und deren Produkten geboten:

- DSTIKE[22]: Auf der Website von DSTIKE werden verschiedene Hacking-Tools angeboten. Dazu zählt beispielsweise das Tool WiFi Duck[23], ein Open-Source-Hotplug-Attack-Tool. Wifi Duck wurde zum Testen von Key-Injection-Angriffen und zum Erlernen von BadUSBs entwickelt. Das Tool ist so konzipiert, dass es einfach selbst hergestellt werden kann.
- Flipper Devices Inc.[24]: Das Team hinter dem Hacking-Gadget Flipper Zero wurde ursprünglich im Rahmen einer Kooperation von Neuron Hackspace[25] mit Design Heroes[26] gegründet. Die mittlerweile in Philadelphia, USA ansässige Firma vertreibt mit dem Flipper Zero ein tragbares Multitool für Pentester und Geeks in einem spielzeugähnlichen Gehäuse. Der Flipper Zero basiert vollständig auf Open Source und ist

20 https://flipperzero.one/

21 https://www.kickstarter.com/projects/flipper-devices/flipper-zero-tamagochi-for-hackers?lang=de.

22 https://dstike.com/; Über die Identität von DSTIKE werden auf der Homepage keine Angaben gemacht. Auch eine who.is-Abfrage brachte keine brauchbaren Informationen.

23 https://wifiduck.com/docs/about.

24 https://flipperdevices.com/.

25 https://neuronspace.ru/.

26 https://designheroes.io/.

nach Belieben erweiterbar.[27]

- Hak5 LLC[28]: Die Firma Hak5 aus Dallas, Texas, USA wurde im Jahr 2005 gegründet und bietet hochspezialisierte Penetrationstest-Tools an. Die Tools konzentrieren sich überwiegend auf verdeckte Implantation, Manipulation und Datenexfiltration. Zu den bekanntesten Tools gehören der USB Rubber Ducky[29], der Bash Bunny[30] oder der SharkJack[31]. Hak5 bietet zudem fachspezifische Podcasts an und unterhält eine große Community.
- Great Scott Gadgets[32]: Die Firma Great Scott Gadgets aus Lakewood, Colorado, USA, wurde 2009 durch den heutigen CTO Michael Ossman gegründet. Great Scott Gadgets bietet verschiedene Hacking-Tools an. Diese basieren ausschließlich auf Open Source. Im Zusammenhang mit Hotplug-Attacks ist das Tool Cynthion zu nennen. Hierbei handelt es sich um ein All-in-One-Tool zum Erstellen, Testen, Überwachen und Experimentieren mit USB-Geräten. Die digitale Hardware von Cynthion basiert auf einer einzigartigen FPGA-basierten Architektur und kann vollständig an die jeweilige Anwendung angepasst werden.[33]
- Mischief Gadgets LLC[34]: Die Firma Mischief Gadgets, aus San Francisco, Kalifornien, USA stellt unter der Marke O.MG[35] verschiedene Hacking-

27 https://flipperzero.one/.

28 https://shop.hak5.org/.

29 https://shop.hak5.org/collections/hotplug-attack-tools/products/usb-rubber-ducky.

30 https://shop.hak5.org/collections/hotplug-attack-tools/products/bash-bunny.

31 https://shop.hak5.org/collections/hotplug-attack-tools/products/shark-jack.

32 https://greatscottgadgets.com/.

33 https://greatscottgadgets.com/cynthion/.

34 https://mischiefgadgets.com/.

35 https://o.mg.lol/.

Tools her. Hierunter findet sich beispielsweise ein Ladekabel für Handys und Tabletts, das Hotplug-Attacks ausführen kann.

- Retia.io[36]: Das Team hinter Retia.io wurde von dem Informatik-Studenten Kody K. gegründet und ist seit 2019 aktiv. Neben Schulungskursen und Bildungsveranstaltungen bietet Retia.io auch den USB-Nugget an. Hierbei handelt es sich um eine Mikrocontroller-Plattform mit Bildschirm, Tasten und WLAN-Unterstützung, über die Payloads ausgeführt werden können.[37]

[36] https://retia.io/.

[37] https://retia.io/products/wi-fi-nugget-s2-nugget-esp32s2.

4 Technische Merkmale ausgewählter Tools

Nachfolgend werden die technischen Merkmale gängiger Hotplug-Attack-Tools dargestellt. Die Tools wurden dabei mit dem Ziel ausgewählt, ein möglichst breites Spektrum von Einsatzzwecken und Anschlussmöglichkeiten abzudecken. Bei der nachfolgenden Darstellung ist zu beachten, dass die Hersteller dieser Geräte in der Regel keine oder nur wenige Informationen über technische Merkmale veröffentlichen. Oftmals stammen die Informationen aus Sekundärquellen bzw. sind für einzelne Tools gar keine validen Aussagen zu den technischen Merkmalen möglich. Sofern keine verlässlichen Informationen zu den technischen Merkmalen einzelner Tools vorliegen und die Tools nicht einfach zerlegt werden können, wurden die fehlenden Informationen entsprechend gekennzeichnet.

4.1 USB Rubber Ducky (USB-A/C)

Der von der Firma Hak5 entwickelte USB Rubber Ducky stellt den Quasi-standard für USB-Angriffe aller Art dar.[38] Der USB Rubber Ducky wurde in seiner ursprünglichen Form erstmals im Jahr 2010 angeboten und im Jahr 2022 neu aufgelegt. Der USB Rubber Ducky ist eine programmierte Tastatur in Form eines USB-Sticks zur Umsetzung von Keystroke-Injections.[39]

Eng verbunden mit dem USB Rubber Ducky ist die in Abschnitt 1.4 vorgestellte Programmiersprache DuckySkript. Auch wenn DuckySkript mittlerweile von einer Vielzahl von Gadgets verwendet wird, wurde die Programmiersprache ursprünglich für den USB Rubber Ducky entwickelt.[40]

---

[38] Kofler, 2023, S 370.

[39] Amberg / Schmid, 2022, S. 784.

[40] https://docs.hak5.org/hak5-usb-rubber-ducky/.

Die nachfolgende Collage zeigt den USB Rubber Ducky aus dem Jahr 2022.

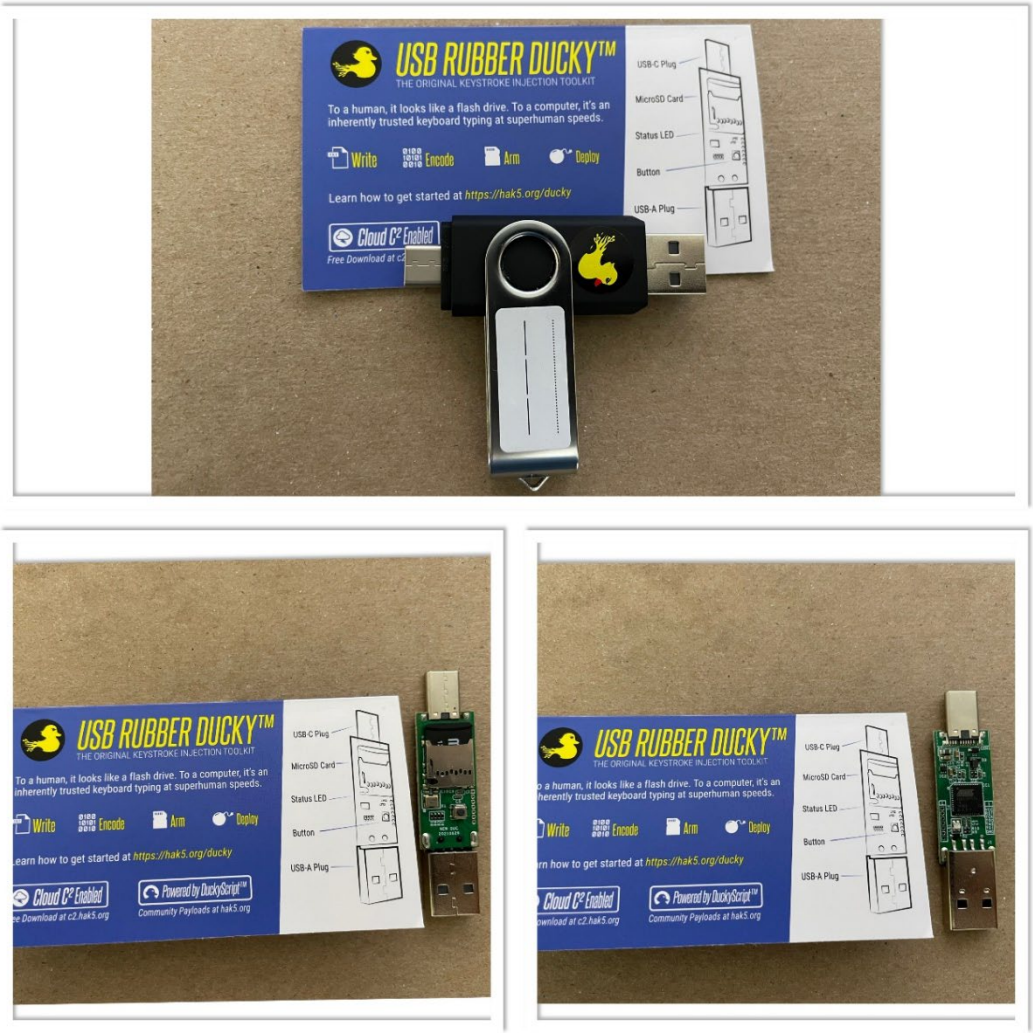


Bild 5: USB Rubber Ducky

Der USB Rubber Ducky in der Form aus dem Jahr 2022 besitzt folgende technische Merkmale:

Prozessor:	Atmel 32bit AVR Mikrocontroller AT32UC3B1256
Interner Speicher:	16 bis 32 MB Flashspeicher
Schnittstellen:	1x USB-C
	1x USB-A
	1x MicroSD

	1x JTAG Schnittstelle
Anzeige:	1x RGB-LED
Bedienelemente:	1x programmierbarer Druckknopf
Stromversorgung	5 Volt über USB-Anschluss (keine separate Stromquelle notwendig)
Betriebsssystem	Entfällt

Tabelle 2: Technsiche Daten USB Rubber Ducky

Der grundsätzliche Aufbau des USB Rubber Ducky kann der nachfolgenden Illustration entnommen werden:

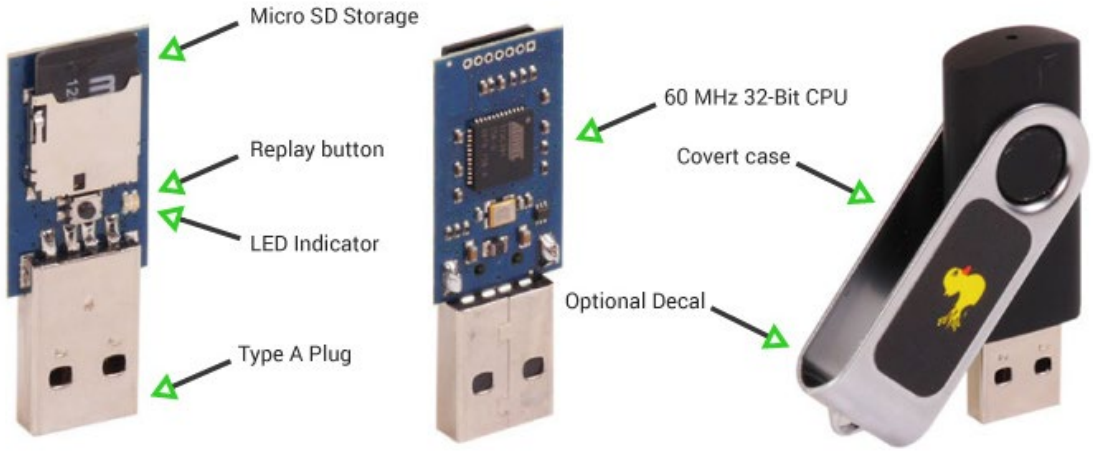


Bild 6: USB Rubber Ducky Aufbau[41]

4.2 Bash Bunny Mark II (USB-A)

Der ebenfalls durch die Firma Hak5 im Jahr 2017 vorgestellte Bash Bunny kann als konsequente Weiterentwicklung des USB Rubber Ducky betrachtet werden. Im Vergleich zum USB Rubber Ducky ist der Bash Bunny nicht auf Keystroke-Injections begrenzt, sondern kann auch als USB-Massenspeichergerät, Gigabit-

[41] Bildquelle: Hak5.

Ethernet-Adapter oder serielle Schnittstelle fungieren. Hierdurch lassen sich eine Vielzahl an weiteren Angriffsmöglichkeiten erschließen. Im Juli 2021 erhielt der Bash Bunny mit dem Bash Bunny Mark II ein Upgrade. Die neue Version erhielt mehr Speicher, ein microSD-XC-Laufwerk mit einer maximalen Kapazität von 2 TB sowie eine Bluetooth LE-Schnittstelle.[42]

Die nachfolgende Collage zeigt den Bash Bunny Mark II aus dem Jahr 2021.

Bild 7: Bash Bunny Mark II

Der Bash Bunny Mark II besitzt folgende technische Merkmale:

[42] Kofler: Hacking & Security: Das umfassende Handbuch, S 385.

Prozessor:	Quad-Core ARM Cortex A7
Interner Speicher:	512 MB DDR3 RAM
	16 MB on-board Flash-Speicher
	8 GB SLC NAND-Speicher
Schnittstellen:	1x USB-A
	1x Bluetooth LE
	1x MicroSD XC
Anzeige:	1x LED
Bedienelemente:	1x 3-Positionen-Schalter
Stromversorgung	Über USB-Anschluss (keine separate Stromquelle notwendig)
Betriebssystem:	Linux Debian

Tabelle 3: Technsiche Daten Bash Bunny Mark II

Der grundsätzliche Aufbau des Bash Bunny Mark II kann der nachfolgenden Illustration entnommen werden:

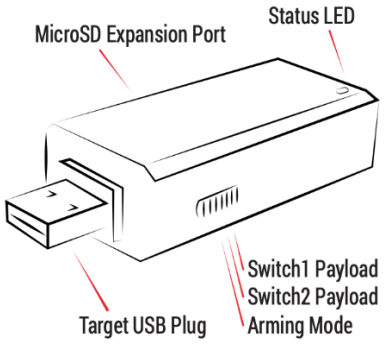


Bild 8: Bash Bunny Mark II Aufbau[43]

[43] Bildquelle: Hak5.

4.3 O.MG Cable Elite (USB-C / Lightning)

Ein Gadget, das in der jüngsten Vergangenheit für Aufsehen gesorgt hat, ist das O.MG-Cable. Das USB-Kabel wurde im Jahr 2019 vom IT-Sicherheitsforscher Mike Grover entwickelt und ist seit dem Jahr 2020 käuflich erhältlich. Beim O.MG-Cable handelt es sich um ein handgefertigtes USB-Kabel mit einem Funk-Implantat zur Fernsteuerung. Das USB-Kabel ist mit verschiedenen Anschlüssen erhältlich.

Das O.MG-Cabel ähnelt handelsüblichen Handy-Ladekabeln bis ins Detail. Im Jahr 2023 erhielt das O.MG-Cabel ein Upgrade und ist nun in einer Basis- und einer Elite-Version erhältlich.

Das nachfolgende Bild zeigt das O.MG-Cabel Elite aus dem Jahr 2023.

Bild 9: O.MG Cable Elite

Das O.MG-Cable besitzt folgende technische Merkmale:

Prozessor:	ESP8266-Modul (system on a chip)* * Nicht durch verlässliche Quellen verifizierbar
Interner Speicher:	Keine Angabe
Schnittstellen:	Wifi 802.11b/g/n (2,4GHz)
	1x USB Typ-C

	1x Lightning
Anzeige:	Entfällt
Bedienelemente:	Entfällt
Stromversorgung	5 Volt über USB-Anschluss (keine separate Stromquelle notwendig)
Betriebssystem:	Keine Angabe

Tabelle 4: Technische Daten O.MG-Cable

Das nachfolgende Bild zeigt eine Röntgen-Aufnahme eines O.MG-Cables.

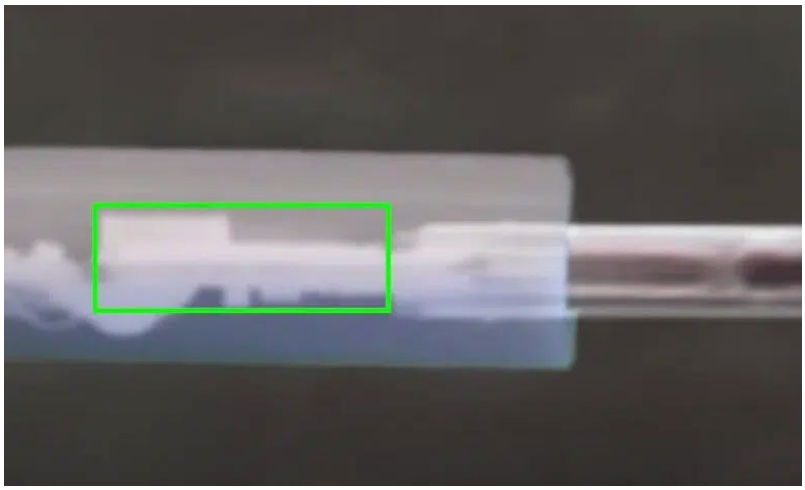


Bild 10: Röntgen-Aufnahme eines O.MG Cables[44]

4.4 O.MG Plug Elite (USB-A)

Vom selben Anbieter wie das O.MG-Cable gibt es auch den O.MG Plug. Technik und Funktionsumfang des O.MG-Plug sind mit dem O.MG-Cable weitgehend identisch, lediglich das Format und damit der Einsatzzweck unterscheidet sich.

Während das O.MG-Cable hauptsächlich für Attacken auf Smartphones gedacht ist, eignet sich der O.MG-Plug für Angriffe auf Laptops und PCs.

[44] Bildquelle: Mike Grover

Das nachfolgende Bild zeigt den O.MG-Plug Elite aus dem Jahr 2023.

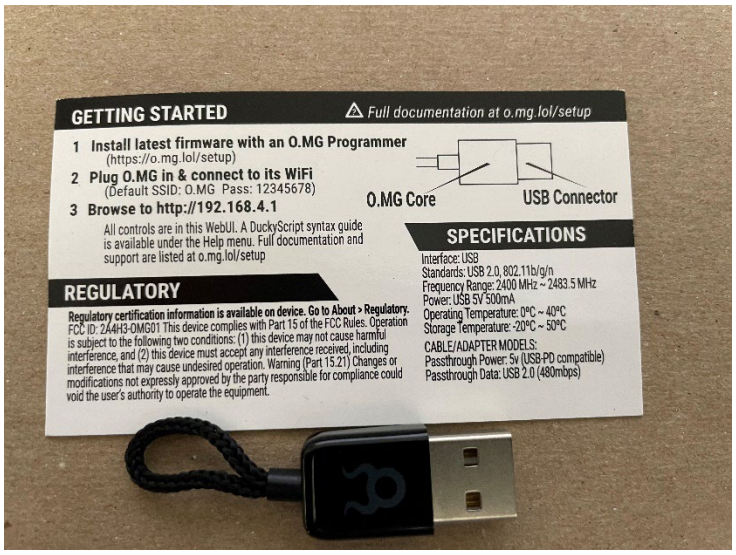


Bild 11: O.MG Plug Elite[45]

Der O.MG-Plug Elite besitzt folgende technische Merkmale:

Prozessor:	Keine Angabe
Interner Speicher:	Keine Angabe
Schnittstellen:	Wifi 802.11b/g/n (2,4GHz)
	1x USB Typ-A
Anzeige:	Entfällt
Bedienelemente:	Entfällt
Stromversorgung	5 Volt über USB-Anschluss (keine separate Stromquelle notwendig)
Betriebssystem:	Keine Angabe

Tabelle 5: Technsiche Daten O.MG Plug Elite

[45] Bildquelle: Hak5.

4.5 Shark Jack (Ethernet)

Der im Jahr 2019 von Hak5 präsentierte Shark Jack wurde im Vergleich zu den zuvor vorgestellten Tools für einen spezielleren Einsatzzweck konzipiert. Zielsetzung des Shark Jack sind kurze Aufklärungs-, Exfiltrations- und IT-Automatisierungsaufgaben im Zielnetzwerk. Der Shark Jack wird standardmäßig mit einem vorinstallierten Programm ausgeliefert, die einen nmap-Scan ausführt und die Scanergebnisse in einem Verzeichnis auf dem Gerät ablegt.[46]

Die nachfolgende Collage zeigt den Shark Jack aus dem Jahr 2023:

Bild 12: Shark Jack

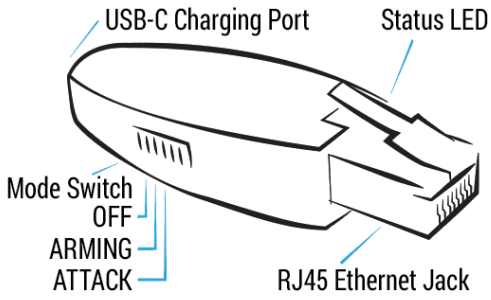
[46] https://docs.hak5.org/shark-jack/getting-started/shark-jack-basics.

Der Shark Jack besitzt folgende technische Merkmale:

Prozessor:	580-MHz-MIPS-Prozessor
Interner Speicher:	64 MB RAM
	64 MB on-board Flash-Speicher
Schnittstellen:	1x USB-C
	1x RJ45 Ethernet-Stecker
Anzeige:	1x LED
Bedienelemente:	1x 3-Positionen-Schalter
Stromversorgung	Über USB-Anschluss (keine separate Stromquelle notwendig)
Betriebssystem:	Router-OS OpenWrt

Tabelle 6: Technische Daten Shark Jack

Der grundsätzliche Aufbau des Shark Jack kann der nachfolgenden Illustration entnommen werden:



[47]

Bild 13: Shark Jack Aufbau

[47] Bildquelle: Hak5..

4.6 FlipperZero (USB-A/C + Bluetooth)

Der im Jahr 2022 erschienene Flipper Zero stellt das mit Abstand umfassendste, der im Rahmen dieser Arbeit vorgestellten Gadgets dar. Der Flipper Zero ist ein tragbares Mulitool für Pentester, mit einem Tamagotchi ähnlichen Design. Neben den Möglichkeiten für Hotplug-Attacks bedient der Flipper Zero eine Reihe weiterer möglicher Angriffsvektoren. So spielt der Flipper Zero insbesondere in Zusammenhang mit Funkprotokollen und Zugriffskontrollsystemen seine Stärken aus.

Die nachfolgende Collage zeigt den Flipper Zero.

Bild 14: Flipper Zero

Der Flipper Zero besitzt folgende technische Merkmale:

Prozessor:	ARM Cortex-M4 32-bit 64 MHz (application processor)
	ARM Cortex-M0+ 32 MHz (network processor)
Interner Speicher:	Flash: 1024 KB
	SRAM: 256 KB
Schnittstellen:	1x USB 2.0 port, type C
	1x Bluetooth LE 5.0
	1x RFID Board
	1x iButtonPad
	1x MicroSD
	1x Infrared-Revicer
	1x Sub-1GHz Antenne
Anzeige:	LCD Monchrome Display
Bedienelemente:	Directional Pad & Back Button
Stromversorgung	5 Volt über USB-C
Betriebssystem:	Eigen entwickeltes Betriebssystem

Tabelle 7: Technsiche Daten Flipper Zero

Der Aufbau des Flipper Zero kann der nachfolgenden Illustration entnommen werden:

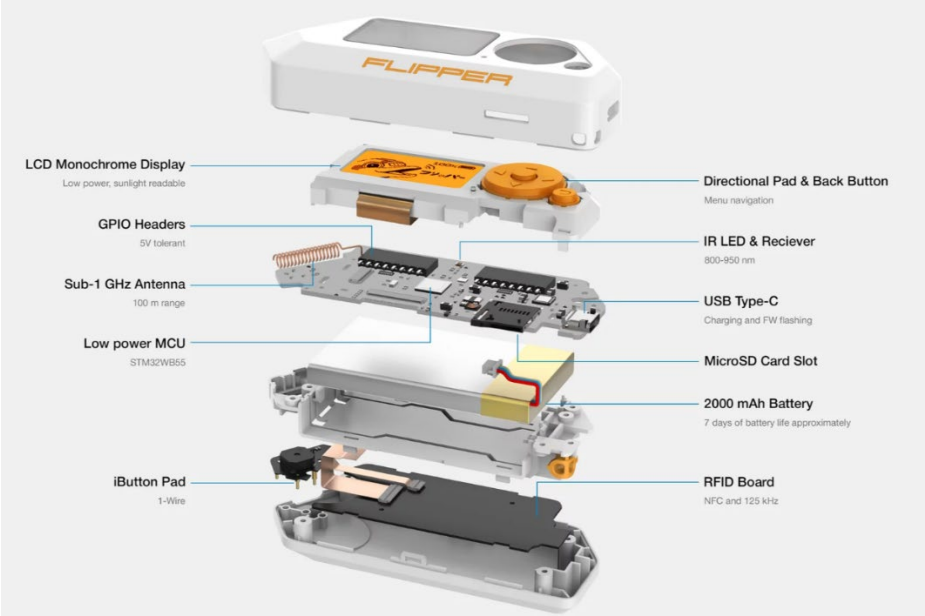


Bild 15: Flipper Zero Aufbau

[48]

4.7 HackyPi (USB-A)

Das mit Abstand günstigste in dieser Arbeit vorgestellte Tool ist der Anfang 2023 erschiene HackyPi von sb Components. Der HackyPi wurde als Lern- und Bildungswerkzeug für angehende ethische Hacker und Programmierer entwickelt. Der HackyPi ist kompatibel mit Windows, Mac und Linux, benötigt keine Treiber und wird mit Python-Unterstützung geliefert. Die Hardware-Komponenten sind vollständig Open-Source.

[48] Bildquelle: https://flipperzero.one/.

Die nachfolgende Collage zeigt den HackyPi.

Bild 16: HackyPi

Der HackyPi besitzt folgende technische Merkmale:

Prozessor:	RP2040 Dual-core Arm Cortex-M0+ processor
Interner Speicher:	QSPI flash
Schnittstellen:	1x USB-Typ A
	1x MicroSD
Anzeige:	TFT 1.14” display

Bedienelemente:	Boot button
Stromversorgung	5 Volt über USB-Anschluss (keine separate Stromquelle notwendig
Betriebssystem:	Keine Angabe

Der Aufbau des HackyPi kann der nachfolgenden Illustration entnommen werden:

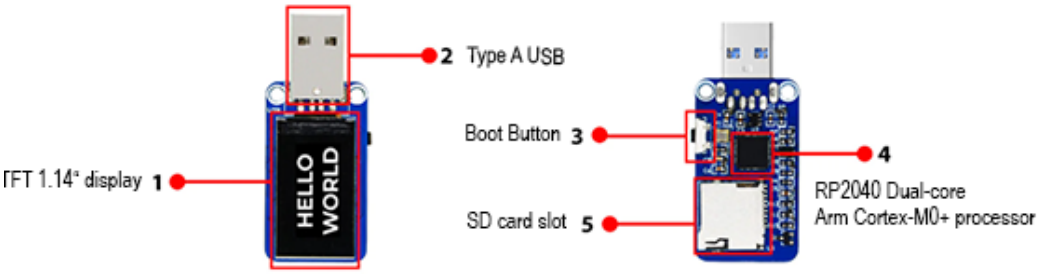


Bild 17: HackiPy Aufbau

[49]

[49] https://shop.sb-components.co.uk/products/hackypi-compact-diy-usb-hacking-tool.

5 Vergleich ausgewählter Tools anhand bestimmter Kriterien

Nachfolgend werden die in Kapitel 4 vorgestellten Tools anhand verschiedener, aus Sicht eines Angreifers relevanter Kriterien miteinander verglichen. Für die Beurteilung wird dabei unterstellt, dass die Variabilität (größerer Umfang an Einsatzmöglichkeiten) und die Bedienbarkeit (kurze Rüstzeiten, höhere Effizienz) wichtiger sind, als eine größtmögliche Präzision. Dies ist dem Umstand geschuldet, dass Hotplug-Attacks oftmals in einem Zustand unvollständiger Information ablaufen. Werden beispielsweise USB-Sticks auf einem Parkplatz zurückgelassen, in der Hoffnung, dass diese gefunden und mit einem Computer verbunden werden, ist oftmals nicht sichergestellt, dass der Computer auch die passende Betriebsumgebung bereitstellt. Zudem handeln Angreifer oftmals unter Zeitdruck. Wird ein Angreifer in einem Büro kurz allein gelassen, muss er die Attacke meist innerhalb weniger Augenblicke durchführen können, um unentdeckt zu bleiben.

Dieser These folgend, wurden die Tools anhand der folgenden Kriterien verglichen:

- möglicher Payload: Je variabler der mögliche Payload gestaltet wird, desto mehr Angriffsmöglichkeiten bestehen. Zudem erhöht eine Open-Source-Sammlung von möglichen Payloads die Effizienz.
- Zugriffs- und Steuerungsmöglichkeiten: Durch Möglichkeiten, mit den Geräten während eines Angriffs zu interagieren, sie fernzusteuern und deren Einsatzgebiet lokal einzuschränken, werden Angriffe flexibler und es steigen die Chancen unentdeckt zu bleiben.
- Möglichkeiten zur Exfiltration von Daten: ein zentraler Aspekt vieler Angriffe ist die Möglichkeit, Daten unentdeckt aus Systemen zu exfiltrieren. Hierzu müssen Daten gegebenenfalls zwischengespeichert werden.
- Schutz vor Entdeckung: Durch verschiedene Techniken können Angriffe maskiert werden. Hierdurch lässt sich der Schutz vor Entdeckung erhöhen.
- Verhalten der Tools im praktischen Einsatz: Da Hotplug-Attacks in der

Regel von verschiedenen äußeren Bedingungen abhängig sind, ist es notwendig, dass diese im praktischen Einsatz verlässlich funktionieren und gängige Payloads wie vorgesehen ablaufen.

5.1 Möglicher Payload

Nachfolgend wird der mögliche Payload und dessen Entwicklung für die jeweiligen Hotplug-Attack-Tools dargestellt.

Die Spalte Payload-Slots zeigt auf, wie viele Payloads auf dem Geräte einsatzfähig bereitgehalten werden können. Beispielsweise kann beim Bash Bunny Mark II über den Schiebeschalter auf zwei Payloads zugegriffen werden, während beim O.MG-Cable Elite bis zu 200 Payloads abrufbar bereit stehen.

Die Spalte Dokumentation gibt an, ob eine offizielle Dokumentation zum jeweiligen Tool vorliegt.

Die Spalte Entwicklungsumgebung listet die möglichen Entwicklungsumgebungen für eigenentwickelten Payload auf.

In der Spalte Verfügbare Payloads finden sich Informationen darüber, ob Open Source-Payloads vorhanden ist.

Tool	**Payload-Slots**	**Dokumentation**	**Entwicklungs-umgebung**	**Verfügbare Payloads**
USB Rubber Ducky	1	Schulungsvideos und Dokumentation von Hak5	Texteditor oder Entwicklungs-umgebung von Hak5	Vorgefertigte Payloads über Payload Hub von Hak5 und zahlreiche GitHub-Archive
Bash Bunny Mark II	2	Dokumentation von Hak5	Texteditor oder Entwicklungs-umgebung von Hak5	Vorgefertigte Payloads über Payload Hub von Hak5 und zahlreiche GitHub-Archive

O.MG Cable Elite	50-200	Website o.mg.lol	Integrierte IDE	Vorgefertigte Payloads über Payload Hub von Hak5 und zahlreiche GitHub-Archive
O.MG Plug Elite	50-200	Website o.mg.lol	Integrierte IDE	Vorgefertigte Payloads über Payload Hub von Hak5 und zahlreiche GitHub-Archive
Shark Jack	1	Dokumentation von Hak5	Texteditor oder Entwicklungs-umgebung von Hak5	Vorgefertigte Payloads über Payload Hub von Hak5 und zahlreiche GitHub-Archive
Flipper Zero	unbegrenzt	Website docs.flipper.net	Entwicklungs-umgebung für C	Eigener App-Store, umfangreiche Git-Hub-Archive
HackyPi	1	Dokumentation über github.com	Python-Entwicklungs-umgebung Thony	Wenige Git-Hub-Archive

Tabelle 8: Möglicher Payload

Hinsichtlich der Dokumentation, der Entwicklungsumgebung sowie der verfügbaren Payloads fällt lediglich der HackyPi ein wenig gegenüber den übrigen Tools ab. Die größte Flexibilität bezüglich des Payloads bieten die beiden O.MG-Produkte sowie der Flipper Zero. Einen Spezialfall stellt der Shark Jack dar, dessen Anwendungsbereich im Vergleich zu den übrigen Tools deutlich begrenzter ist.

5.2 Zugriffs- und Steuerungsmöglichkeiten

Die folgende Tabelle gibt einen Überblick über die verschiedenen Zugriffs- und Steuerungsmöglichkeiten. Dabei werden die Bereiche Interaktivität, d.h. die Eingriffsmöglichkeiten bei der Ausführung des Payloads, Remote-Controll, d.h. die Möglichkeiten zur Fernsteuerung und Geo-Fencing, d.h. die Beschränkung der Payload-Ausführung auf bestimmte Gebiete, näher betrachtet.

Tool	**Interaktivität**	**Remote Control**	**Geo-Fencing**
USB Rubber Ducky	Versteckter Button	Nein	Nein
Bash Bunny Mark II	3-Positionen-Schalter	Fernauslösen von Payload über Bluetooth möglich	Ja
O.MG Cable Elite	Wifii	Über Wifi	Ja
O.MG Plug Elite	Wifi	Über Wifi	Ja
Shark Jack	2-Positionen-Schalter	Über Netzwerk und Internet	Nein
Flipper Zero	Directional Pad & Back Button	Nein	Nein
HackyPi	Button	Nein	Nein

Tabelle 9: Zugriffs- und Steuerungsmöglichkeiten

Hinsichtlich der Zugriffs- und Steuerungsmöglichkeiten unterscheiden sich die einzelnen Tools teilweise deutlich. Dies ist teilweise auf den Einsatzzweck zurückzuführen. Während die O.MG-Tools für den unentdeckten Fernzugriff konzipiert wurden, sind die übrigen Tools überwiegend für den unmittelbaren Einsatz vor Ort durch einen Angreifer ausgelegt. Das aus Sicht eines Angreifers

umfassendste Tool ist sicherlich der Bash Buny Mark II. Über den 3-Positionen-Schalter in Kombination mit der Status LED lassen sich Angriffe vor Ort steuern, durch die Bluetooth-Verbindung kann Payload aus einiger Entfernung ausgelöst werden und die Geo-Fencing-Funktion verhindert, dass der Payload an einem falschen Ort ausgeführt wird.

5.3 Exfiltration von Daten

Eine wichtige Funktionalität von Hotplug-Attack-Tools ist die Möglichkeit, Daten aus Systemen zu exfiltrieren. Hierzu gibt es grundsätzlich zwei Möglichkeiten[50]. Entweder können die Daten auf dem Gerät gespeichert werden und das Gerät wird anschließen wieder eingesammelt oder das Tool ermöglicht anderweitig die Übertragung der Daten. Die folgende Tabelle vergleicht die Tools hinsichtlich Speichermöglichkeiten und Übertragungswegen.

Tool	**Speichermöglichkeiten**	**Übertragungswege**
USB Rubber Ducky	SD-Karte	--
Bash Bunny Mark II	SD-Karte	Bluetooth
O.MG Cable Elite	Nein	Wifi
O.MG Plug Elite	Nein	Wifi
Shark Jack	Auf internen Speicher bregrenzt	Netzwerk und Internet
Flipper Zero	SD-Karte	Bluetooth, Wifi
HackyPi	SD-Karte	--

Tabelle 10: Exfiltration von Daten

[50] Eine dritte Möglichkeit besteht darin, die Daten im angegriffenen System zwischenzuspeichern und z.B. über eine Internetverbindung zu übertragen.

Die untersuchten Hotplug-Attack-Tools lassen sich grob in zwei Gruppen unterscheiden, Geräte welche die Informationen auf einer SD-Karte und/oder internen Speicher sichern und nach Verwendung wieder eingesammelt werden müssen (insbesondere der USB Rubber Ducky) und Geräte die für eine direkte Übertragung der Daten (O.MG-Produkte) ausgelegt sind. Die umfassendsten Funktionen bietet der Flipper Zero.

5.4 Schutz vor Entdeckung

Viele Angriffe hängen davon ab, dass die Angriffe unentdeckt bleiben. Hierfür gibt es verschiedene Techniken wie die Tarnung von Payloads, Stealthing (das Gerät bleibt für das Opfer unsichtbar), oder Selbstzerstörungsfunktionen.

Tool	**Tarnung von Payloads**	**Stealthing**	**Selbstzerstörung**
USB Rubber Ducky	Ja	Nein	Programmierbar
Bash Bunny Mark II	Ja	Nein	Programmierbar
O.MG Cable Elite	Nein	Ja	Ja
O.MG Plug Elite	Nein	Port Stealthing	Ja
Shark Jack	Nein	Nein	Nein
Flipper Zero	Nein	Nein	Nein
HackyPi	Ja	Nein	Nein

Tabelle 11: Schutz vor Entdeckung

Der Schutz der Geräte vor Entdeckung variiert abhängig vom Einsatzzweck. Insbesondere Geräte die für Schulungszwecke entwickelt wurden (Flipper Zero und HackyPi) weisen keinen oder nur geringen Schutz vor Entdeckung auf. Ausgeprägt ist hingegen der Schutz vor Entdeckung insbesondere bei den O.MG-Geräten, die unter der Prämisse der Tarnung entwickelt wurden.

5.5 Praktischer Einsatz

Final wurden die vorgestellten Geräte nochmals im praktischen Einsatz getestet. Da die Geräte teilweise für unterschiedliche Einsatzzwecke konzipiert wurden, wurde der Payload gerätespezifisch festgelegt. Hier wurde versucht, Geräte mit dem gleichem Einsatzzweck mit möglichst identischem Payload zu testen.

Als Testumgebung kommt ein Lenovo Thinkpad T16 mit Windows 11 zum Einsatz. Standardmäßig wird ein deutsches Tastaturlayout verwendet. Auf dem Laptop ist u.a. Outlook lokal installiert.

Das Testsystem ist in ein Heimnetzwerk eingebunden. Als Router fungiert eine Fritzbox 7590.

5.5.1 USB Rubber Ducky

Für den USB Rubber Ducky wurde als Szenario festgelegt, dass der Angreifer Zugriff auf einen Computer mit Outlook als Mailprogramm bekommt. Von diesem Computer aus will er eine Phishing-Mail an einen anderen Kollegen verschicken.

Der Code für das Beispiel wurde im Hak5-Payload-Studio verfasst. Die Entwicklungsumgebung prüft die Syntax des Codes automatisch, erstellt eine inject.bin-Datei, welche per drag&drop auf den USB Rubber Ducky gezogen werden kann. Der USB Rubber Ducky ist anschließend sofort einsatzfähig.

Der nachfolgend dargestellte Code wurde für Illustrationszwecke bewusst einfach gehalten. Als Standardtastaturformat wurde de gewählt. Für Rekonstruktionszwecke muss die Zeichenkette „INSERT_TARGET_MAIL_ ADRESS_HERE“ durch die Mailadresse des Opfers ausgetauscht werden.

```
REM ###############################################################################################
REM #                                                                                             |
REM # Title         : Starting Outlook via Powersehll and send a Mail                             |
REM # Author        : F. Winterer                                                                 |
REM # Version       : 1.0                                                                         |
REM # Category      : Execute                                                                     |
REM # Target        : Windows 11                                                                  |
REM #                                                                                             |
REM ###############################################################################################

REM Requirements:
REM     - Nothing

DELAY 1000
GUI x
DELAY 500
STRING a
DELAY 500
LEFTARROW
DELAY 500
ENTER

DELAY 2000
STRINGLN Start-Process 'C:\Program Files\Microsoft Office\root\Office16\OUTLOOK.EXE'
DELAY 500
CTRL n
DELAY 500
TAB
DELAY 500
STRING INSERT_TARGET_MAIL_ADRESS_HERE
TAB
DELAY 500
TAB
DELAY 500
STRINGLN Here is a litte phishing mail
TAB
DELAY 500
STRINGLN Hello,
ENTER
ENTER
STRINGLN this is a phishing mail.
CTRL ENTER
DELAY 500

ALT F4
```

Bild 18: Beispielcode für USB Rubber Ducky

Im Ergebnis arbeitet der USB Rubber Ducky die Befehle ordnungsgemäß ab und verschickt eine E-Mail über Outlook. Der USB Rubber Ducky erscheint dabei nicht als USB-Stick. Lediglich ein kurzes akustisches Signal deutet darauf hin, dass eine neue Hardware angeschlossen wurde.

Soll auf den USB Rubber Ducky zugegriffen werden, genügt ein einfaches Drücken des versteckten Knopfs und der USB Rubber Ducky wird als USB-Stick angezeigt.

5.5.2 Bash Bunny Mark II

Für den Bash Bunny Mark II wurde für den direkten Vergleich dasselbe Szenario wie für den USB Rubber Ducky verwendet. Hier müssen im Vergleich zum USB Rubber Ducky einige zusätzliche Einstellungen geändert werden. Der Payload wurde ebenfalls über das Payload-Studio von Hak5 erstellt. Es handelt sich hier allerdings nicht um eine .bin-Datei sondern um eine .txt-Datei, die in jedem beliebigen Texteditor erstell werden kann.

Zusätzlich wurde für den Bash Bunny Mark II noch ein Standard-Payload von Hak5 (USB-Exfiltration)[51] getestet. Der Payload dient der Exfiltration von Dateien aus dem Ordner „Dokumente“ des Benutzers. Die Speicherung erfolgt im Ordner „loot“ auf der USB-Massenspeicherpartition von Bash Bunny. Die Bezeichnung des Ordners setzt sich aus dem Hostname, dem Datum und einem Zeitstempel zusammen.

Beide Payloads wurden vom Bash Bunny Mark II ordnungsgemäß abgearbeitet. Im Vergleich zum USB Rubber Ducky war die Usability des Bash Bunny Mark II aufwendiger und die Reaktionszeit merklich länger.

5.5.3 O.MG Cable Elite (USB-C / Lightning)

Im Gegensatz zu den meisten hier vorgestellten Tools, sind die O.MG-Produkte nicht sofort einsatzbereit. Für die Programmierung der Gadgets wird ein separates Gerät für die Programmierung benötigt, das zwischen den PC und das jeweilige Tool gesteckt wird. Vor dem ersten Einsatz müssen die O.MG-Produkte zudem erst einmal den WebFlasher durchlaufen.

[51] https://github.com/hak5/bashbunny-payloads/tree/master/payloads/library/exfiltration/usb_exfiltrator.

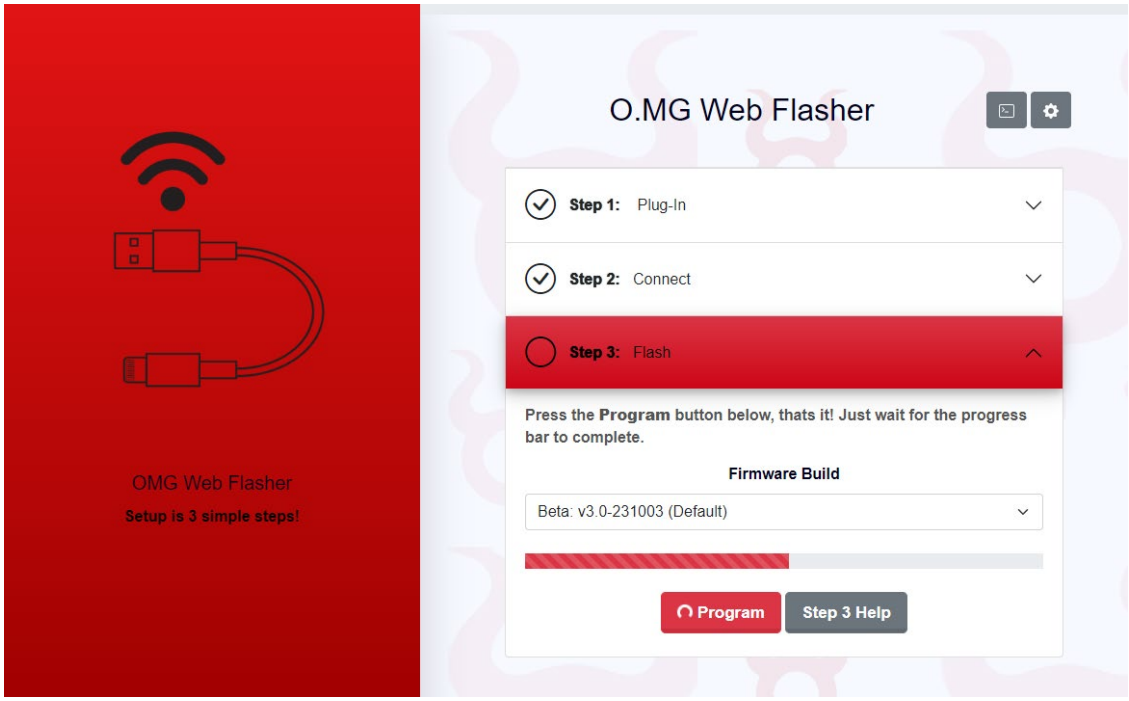


Bild 19: Interface des O.MG-WebFlasher

Für das O.MG Cable wurde das Szenario modifiziert, da das Kabel standardmäßig an ein Handy angeschlossen wird. Aus diesem Grund wurde ein Open Source Code modifiziert, der eine iMessage versenden soll.

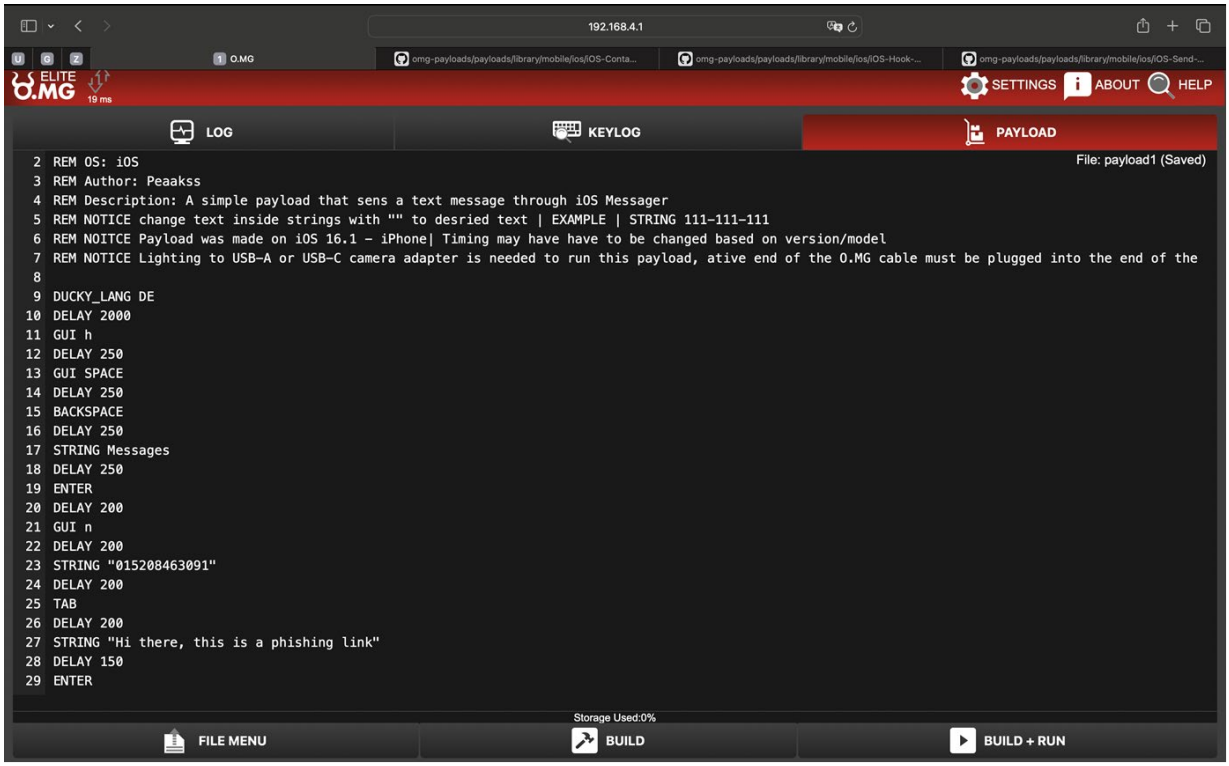


Bild 20: Beispielcode für O.MG Cable

Wichtig beim Einsatz des O.MG Cable ist, dass das USB-Kabel nur über ein aktives Ende verfügt. Lediglich dieses Ende löst den Payload aus. Das aktive Ende ist immer ein USB-Anschluss. Dies führt dazu, dass Payload nie über den Lightning-Anschluss an ein iPhone gegeben werden kann. Mit Umstellung der iPhones auf USB-C ergibt sich somit ein neuer Angriffsvektor.

Im praktischen Einsatz wird der hinterlegte Payload unmittelbar nach Verbindung des Kabels mit dem Gerät ausgeführt. Mit dem oben dargestellten Code konnten iMassages über ein MacBook Pro M1 versendet werden.

Zudem spannt das O.MG Cable ein Wifi-Netz auf. Hierdurch ist ein Fernzugriff auf das Kabel und die Auswahl der Payloads in Echtzeit möglich. Das Kabel ermöglicht zudem Keylogging in Echtzeit.

Mit dem O.MG Cable wurden noch weitere Standard-Payloads getestet. Die Reaktionszeit des O.MG Cable ist sehr schnell. Der vorgegebene Code wird ohne Probleme bearbeitet.

5.5.4 O.MG Plug Elite (USB-A)

Der O.MG Plug ist das kleinste der getesteten Tools. Auch er muss zuerst mit dem Programmer verbunden werden und den WebFlasher durchlaufen.

Die Funktionsweise und Bedienbarkeit des O.MG Plug unterscheidet sich nicht vom O.MG Cable. Die Reaktionszeit des O.MG Plug ist sehr schnell. Der vorgegebene Code wird ohne Probleme bearbeitet.

5.5.5 Shark Jack (Ethernet)

Für den Shark Jack wurde bewusst auf ein eigenes Code-Beispiel verzichtet. Der Shark Jack wird standardmäßig mit einem vorinstallierten nmap-Scanner ausgeliefert. Der Shark Jack muss lediglich zum Aufladen über ein USB-C-Kabel an eine Stomquelle angeschlossen werden. Anhand der verschiedenen LED-Farben kann der jeweilige Status des Shark Jack nachvollzogen werden.

Für den Test wurde der vorinstallierte nmap-Scan benutzt. Im Ergebnis lieferte der Shark Jack einen vollständigen nmap-Scan über das getestete Netzwerk. Der Scan war ordnungsgemäß auf dem Shark Jack gespeichert.

5.5.6 Flipper Zero (USB-A/C + Bluetooth)

Die Aufgabe für den Flipper Zero bestand wie beim USB Rubber Ducky und Bash Bunny Mark II zunächst im Aufruf von Outlook und dem Versand einer E-Mail. Hierfür wurde der bekannte in DuckyScript verfasste Payload verwendet. Der Flipper Zero verarbeitet DuckyScript ohne Probleme. Die Bedienung des Flipper Zeros gestaltet sich einfach und die Reaktionszeiten sind schnell.

Zusätzlich wurde eine im Vergleich zu den übrigen Tools einzigartige Funktion des Flipper Zero getestet. Im Appstore des Flipper Zero ist eine App erhältlich, die den Remote-Zugriff über USB-C oder Bluetooth erlaubt. Der Flipper Zero dient dabei als Tastatur. Die Tasten werden ähnlich wie bei einer Videospielkonsole über das Steuerkreuz ausgewählt. Das Tippen erfordert in diesem Fall etwas Übung, geht aber grundsätzlich ohne Probleme.

5.5.7 HackyPi (USB-A)

Der HackyPi wurde ebenfalls mit dem bereits bekannten Szenario konfrontiert. In diesem Fall konnte der bekannte DuckyScript-Code nicht einfach verwendet werden, da die Programmierung des HackyPi in der Programmiersprache C zu erfolgen hat. Die Programmierung in C ist deutlich weniger intuitiv als DuckyScript. Gleichwohl lässt sich auch in C das Szenario programmieren.

Die Reaktionszeit des HackyPi ist ähnlich der des USB Rubber Ducky sehr schnell. Der vorgegebene Code wird ohne Probleme bearbeitet. Ein im Vergleich zu den übrigen Tools von Hak5 positives Feature ist der Bildschirm, über den Statusmeldungen ausgegeben werden können.

6 Erkennung von Hotplug-Attacks

Die Erkennung von Hotplug-Attacks wird durch die technischen Anforderungen und Restriktionen, die mit Interaktion von USB-Geräten der HID-Klasse einhergehen (siehe Abschnitt 1.2), limitiert. In der Regel ist es für einen Computer nicht möglich, Hotplug-Attack-Tools von anderen erlaubten HID-Geräten zu unterscheiden, soweit die Tools entsprechend konfiguriert sind.

Eine Möglichkeit, Hotplug-Attacks bereits vor der Ausführung zu erkennen, ist der Einsatz eines Whitelistings. Hierbei werden Regeln definiert, die die Nutzung von HID-Geräten auf Geräte mit vorgegebenen Spezifikationen einschränken. Dies kann geschehen, indem beispielsweise nur ein bestimmter Tastaturtyp eines bestimmten Herstellers erlaubt wird. Wird ein abweichend konfiguriertes HID-Gerät erkannt, wird dessen Verwendung unterbunden. Eine fortgeschrittene Form eines solchen Whitelistings beruht auf einer Zugangskontrolle, bei der die Informationen und der Treiber des eigentlichen installierenden Gerätes zur Überprüfung der Firmware verwendet werden.[52]

Problematisch am Whitelisting ist, dass hierdurch nur ein begrenzter Grad an Sicherheit erlangt werden kann. Gerade wenn Angreifern die eingesetzte Hardware bekannt ist, kann das Whitelisting einfach umgangen werden. Aus diesem Grund bedarf es zusätzlicher Maßnahmen.

Standardmäßig wird versucht, auffälliges Verhalten von HID-Geräten zu erkennen. Dies kann einerseits durch fortschrittliche softwarebasierte Erkennungssysteme (siehe Abschnitt 6.3) oder durch Hardwaremaßnahmen geschehen. Ein Beispiel für ein hardwaregestütztes, dynamisches USB-Bedrohungserkennungs-Framework ist USB-Watch. USB-Watch nutzt Hardware, die zwischen einem USB-Gerät und dem Host-Computer platziert wird, um sich in die USB-Kommunikation einzuklinken, USB-Daten zu sammeln und die unveränderte USB-Protokollkommunikation mitzuschneiden. Diese Daten werden anschließend in einen auf maschinellem Lernen basierenden

[52] Lee, et. al., , 2016, S. 378.

Klassifikator eingespeist, der dynamisch die wahre Natur des USB-Geräts bestimmt.[53]

Anzumerken ist, dass auch eine Kombination der verschiedenen Erkennungsmethoden nicht zu einem absoluten Schutz vor Hotplug-Attacks führt. Zum einen weisen selbst modernste Erkennungssysteme keine 100-prozentige Trefferquote auf, zum anderen kann auch ein sehr kurzer Payload bereits beträchtlichen Schaden anrichten. Dadurch kann die Entdeckung eines solchen Angriffs für die Schadensabwehr wertlos sein.

[53] Denney, et. al., 2019, Seite 126.

7 Präventionsmaßnahmen gegen Hotplug-Attacks

Nachdem im vorherigen Abschnitt Maßnahmen zur Erkennung von Hotplug-Attacks vorgestellt wurden, sollen nachfolgend einige Maßnahmen beschrieben werden, wie sich Hotplug-Attacks verhindern bzw. eindämmen lassen. Hierbei wird zwischen organisatorischen Maßnahmen (keine Hardwareveränderung oder zusätzliche Software), Hardwaremaßnahmen und Softwaremaßnahmen unterschieden. Nachfolgend werden je Gruppe einige gängige Maßnahmen vorgestellt.

7.1 Organisatorische Maßnahmen

7.1.1 Automatische Zugriffssperren / Passwort Richtlinien

Viele grundlegende Hotplug-Attacks setzen einen Zugriff auf das System voraus. Hierzu muss in der Regel das angegriffene Gerät entsperrt sein. Durch automatische Zugriffssperren, wie timeout extension, können Hotplug-Attacks deutlich erschwert werden.[54] Zur Vermeidung von unberechtigten Zugriffen durch Attacken auf die Anmeldung bieten sich die Verwendung von starken Passwörtern, die Vermeidung von Mustern und das automatische Zurücksetzen auf die Werkseinstellungen bei wiederholter Passwortfalscheingabe an.[55]

7.1.2 Group Polices

Durch Nutzung von Group Polices können Administratoren die Nutzung von HID-Geräten wirksam eingeschränken bzw. verbieten.

7.1.3 Deaktivierung von USB-Ports

USB-Ports können über das Betriebssystem dauerhaft deaktiviert werden.

---

[54] Potocký /. Stulrajter, 2022, S. 96.

[55] Khande et. al, 2023, Seite 96.

7.2 Hardware-Maßnahmen

7.2.1 Physische Schließung von USB-Ports

Eine sehr wirksame, wenn auch radikale Präventionsmaßnahme gegen Hotplug-Attacks stellt die physische Schließung von USB-Ports durch Versiegelung oder Entfernung der USB-Ports dar. Diese Maßnahme kommt natürlich nur in Betracht, sofern die USB-Ports nicht benötigt werden.

7.2.2 USB-Firewall

Verschiedene Anbieter haben mittlerweile Hardware Firewalls, wie die USB v1.0 Hardware Firewall[56], im Portfolio. Diese Geräte werden zwischen den USB-Port und das anzuschließende USB-Gerät gesteckt und kontrollieren den Datenverkehr. Durch entsprechende Regeln kann auffälliger Datenverkehr entdeckt und unterbunden werden.

7.3 Software-Maßnahmen

7.3.1 Duckhunt

Bei Duckhunt[57] handelt es sich um ein Skript, das die Tastaturnutzung (aktuelle Geschwindigkeit und ausgewählte Fenster) kontinuierlich überwacht, um Keystroke-Injection-Angriffe abzufangen und zu verhindern. Das Programm zeichnet hierzu die Tastaturanschläge des Nutzers auf, um die durchschnittliche Schreibgeschwindigkeit zu protokollieren. Wenn sich die individuelle durchschnittliche Geschwindigkeit signifikant verändert, führt Duckhunt je nach hinterlegtem Modus eine Aktion aus. Insgesamt gibt es vier Modi:

- Paranoid: Wird ein Angriff erkannt, wird die Tastatureingabe solange gesperrt, bis ein Passwort eingegeben wird. Der Angriff wird protokolliert.
- Normal: Wird ein Angriff erkannt, wird die Tastatureingabe vorübergehend

[56] https://globotron.nz/collections/products-by-globotron/products/usg-v1-0-hardware-usb-firewall.

[57] https://github.com/pmsosa/duckhunt.

gesperrt. Der Angriff wird protokolliert.

- Sneaky: Wird ein Angriff erkannt, werden nur ein paar Tastaturbefehle zugelassen. Hierdurch soll der Eindruck erweckt werden, dass der Angreifer einen Fehler gemacht. Der Angriff wird protokolliert.
- LogOnly: Wird ein Angriff erkannt, wird der Angriff lediglich protokolliert.[58]

Bezüglich der Zuverlässigkeit von Duckhunt ist anzumerken, dass die Erkennung von Hotplug-Attacks durch modifizierte Attacken erschwert wird.

7.3.2 Beamgun

Bei Beamgun[59] handelt es sich um ein OpenSource-Programm, das im Hintergrund läuft und nach angeschlossenen USB-Geräten sucht. Beamgun basiert auf dem Prinzip des Whitelistings. Wird ein USB-Gerät, das nicht in der Whitelist steht, von Beamgun erkannt, wird eine der folgende Maßnahmen ausgeführt:

- FocusStealing: Der Fokus der Tastatur wird auf ein bestimmtes Fenster umgeleitet.
- Lock Computer: Der Computer wird gesperrt.[60]

58 https://sarwiki.informatik.hu-berlin.de/USB:_Rubber_Ducky#Group_Policies.

59 https://github.com/JLospinoso/beamgun.

60 https://sarwiki.informatik.hu-berlin.de/USB:_Rubber_Ducky#Group_Policies.

8 Fazit

Hotplug-Attacks sind aufgrund der Grundkonzeption von USB-Schnittstellen und Geräten der HID-Klasse eine ernstzunehmende Gefahr und können ohne größere Funktionsbeeinträchtigungen nicht vermieden werden. Am Markt sind mittlerweile hochentwickelte frei erhältliche Gadgets verfügbar, die sich in Funktionsumfang und Einsatzzweck teils deutlich unterscheiden. Diese Geräte können auch von Laien in der Regel einfach bedient werden. Zudem ist für den überwiegenden Teil dieser Geräte Open Source Payload verfügbar, der nur noch marginal angepasst werden muss. Durch die Hinzunahme von Remote-Funktionen können manche Geräte, wie die O.MG-Tools, mittlerweile ferngesteuert werden, was neue Angriffsvektoren erschließt.

Die Auswahl der Geräte sollten Pentester anhand des jeweiligen Einsatzzwecks vornehmen. Gegebenenfalls können die Geräte auch kombiniert werden. Im praktischen Einsatz entfalten die vorgestellten Geräte insbesondere in Kombination mit anderen Angriffen, wie Social Engineering ihre Wirkung.

Literaturverzeichnis

[1] E. Amberg und D. Schmid: Hacking - Der umfassende Praxis-Guide. 2. Aufl., Frechen: mitp Verlags GmbH & Co. KG, 2022.

[2] M. Kofler, K. Gebeshuber, P. Kloep, F. Neugebauer, A. Zingsheim, T. Hackner, M. Widl, R. Aigner, S. Kania, T. Scheible und M. Wübbeling: Hacking & Security. 3. Aufl., Bonn: Rheinwerk Verlag, 2023.

[3] M. Nicho und I. Sabry: Threat and Vulnerability Modelling of Malicious Human Interface Devices. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM),* Nr. 21, Seiten. 241 - 247, 2022.

[4] S. Potocký und J. Stulrajter: The Human Interface Device Attack form the Perspective of the Attacker and the Forensic Analyst in *New Trends in Signal Processing (NTSP).* Demanovska Dolina, Slovakia, 2022.

[5] R. Khande, S. Rajapurkar, A. Dubey, P. Varade und P. Mahajan: Prevention of code injection from Human Interface Devices (HID). *Scandinavian Journal of Information Systems,* Bd. 35, Nr. 1, Seiten. 699 - 706, 2023.

[6] SarWiki: System Architecture @ Humboldt-Universität zu Berlin: Humbold Universität, 17 Oktober 2017. [Online]. Available: https://sarwiki.informatik.hu-berlin.de/USB:_Rubber_Ducky. [Zugriff am 24 September 2023].

[7] M. Kofler: Linux - Das umfassende Handbuch. 17. Aufl., Bonn: Rheinwerk Verlag, 2021.

[8] N. Nissim, R. Yaholom und Y. Elovici: USB-based attack. *Computer & Security,* Nr. 70, Seiten. 675 - 688 , September 2017.

[9] Y. Lee, H. Lee und K. Yim: Cognitive Countermeasures against BAD USB. in *International Conference on Broadband and Wireless Computing, Communications and Applications*, Asan, Republic of Korea , 2016.

[10] K. Denney, E. Erdin, L. Badun, M. Vai und S. Ulugac: USB-Watch: A Dynamic Hardware-Assisted USB Threat Detection Framework. in *International conference on Security and Privacy in communication Systems*, Orlando, United States, 2019.

Bilderverzeichnis

Tabellenverzeichnis

Verzeichnis der Abkürzungen

HID	Human Interface Device

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die hier vorliegende Arbeit selbstständig, ohne unerlaubte fremde Hilfe und nur unter Verwendung der in der Arbeit aufgeführten Hilfsmittel angefertigt habe.