Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

Projektarbeit Informatik

BlueBorne War Driving

Autoren	Benjamin Gehring Béla Horváth
Hauptbetreuung	Prof. Dr. Bernhard Tellenbach
Nebenbetreuung	Thomas Sutter

Datum	20.12.2019

Inhaltsverzeichnis

1 Formular

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

Erklärung betreffend das selbständige Verfassen einer Projektarbeit an der School of Engineering

Mit der Abgabe dieser Projektarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Projektarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:	Unterschriften:
……	……
	……
	……

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Projektarbeiten zu Beginn der Dokumentation nach dem Titelblatt mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

2 Zusammenfassung

3 Abstract

4 Vorwort

Hier folgt eine Danksagung sowie eine Erwähnung aller beteiligten Personen welche zum Erfolg der Arbeit geführt haben.

5 Einleitung

5.1 Ausgangslage

Das Thema „BlueBorne“ ist keinesfalls neu. Trotzdem finden sich noch immer diverse IoT Geräte welche genau durch diesen Angriff infiltriert werden können. „BlueBorne“ setzt sich aus den Wörtern „Bluetooth“ und „Airborne“ zusammen und beschreibt einen möglichen, sehr infektiösen Angriff über das Bluetooth Protokoll. Um die Sicherheitslücken testen zu können werden Geräte benötigt, welche auf einem veralteten Softwarestand betrieben werden können. In dieser Arbeit wird ein Google Nexus 5x[1] und ein Raspberry Pi Model 3b+[2] als Testgeräte verwendet. Die zugrundeliegenden Whitepaper von Armis [2, 1] wurden als Einstiegspunkt in die Materie gewählt

5.2 Zielsetzung

Das Ziel dieser Arbeit ist es, ein besseres Verständnis der BlueBorne Sicherheitslücke zu erlangen, sowie ein BlueBorne Testing Modul zu entwickeln um potentiell gefährdete Geräte zu erkennen und die Benutzer dieser zu warnen. Dies wird erreicht durch eine vertiefte Einarbeitung in die BlueBorne Thematik mit den korrelierenden CVE [3] Einträgen welche die Angriffe überhaupt ermöglichen. Der Fokus wird dabei auf den Betriebssystemen Android und Linux liegen. Diese Systeme sind am leichtesten in eine unsichere Version zu bringen wodurch sie sich als Testobjekte eignen.

5.3 Anforderungen

Als Ergebniss der Arbeit soll ein Scanner hervorgehen der möglichst viele der 8 BlueBorne Sicherheitslücken auf Geräten, innerhalb eines bestimmten Suchbereichs, erkennt. Der Scanner versucht einen Angriff durchzuführen und meldet bei Erfolg ein entsprechendes vorhandensein der Sicherheitslücke.

5.4 Aufgabenstellung

5.4.1 Research-Ziel

1. Analyse der Bluetooth BlueBorne Sicherheitslücken und vor allem auch der Sicherheitspatches

[1] Getestet mit verschiedene Versionen von Android 7.1.2 und älter
[2] Installiert mit einem Ubuntu Mate 16.04
[3] Eine Liste, gefüllt mit Einträgen für öffentlich bekannte Sicherheitslücken

2. Durchführen einer empirische Analyse
3. Suchen nach anfälligen Geräten an der ZHAW oder allenfalls im Raum Winterthur

5.4.2 Engineering-Ziel

1. Einarbeiten in die Bluetooth-Architektur
2. Entwickeln eines BlueBorne Testing Modules (Für oder ohne das Malware App)

6 Theoretische Grundlagen

6.1 BlueBorne

BlueBorne beschreibt eine Sicherheitslücke in der Bluetooth Implementation von Android, IOS, Linux und Windows Geräten. Alle Geräte welche Bluetooth aktivert haben sind potentiell gefährdet. Die Sicherheitslücke wurde von der Firma Armis entdeckt und im September 2017 öffentlich präsentiert. BlueBorne umfasst total 8 Sicherheitslücken auf verschiedenen Plattformen welche alle zu funktionalen Exploits umgesetzt werden konnten.

6.2 Linux Kernel RCE vulnerability - CVE-2017-1000251

6.3 BlueZ information Leak vulnerability- CVE-2017-1000250

6.4 Android information Leak vulnerability - CVE-2017-0785

Diese Sicherheitslücke wurde im SDP Protokoll des Bluetooth Stacks gefunden und ermöglicht es an bestimmte Memory Daten eines Gerätes zu gelangen. Das SDP Protokoll vermittelt anderen Geräten welche Bluetooth Services das eigene Gerät unterstützt. Dabei stellt das externe Gerät(Client) einen Request der vom eigenen Gerät (Server) mit einer Response beantwortet wird. Diese Response kann maximal die Grösse der MTU des Clients annehmen und muss ansonsten fragmentiert werden. Der Fehler liegt nun im abhandeln dieser Responses, wo der Server berechnet welche Fragmente noch gesendet werden müssen. Der Server schickt im Falle einer Fragmentierung einen Continuation State mit der Response zu dem Client. Der Client sendet darauf einen identischen Request nur mit diesem Continuation State als Zusatz. Der Server weiss nun welches Fragment er als nächstes schicken muss. Ist z.B. die MTU 50 Bytes gross und die Response 80 Bytes sind 2 Fragmente nötig. Die Variable „remaining_handles“ berechnet sich aus „number_response_handles“ und „continuation_offset“. Schafft man es nun dass number_response_handels kleiner ist als der continuation_offset erreicht man einen Underflow. Um dies zu erreichen benötigt man zwei Verbindungen. Im ersten Schritt baut man eine Verbindung zu einem Service auf der eine Response schickt die grösser als die angegebene MTU ist. Den dabei erhaltenen Continuation State verwendet man für den zweiten Schritt wo eine Verbindung zu einem Service aufgebaut wird der eine kleiner Response als die MTU zurückschickt. Innerhalb der Berechnung der remaining_handels Variable wird nun z.B. 2 - 1 gerechnet. Dies führ bei einem uint16 zu einem Underflow und der Server denkt nun er müsse ganz viele Responses zurückschicken. All diese Responses

werden mit einer for-Schleife abgefangen und enthält wichtige Daten des Speichers auf dem Gerät.

6.5 Android RCE vulnerability Nr. 1 - CVE-2017-0781

7 Vorgehen / Methoden

7.1 Research

Als ersten Schritt, analysierten wir die offiziellen WhitePaper der Veröffentlicher von BlueBorne. Um die beschriebenen Erkenntnisse zu bestätigen versuchten wir mithilfe der bereitgestellten Exploit-Programmen der Entdecker von BlueBorne die Attacke nachzubilden.

7.1.1 Sicherheitslücke auf anderen Android-Versionen ausnutzen

Um die Sicherheitslücke zu verstehen wurde unser Google Nexus 5X Testgerät mit einem Android v. 7.1.2 ausgestattet. Bei diesere Androidversion konnten wir sichergehen dass sie verwundbar ist, da wir Testberichte entdeckt haben, in welchen die gleiche Version verwendet wurde. Nach dem der Exploit auf der vogegebenen Basis durchgeführt werden konnte, werden wir nach dem gleichen Prinzip auf anderen Android Versionen vorgehen. Dies soll uns dabei helfen ein besseres Verständnis für die Sicherheitslücke zu erhalten, als auch die Hilfswerkzeuge zu verstehen. Des weiteren wird dies als theoretische Grundlage benötigt, um in erweiterten Teilen der Arbeit auch die nötigen Voraussetzungen zu erfüllen um eigene Test-Maleware zu entwickeln.

7.1.2 Sicherheitslücke in Linux verstehen

Hier wird der Raspberry Pi 3b+ mittels einem veralteten Ubuntu Mate 16.04 installiert. Diese Version verwendet einen Linux Kernel v. X und Bluethooth(BlueZ) v. X. Beides davon ist genügend alt und somit noch von den Sicherheitslücken beschrieben im Punkt 6.2 und 6.3 betroffen.

7.2 Entwicklung

8 Resultate

9 Diskussion und Ausblick

10 Verzeichnisse

10.1 Literaturverzeichnis

11 Literatur

[1] B. Seri and A. Livne. Exploiting blueborne in linux based iot devices, 2019.

[2] B. Seri and G. Vishnepolsky. Blueborne, September 2017.

11.1 Glossar

Begriff	Erklärung
Exploit	Eine Sicherheitslücke (Vulnerabilität) wird auch Exploit genannt. Ein Exploit ist ein nicht absichtlich eingeführter und unbehobener Fehler im Softwarecode, welcher von Angreifern ausgenutzt werden kann, um das System zu infiltrieren oder Malware einzuschleusen.
Malware	Malware beschreibt Bösartige Software / Schadsoftware. Es beschreibt Software welche ausschliesslich dazu entwickelt wurde, schädliche oder böswillige Funktionen auf einem System auszuführen.
Test	test

11.2 Abbildungsverzeichnis

11.3 Tabellenverzeichnis

11.4 Symbolverzeichnis

11.5 Abkürzungsverzeichnis

11.6 Stichwortverzeichnis

12 Anhang

13 Projektmanagement

14 Weiteres