Hello Barbie Initial Security Analysis

Somerset Recon, Inc.

Abstract

Not all Internet of Things (IoT) products are created equal. They span a wide-range of hardware, software, and services. The attack surface of a product can vary depending on the technology that a manufacturer is willing to adopt. This means that poor design decisions and poor implementation can lead to vulnerable devices. Hello Barbie is a combination of Marvell EZ-Connect technology and in-house software and services. Our goal was to analyse the technology being used by Hello Barbie and identify any security issues. Our initial analysis led to the identification of 14 vulnerabilities (4 Medium risk and 10 Low risk). These vulnerabilities represent a snapshot in time[1] and are intended to give Hello Barbie users, security researchers, and IoT companies more information about the security of the Hello Barbie device. The technical information provided throughout this paper will give an insight into Hello Barbie's security and can hopefully be used to enhance the security of IoT users and future IoT products.

1. Introduction

Hello Barbie is an IoT device developed by American toy monolith Mattel and San Francisco startup ToyTalk. Mattel was founded in 1945 and is known for the creation of popular toys, such as Hot Wheels, Chatty Cathy, and Barbie. Mattel had expressed[2] it has been interested in a fully conversation-capable Barbie doll for a while now. ToyTalk, founded by former Pixar employees Oren Jacob and Martin Reddy, is known for its series of educational mobile applications, such as "The Winston Show" and "Thomas & Friends: Talk to You". Together Mattel and Toytalk have created the first IoT Barbie device intended to communicate to children.

Children can start a conversation with the Hello Barbie doll by pressing a button on its belt and talking through the microphone in its chest. Using WiFi, the child's audio recording is sent to ToyTalk's servers, which send back a pre-recorded response. All the conversations are stored in the cloud and can be viewed and managed by parents on ToyTalk's website.

Since the doll's announcement, the doll has sparked quite a bit of controversy within the realm of privacy[3] and security[4]. In particular, there have been concerns about children's data

---

[1] Release date of this Whitepaper: January 25, 2016

[2] http://www.fastcompany.com/3045676/tech-forecast/after-the-fracas-over-hello-barbie-toytalk-responds-to-its-critics

[3] https://www.washingtonpost.com/news/the-switch/wp/2015/03/11/privacy-advocates-try-to-keep-creepy-eavesdropping-hello-barbie-from-hitting-shelves/

being mined, the doll being used as a tool to serve advertisements, and the potential of the doll or the child's voice data being hacked. We were particularly interested in analysing the security of the device and its associated software and services.

2. Summary of Vulnerabilities

The following table is a list of vulnerabilities that were discovered throughout our initial analysis. The severity is a rough estimate of the risk that each vulnerability poses and is rated at either critical, high, medium, or low.

Table 1: Summary of Vulnerabilities (Severity is based on the OWASP risk rating methodology[5])

Vulnerability Description	Severity	Fixed	Additional Information
Weak Passwords	Medium	No	The mobile API and ToyTalk website allows users to use weak passwords
No Password Brute Force Protections	Medium	No	The mobile API and Toytalk website do not prevent brute force password attacks and allow unlimited password guesses
URL Redirect	Medium	Yes	Clients could be sent malicious ToyTalk links, which could redirect to phishing websites or an HTTP version of the ToyTalk website that could expose their session cookie
Sensitive information can be sent over HTTP	Medium	Yes	Several domains used by technology, associated with Hello Barbie, allow unencrypted communication
Stored Cross-Site Scripting	Low	Yes	Malicious Javascript could be stored on the tools.toytalk.com website to allow persistant backdoor access to a ToyTalk user account
Secure flag not set on session cookie	Low	Yes	The session cookie could be transferred unencrypted over HTTP

[4] http://www.sfgate.com/business/article/Will-Barbie-be-hackers-new-plaything-6562963.php
[5] https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

Session cookies do not expire	Low	Yes	The session ID used by ToyTalk does not expire
Hello Barbie device uses unencrypted WiFi network	Low	No	When a Hello Barbie device is in pairing mode[6], the WiFi configuration network is unencrypted
Hello Barbie device does not require unique authentication to modify the configuration of the device	Low	No	Hello Barbie devices use the same mutual authentication certificate. This client certificate can be found in the Hello Barbie Companion mobile application(s). Mutual authentication and configuration of the device is conducted in pairing mode
Password reset page does not expire	Low	Yes	The password reset page on the ToyTalk website does not expire
Android Mobile Application logs Application ID to logcat	Low	Yes	The Application ID is logged in Logcat[7]. If this is leaked through Logcat output, it could be used to hijack a Hello Barbie's session with puppeteer.toytalk.com
Audio files can be accessed without authentication	Low	No	User audio files that are stored on Cloudfront can be accessed without authentication
Cross-Origin Resource Sharing (CORS) is enabled and is not restricted to certain sites	Low	No	Cross-Origin request can be made to puppeteer.toytalk.com from any site
Username Enumeration	Low	No	The mobile API allows attackers to validate the existence of ToyTalk user accounts

[6] Pairing mode is a limited mode initiated by holding two buttons down for 3 seconds on the device. Pairing mode allows a mobile application to configure the device over WiFi.

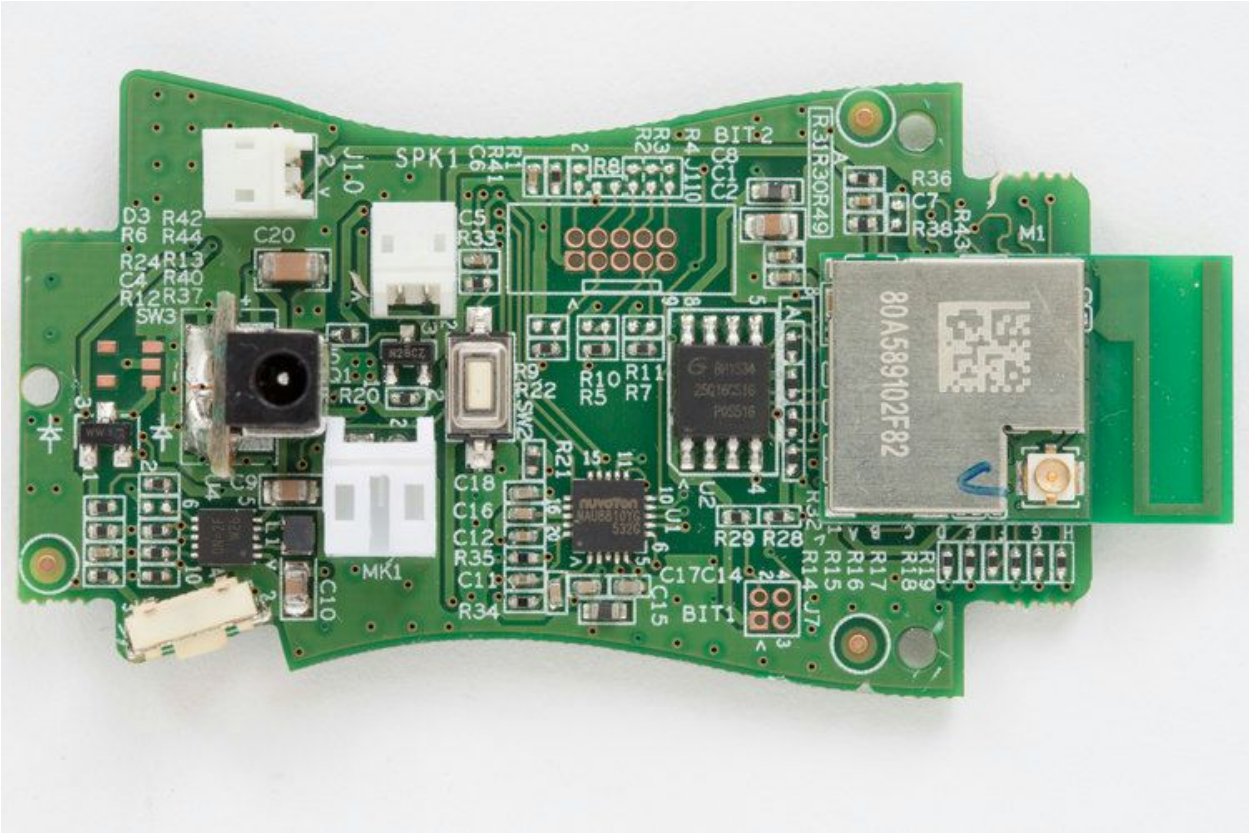
[7] Logcat is available to Android applications with READ_LOG permission running on Android 4.1 and below

3. Analysis

In this section we will describe the methods we used in analyzing Hello Barbie's hardware, firmware, mobile application, and web services.

3.1 Hardware Analysis

Figure 1: Back of Hello Barbie Printed Circuit Board

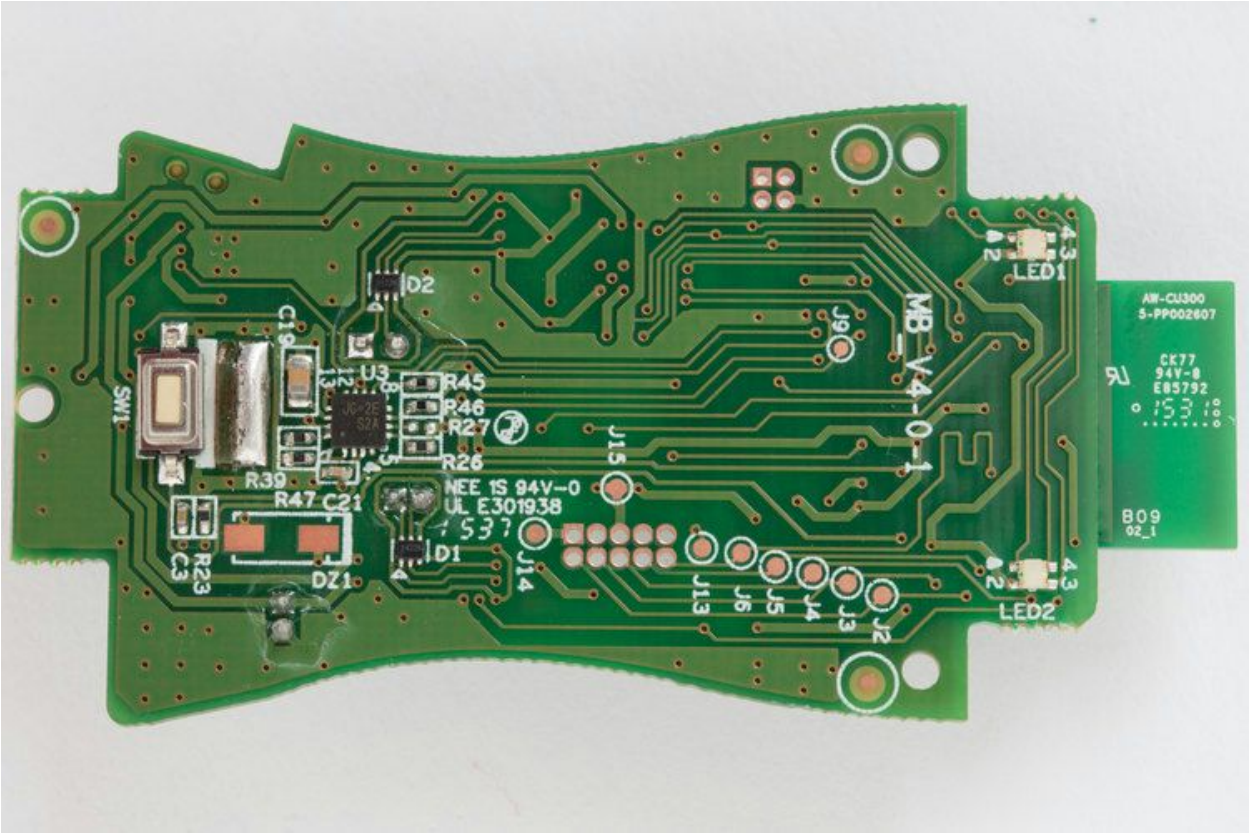


After disassembling the doll and extracting its mainboard, we were able to identify a number of significant chips, modules, and signal connections. The AzureWave AW-CU300E 802.11 b/g/n WiFi Microcontroller Module (M1) carries both the board's Marvell 88MW300 SoC and WiFi module. This is the doll's main chip and all the firmware that was analyzed is utilized by this chip.

The Nuvoton NAU8810 24-bit audio codec (U1) provides ADC, DAC, gain, and input/output mixers for both the doll's microphone and speaker. Its I2C bus connector (J7) was identified by following the traces and through process of elimination. The AW-CU300E is a

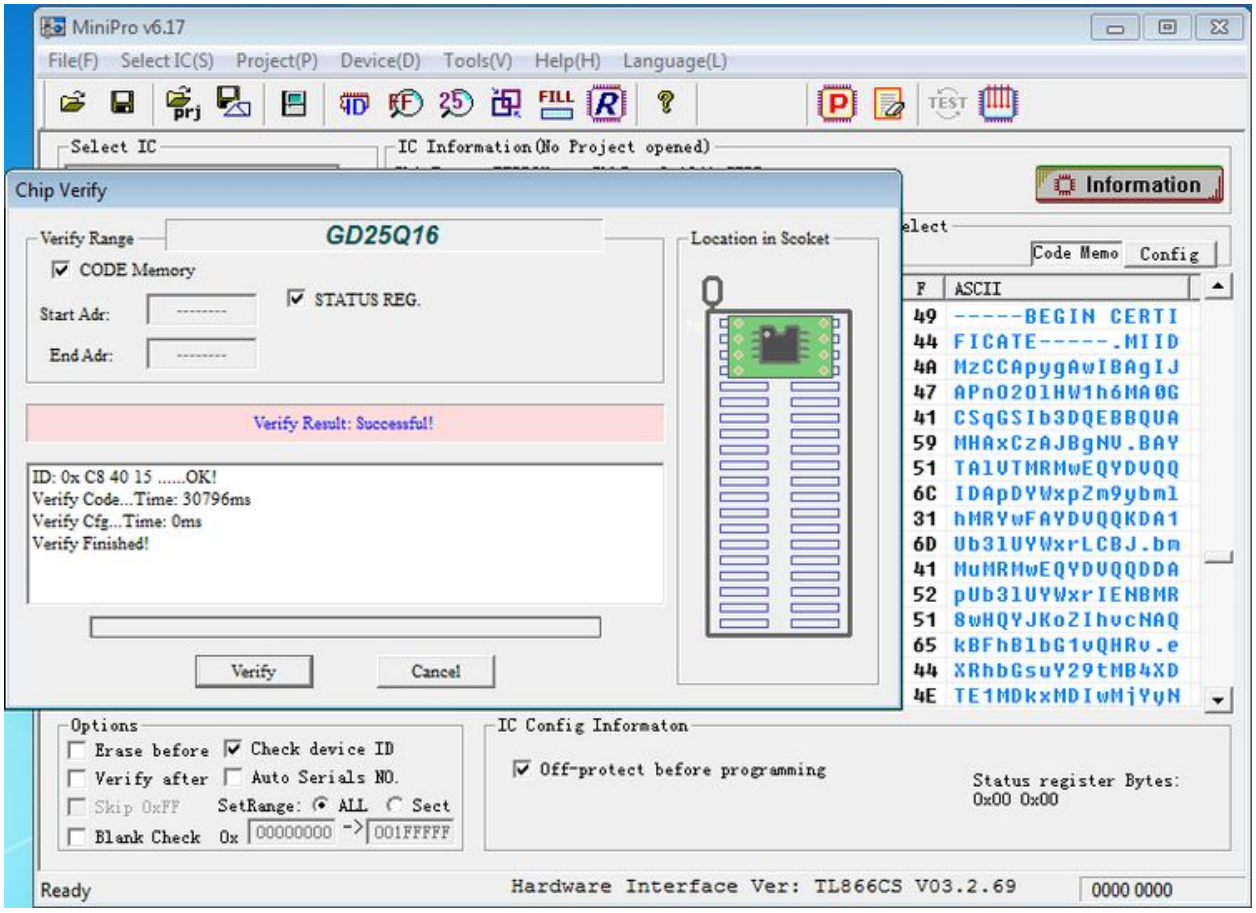
Gigadevice GD25Q16 16Mbit SPI Flash (U2) is the system's main non-volatile memory. This is where the doll's firmware and resource files are stored.

Figure 2: Front of Hello Barbie Printed Circuit Board



Multiple attempts to interface with JTAG were made. We first tried using a JTAGulator and later a SEGGER J-Link. All tests were inconclusive. There existed the possibility that the JTAG passive components were not populated during production, but upon manually populating and testing again this theory was ruled out.

Figure 3: MiniPro Software being Utilized to Dump Flash Memory



However, dumping Barbie's flash memory was successful. It required desoldering the flash chip (GD25Q16) with a hot air rework station and using a flash chip programmer to read the 16-bit address space. Dumping the flash memory gave us a considerable amount of information and allowed us to further analyze how the Hello Barbie device works.

3.2 Binary Analysis

Performing a quick analysis on the memory dump revealed server URLs, API paths, certificates, and even a few MP3 files (network error and shutdown responses). However, the details of how to communicate with ToyTalk's API was not apparent from this information and required locating and reverse-engineering of the microcontroller unit (MCU) firmware. Because the memory dump was in a contiguous binary format, we did not get a clean initial disassembly. We knew that the code should be ARM 32-bit little-endian, but it was not clear where the exact address of the executable section and data sections began. To fix this we generated and applied custom IDA FLIRT signatures. We then realized that pointers to the data section from within the code section were off by a consistent number. Calculating this file-offset, along with

the firmware's loading address cleared everything up and allowed us to further investigate how the doll behaved.

Figure 4: IDA Pro memory dump offset configurations

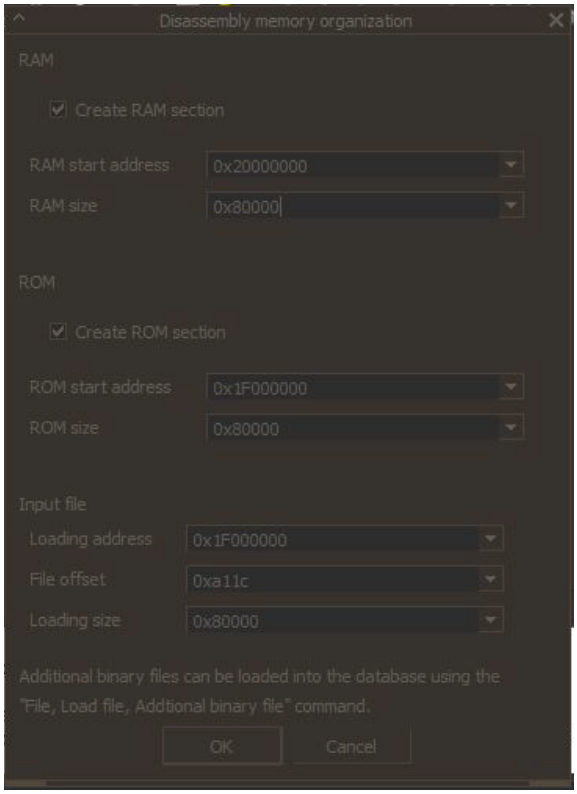
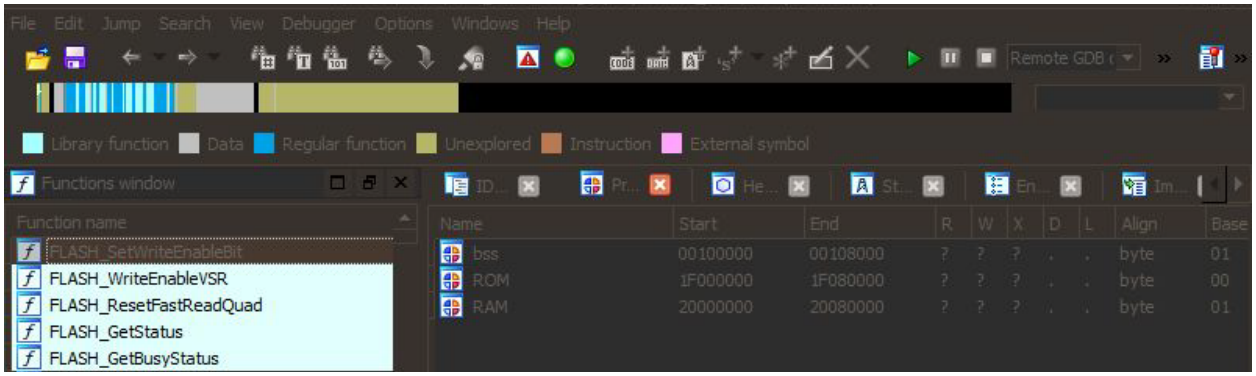


Figure 5: Memory segments



A few memory segments were discovered, but two particularly stood out. The first in particular was the "WM" partition table. The WM partition table mapped out the offsets of all the partitions.

Figure 6: Partition table

Table 2: Partition Table Layout

0x0	Secondary Stage Boot Loader (BOOT2)	Starts up MCU and WiFi modules
0x6000	Persistent Storage Manager (PSM)	An encrypted partition meant to store configuration variables like WiFi network names and WiFi passphrases. The encryption is made with an AES key stored in the MCU's security chip, which makes it nontrivial to extract without dynamic analysis
0xA000, 0x7D000[8]	Microcontroller unit firmware (MCUFW)	Contains most of Barbie's logic and communication code
0xF0000, 0x122000[8]	WiFi firmware (WiFiFW)	Firmware that allows the device to communicate over 802.11 WiFi
0x154000, 0x1AA000[8]	File Table File System (FTFS)	is a read-only FS used to store assets like audio files and certificates

The other interesting memory segment was the File Table File System (FTFS). The FTFS revealed the names and locations of each file in the file system.

[8] Copies are most likely used in a failover scenario.

Figure 7: File Table File System Details

```
ROM:00154116                DCB "êêêêcozy"          ; magic
ROM:00154116                DCD 0x61ED16D3          ; crc
ROM:00154116                DCD 0                   ; obsolete
ROM:00154116                DCD 0x64                ; backend_version
ROM:00154116                DCB 0x30, 0x31, 0x30, 0x30; version
ROM:0015412E                ft_entry <"VeriCA-4.der", 0x280, 0x240>
ROM:0015414E                ft_entry <"PowerDown_06.mp3", 0x4C0, 0x21C9>
ROM:0015416E                ft_entry <"DigiCA-1.der", 0x2689, 0x392>
ROM:0015418E                ft_entry <"DigiCA-2.der", 0x2A1B, 0x243>
ROM:001541AE                ft_entry <"barbie_cert.pem", 0x2C5E, 0xE48>
ROM:001541CE                ft_entry <"Release_04a.raw", 0x3AA6, 0x6780>
ROM:001541EE                ft_entry <"DigiCA-0.der", 0xA226, 0x3B3>
ROM:0015420E                ft_entry <"Press_04.raw", 0xA5D9, 0x6C00>
ROM:0015422E                ft_entry <"barbie_cert_sample.pem", 0x111D9, 0xE48>
ROM:0015424E                ft_entry <"version.txt", 0x12021, 4>
ROM:0015426E                ft_entry <"HB_DX_Embedded_007.mp3", 0x12025, 0x7B12>
ROM:0015428E                ft_entry <"VeriCA-2.der", 0x19B37, 0x4BD>
ROM:001542AE                ft_entry <"toytalk_ca.der", 0x19FF4, 0x2B2>
ROM:001542CE                ft_entry <"HB_DX_Embedded_001.mp3", 0x1A2A6, 0x4580>
ROM:001542EE                ft_entry <"VeriCA-1.der", 0x1E826, 0x4D7>
ROM:0015430E                ft_entry <"VeriCA-3.der", 0x1ECFD, 0x41E>
ROM:0015432E                ft_entry <"VeriCA-0.der", 0x1F11B, 0x240>
ROM:0015434E                ft_entry <"HB_DX_Embedded_005.mp3", 0x1F35B, 0x6AA5>
ROM:0015436E                ft_entry <"HB_DX_Embedded_022.mp3", 0x25E00, 0x2C80>
ROM:0015438E                ft_entry <0>
```

Researching the library of the IDA FLIRT signature that matched was also invaluable in this process. The library was the Marvell WMSDK[9] version 2.13.82, and is described as the Board Support Package that enables cloud service agents, such as Arrayent Connect and Amazon IoT[10], to use the OS (FreeRTOS) and networking resources of Marvell modules. Hello Barbie is essentially this without a major IoT service provider and instead using a proprietary REST protocol to communicate with their servers (Amazon EC2 instances). Despite this, the overarching network design is very similar.

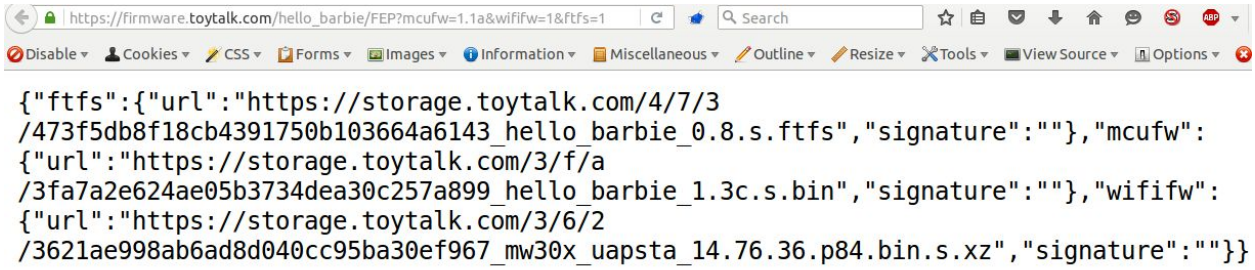
1. Barbie will initially have no wireless access point (WAP) to connect to.
2. Putting Barbie into pairing mode will start a micro-AP (UAP), which is simply an open wireless access point broadcasting a SSID following the pattern “Barbie2xxx”.
3. The mobile application is logged into a user’s account, previously created using either the application or the ToyTalk website. As part of this association, the mobile application will receive an Account ID (account_id).
4. When pairing the mobile application with a Barbie’s UAP, the mobile application will provide Barbie with the Account ID and network configuration to access the Internet.
5. Once pairing mode is complete, Barbie will terminate the UAP.
6. Barbie will initiate a HTTPS (TLS 1.2) connection with puppeteer.toytalk.com.

[9] http://developer.arrayent.com/devkit/marvell/index.php
[10] https://github.com/marvell-iot/aws_starter_sdk/wiki/Getting-Started-Guide

7. Barbie will send/request audio conversations from the puppeteer.toytalk.com using the Account ID as the only means of authentication.
8. Audio from Barbie is uploaded to puppeteer.toytalk.com as 16-bit PCM @ 16KHz. The pupeteer.toytalk.com service then makes the audio available for retrieval at cloudfront.net[11].
9. Audio to Barbie from cloudfront.net is sent using public unauthenticated HTTPS links and can be retrieved by logging into the users ToyTalk account at www.toytalk.com.
10. Additionally, Barbie can receive OTA updates from firmware.toytalk.com, which is unauthenticated but uses HTTPS.

The firmware update mechanism for the Hello Barbie device is interesting from a security perspective. The firmware update sites (firmware.toytalk.com and storage.toytalk.com) use HTTPS (TLS 1.2). The device first makes a request to firmware.toytalk.com[12] and the web service responds with links to the MCU firmware, WiFi firmware and FTFS filesystem image. The firmware images themselves are located at storage.toytalk.com.

Figure 8: Request for Firmware



While the Hello Barbie device does a good job at using an encrypted communication to access firmware updates, there still appear to be some security concerns. OTA updates are a concern because the firmware server does not require authentication and there was no indication of signature checking done in the firmware. Initial investigation suggests that the firmware checking is done using a CRC and not a digital signature. We are currently still investigating the viability of this attack vector.

The Hello Barbie device primarily communicates with puppeteer.toytalk.com and storage.toytalk.com over HTTPS. The device first makes a request to puppeteer.toytalk.com to get a conversation.

[11] https://aws.amazon.com/cloudfront/
[12] Firmware update URL: https://firmware.toytalk.com/hello_barbie/FEP?mcufw=1.1a&wififw=1&ftfs=1

Figure 9: MCU Control Flow of Initial Puppeteer Web Service Call

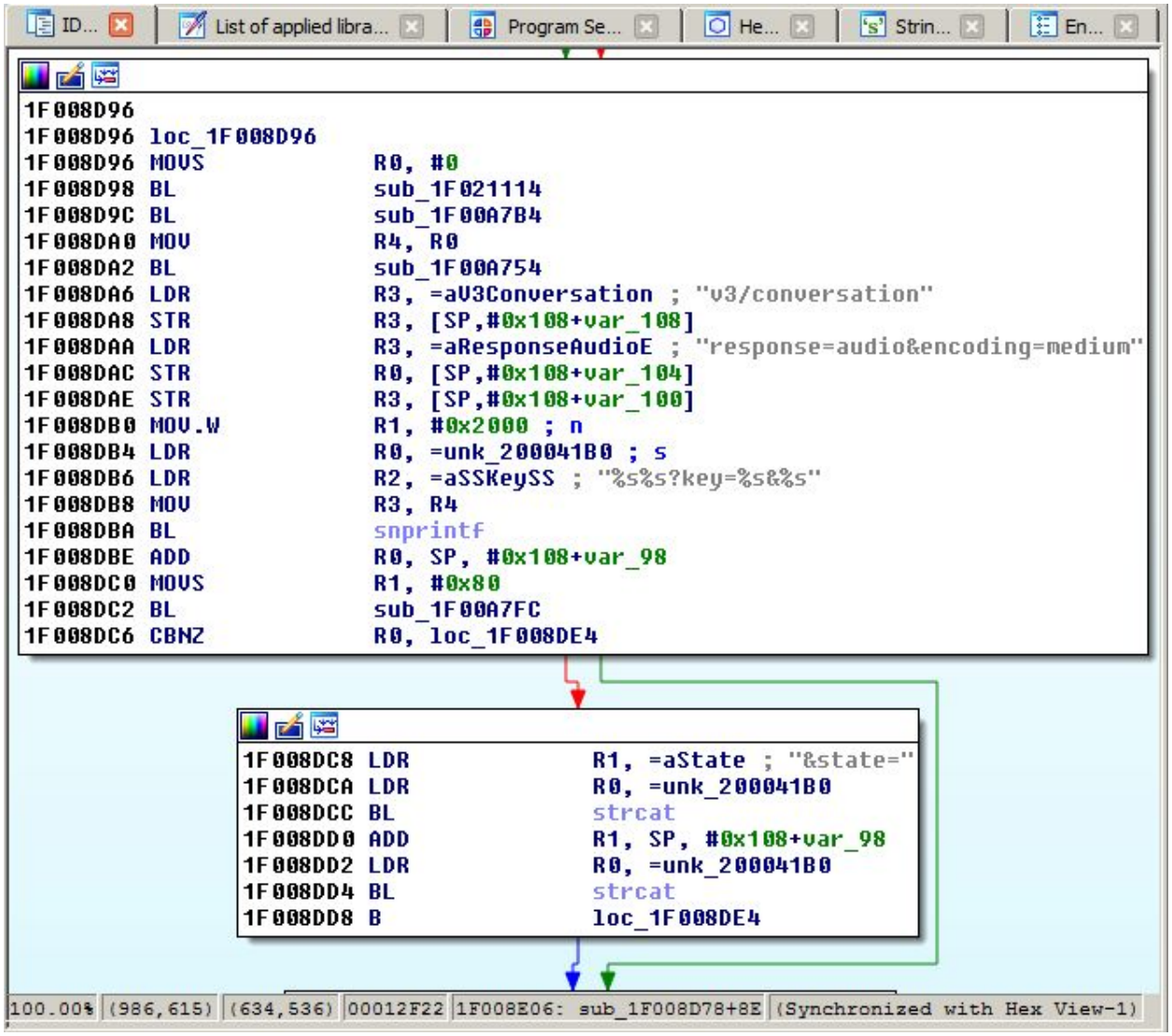
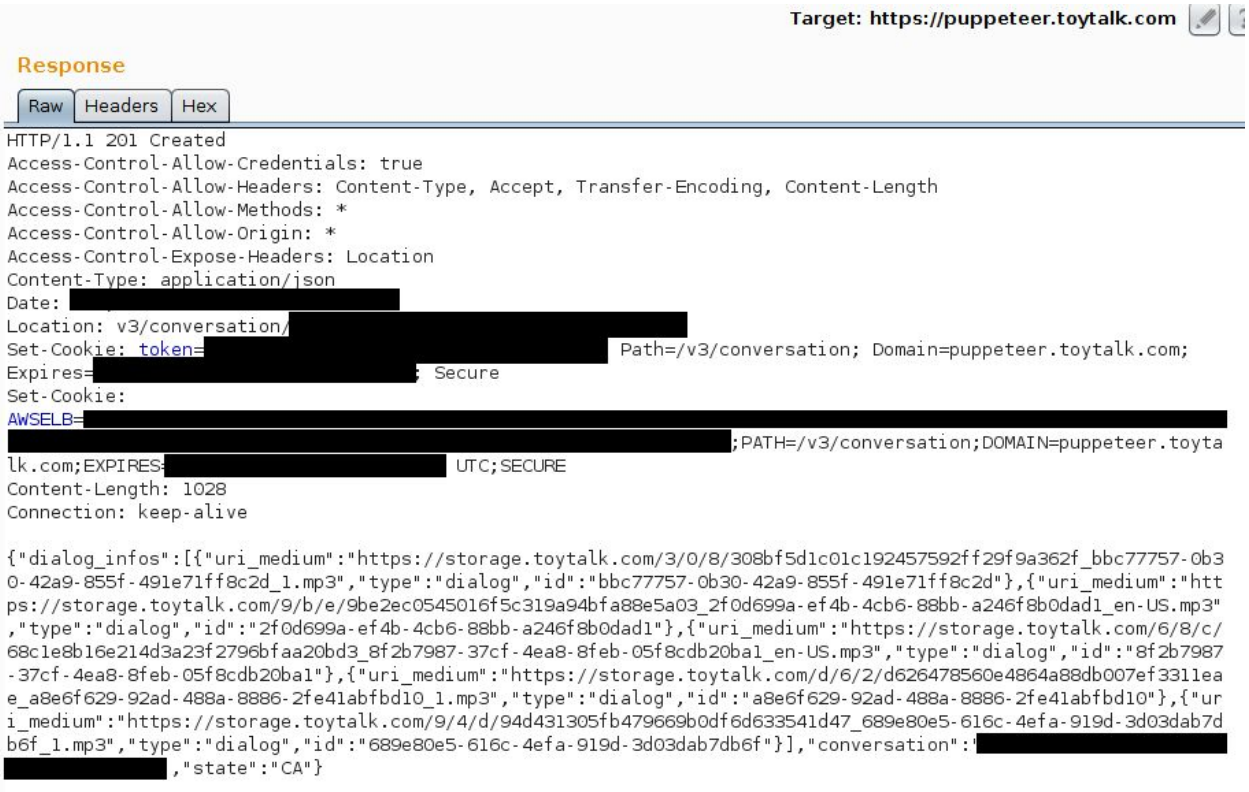


Figure 10: Initial Request for Hello Barbie Audio

Figure 11: JSON Response with Links to Audio Files for Hello Barbie



The puppeteer web service responds with a JSON response and a location of audio files on storage.toytalk.com. The doll retrieves an audio file and plays it to the user. After the user holds down the talk button and replies to the question, the doll sends the voice data to puppeteer.toytalk.com. The request sent with the voice data is sent with an API key and the Account ID.

Figure 12: Example Web Request to Upload Audio to Hello Barbie Account

This means that if a malicious user can steal an Account ID, they can upload arbitrary voice data to the user's account.

3.3 Mobile Application Analysis

The next target in our analysis was the Hello Barbie Companion mobile application. At this point, we had already reversed the protocol between the mobile application and the doll.

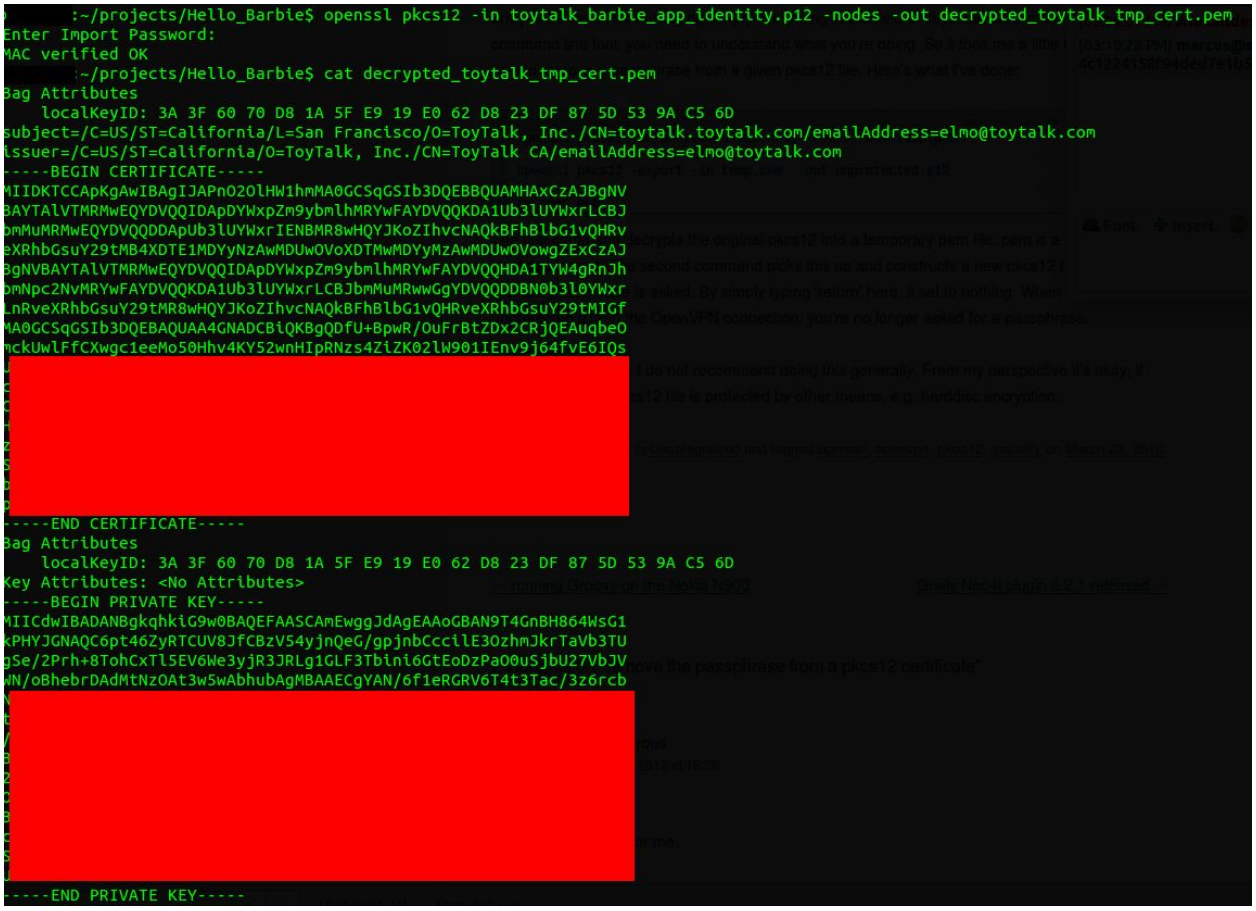
However, we were not able to test it due to the devices implementing mutual authentication. So, our primary goal was to extract the credentials necessary to break this communication. The second reason was to analyze the traffic between the mobile application and ToyTalk, which ended up revealing a few vulnerabilities. We also wanted to find any general vulnerabilities that could be leveraged.

For the sake of simplicity, we analyzed the Hello Barbie Companion Android application and did not analyze the Hello Barbie IOS application. The Hello Barbie Companion Android application is comprised of Unity 3D, Java, natively compiled objects, certificates, and other miscellaneous assets. The Java code is platform specific and most of the proprietary code exists in the ToyTalk library (libToyTalk.so).

It turned out that the mobile application communicates with api.2.toytalk.com and performs mutual authentication, as well. This discovery led us to find a password-protected BouncyCastle Keystore (cacert.bks) in the APK's assets. By decompiling and patching the application, we were able to force the application to disclose the password to us. This was accomplished using CFR decompiler[13] and APKTool to transform the Java bytecode into Smali code. The P12 private certificate (toytalk_barbie_app_identity.p12) that was being used to perform mutual authentication was also password protected. Patching another area of the application force it to disclose this password as well. The result was a public certificate and private key.

[13] http://www.benf.org/other/cfr/

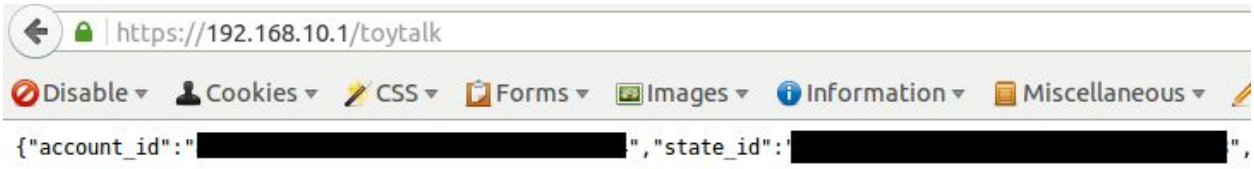
Figure 13: Extracting Public Certificate and Private Key from p12 Certificate File



All of this allowed us to intercept and manipulate mobile application communication (api.2.toytalk.com) and pairing mode communication (https://192.168.10.1).

Figure 14: Mobile Device Logging into Web Service (api.2.toytalk.com)

Figure 15: Hello Barbie Device Web Service Response to Extract Account ID



We were able to verify that most of the routes discovered in the firmware were in fact being served. The information returned from this pairing mode included a couple of interesting things, such as the Account ID, State ID (state_id), network SSIDs, but not the network passphrases.

After performing one last patch to disable certificate pinning, we were able to analyze the mobile API. We discovered a username enumeration flaw. Utilizing this, along with publically accessible information, we enumerated what we believe are 29 ToyTalk employee usernames. We also noticed that there were no brute force password protections on the mobile API and that users are not required to use a strong password for their account. This is a potentially dangerous combination of flaws. This could allow an attacker to find usernames, bruteforce there passwords and potentially take over accounts.

3.4 Web Analysis

The web application was the final and most interesting part of this analysis, for it changed in some way each time we looked at it. The focus was to map out and perform classic penetration-testing on each of the endpoints. These endpoints were extracted from multiple sources, including the firmware, mobile application, and web application.

The following table maps the different domains and how they are used throughout the Hello Barbie technology ecosystem.

Table 3: Hello Barbie Web Technology and Services Overview

Endpoint	URL	Clients	Purpose	Notes
Toytalk Portal	www.toytalk.com	Web Browser, Mobile App	Account administration	Amazon AWS; Authenticated HTTPS
Storage	cloudfront.net/storage.toytalk.com	Web Browser, Mobile App, Doll	Audio file storage	Amazon Cloudfront; Unauthenticated HTTPS
Mobile API	api.2.toytalk.com	Mobile App	Mobile account administration	Amazon AWS; Authenticated HTTPS

Firmware	firmware.toytalk.com	Doll	OTA Firmware Update	Amazon Cloudfront; unauthenticated HTTPS
Puppeteer	puppeteer.toytalk.com	Doll	IoT Service	Amazon AWS; HTTPS with API key and Account ID
SendGrid	sendgrid.toytalk.com	Web Browser	Password Reset	Password Reset over HTTP

Figure 16: User does not Exist

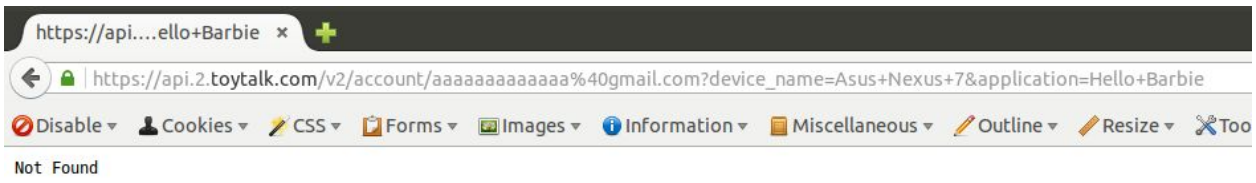
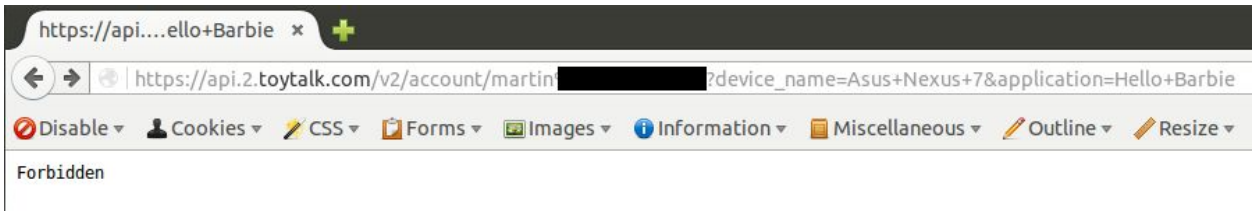


Figure 17: User does Exist



Burp Suite Pro aided us in discovering several web vulnerabilities, which would be rated at medium or low severity, by themselves. However, in combination these issues could lead to a valuable attack. For instance, the fact that the session cookie did not have the secure flag set on it, session cookies did not expire and the site used HTTP as a means of communication, all increased the risk of an attacker being able to compromise a cookie and hijack an account. Likewise, the fact that the password reset site (SendGrid) was using unencrypted HTTP and the password reset page did not expire created a more likely scenario for which an attacker could sniff traffic on a network and reset a user's password.

Many of these web vulnerabilities have now been fixed. It is likely that other bounty hunters independently identified some of the same issues and reported them on hackerone. In the world of bounty hunting these type of vulnerabilities are referred to as "dupes" (or duplicates). They are a valuable metric for the companies hosting the bounty program but they are not monetarily valuable to security researchers.

It was interesting to see how quickly Toytalk patched some of these vulnerabilities. For instance, one night we were taking a look at tools.toytalk.com and the next thing we knew tools.toytalk.com was serving a default nginx page. While the response to these vulnerabilities was quick, it is important to note that many of these issues could have been identified during a pre-production security assessment.

4. Vulnerability Details

In this section we will describe the vulnerabilities discovered in greater detail.

4.1. Weak Passwords

Vulnerability Description: The Hello Barbie Companion mobile application and ToyTalk web interface allow users to use weak passwords. The mobile application specifies that the user's password must be 8 characters long but does not require any additional password complexities.

Affected IPs/URLs:

- https://api.2.toytalk.com
- https://www.toytalk.com

Impact: Users can use weak passwords when signing up for ToyTalk accounts. This allows attackers to remotely brute force user account passwords and gain access to ToyTalk accounts.

Status: Not Fixed

4.2. No Password Brute Force Protections

Vulnerability Description: The mobile API and toytalk website do not protect against remote password brute force attempts. Attackers are allowed unlimited password attempt retries.

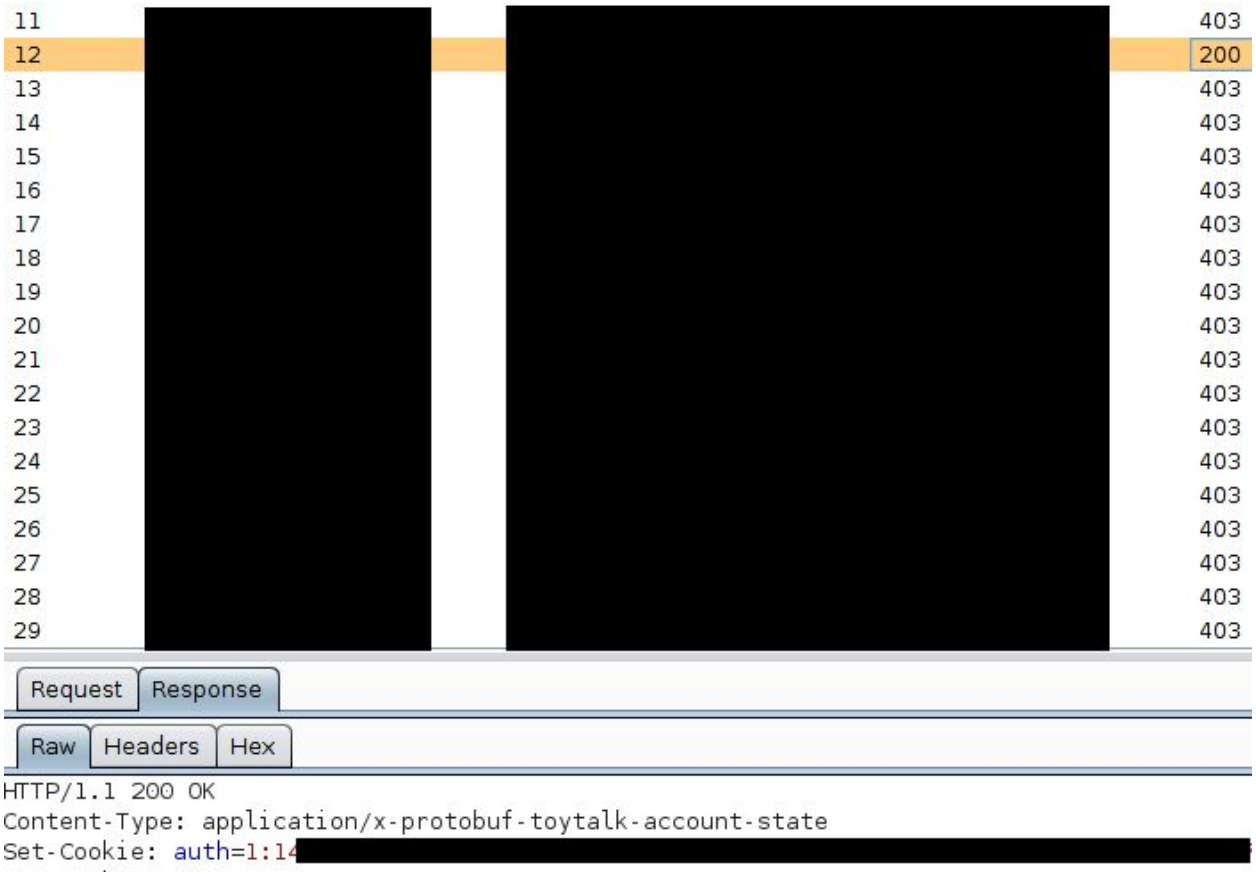
Affected IPs/URLs:

- https://api.2.toytalk.com
- https://www.toytalk.com

Impact: This allows attackers to remotely brute force user account passwords and gain access to ToyTalk accounts. This would provide access to audio content spoken to the Hello Barbie device.

Screenshots:

Figure 18: Burp Intruder Showing Password Brute Forcing and that one Login Request to the Mobile API Successfully Returned an HTTP 200



Status: Not Fixed

4.3. URL Redirect

Vulnerability Description: The ToyTalk website is vulnerable to a URL redirect on its main login page.

Affected IPs/URLs:

- https://www.toytalk.com/login/?next=http://www.idabook.com

Impact: This vulnerability would allow an attacker who sends a malicious ToyTalk URL to a victim to redirect the victim to a malicious website.

Screenshots:

Figure 19: User Visits URL Redirect

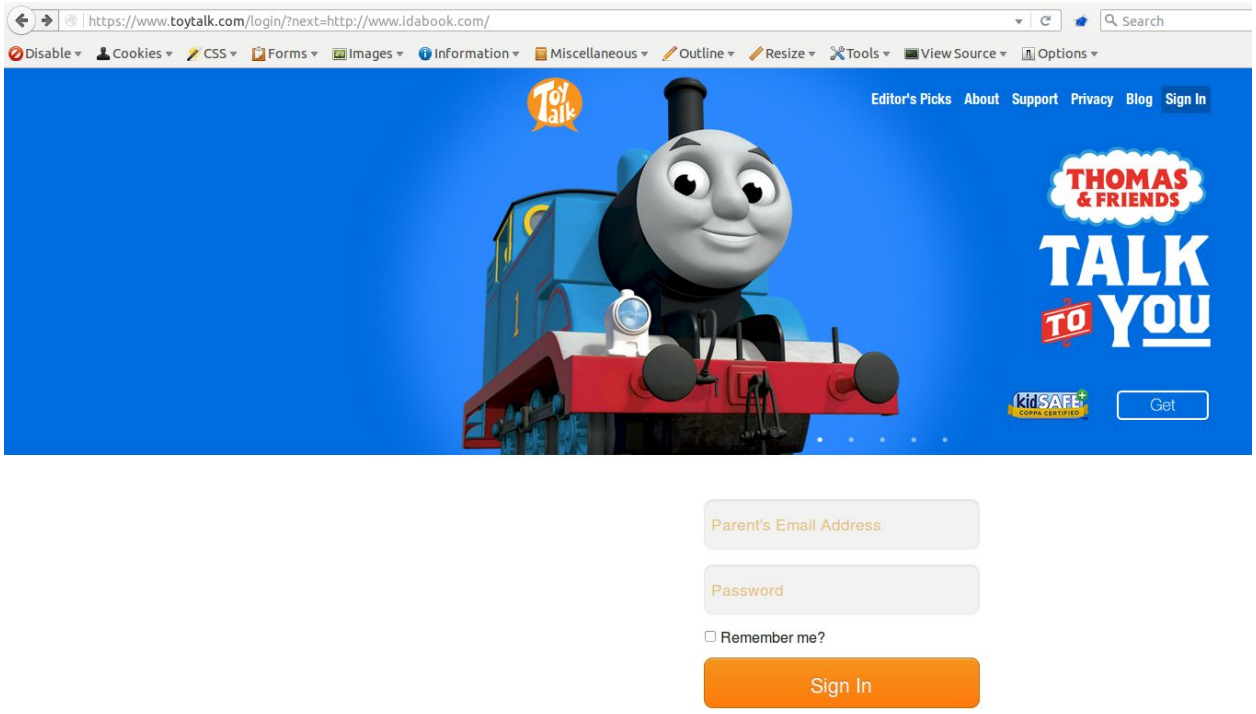
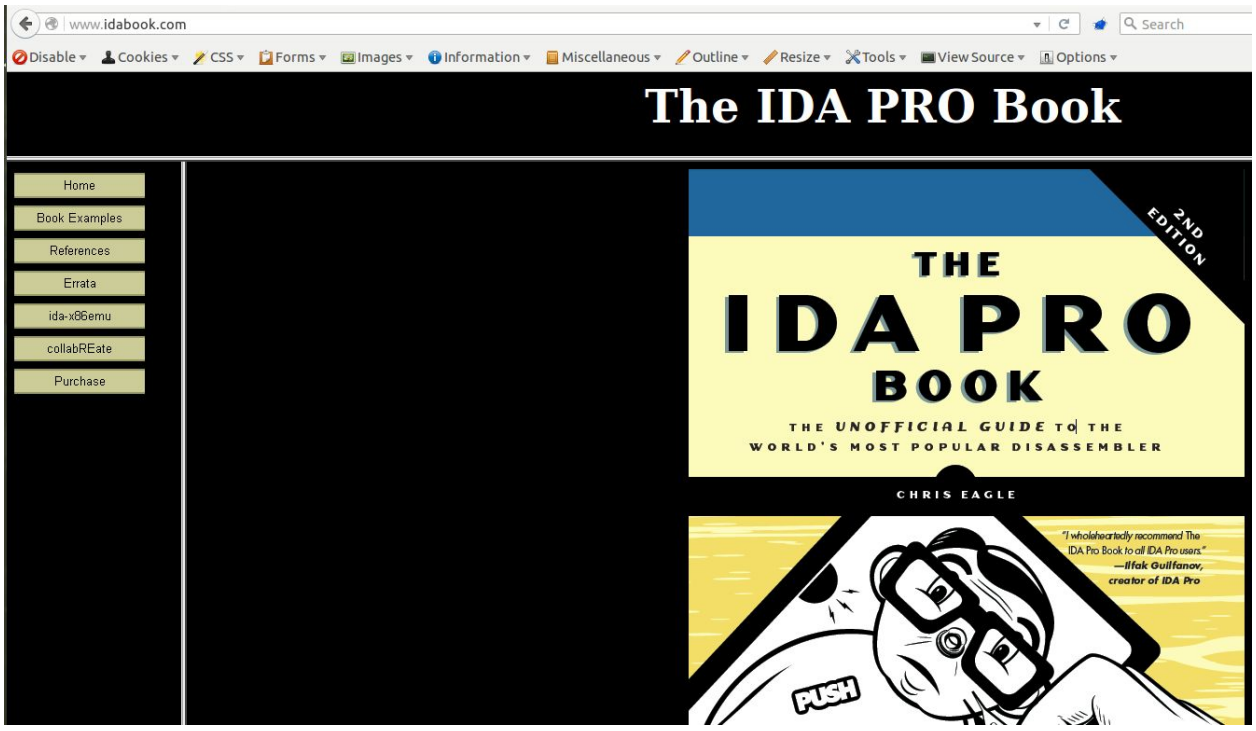


Figure 20: Result After User Logs in and is Redirected



Status: (Fixed) This vulnerability has since been remediated and the website now blocks specific characters needed to redirect to another remote website.

4.4. Sensitive Information can be Sent Over HTTP

Vulnerability Description: Sensitive information, such as password reset information, can be sent over HTTP.

Affected IPs/URLs:

- http://sendgrid.toytalk.com/wf/click?upn=H
- http://api.toytalk.com/liveness.txt
- http://puppeteer.toytalk.com/time
- http://www.toytalk.com/moments/conversations/show/

Impact: If an attacker were able to sniff network traffic and a user reset their password, the attacker would be able to access the reset password page and hijack the user's account.

Screenshots:

Figure 21: Password Reset being Sent Over HTTP

Status: Fixed

4.5. Stored Cross-Site Scripting

Vulnerability Description: The tools.toytalk.com website is vulnerable to stored cross-site scripting (XSS). An authenticated user can change account information and store HTML/Javascript that will be stored on the website and executed after the user logs in.

Affected IPs/URLs:

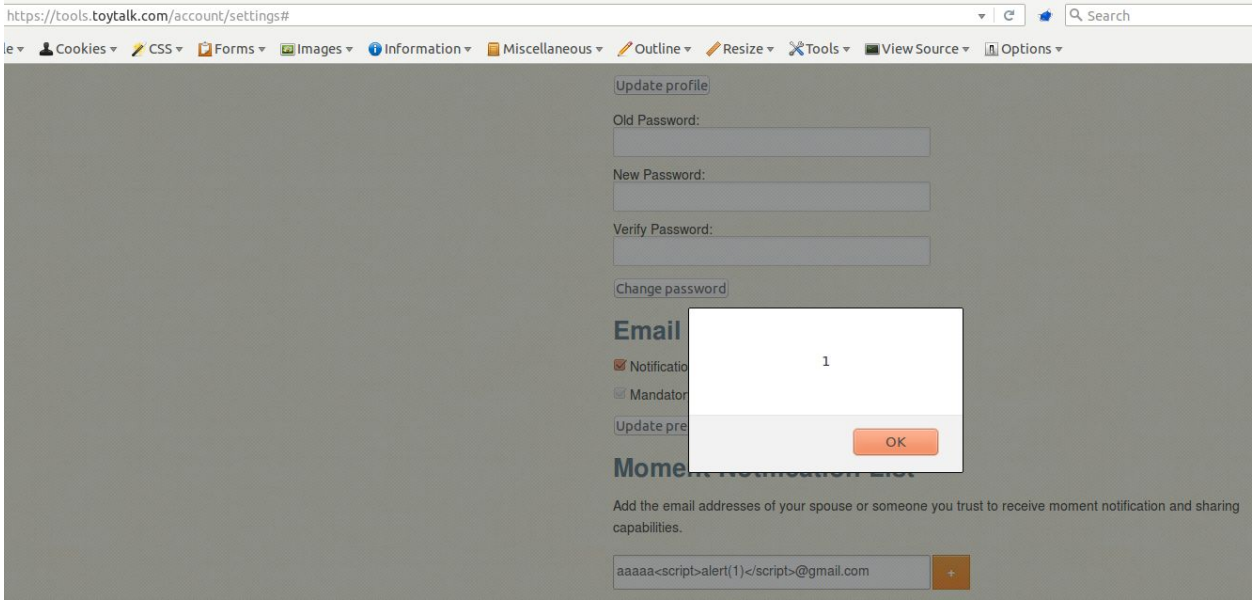
- https://tools.toytalk.com

Impact: If an attacker was able to compromise a ToyTalk account, they could maintain persistence on that account by storing malicious Javascript on the victim's account.

If an attacker were able to perform a man-in-the-middle attack and set the users session cookie, they could redirect them to https://tools.toytalk.com and execute malicious Javascript in the context of https://tools.toytalk.com.

Screenshots:

Figure 22: Stored XSS



Status: (Fixed) This vulnerability has been fixed and the tools.toytalk.com website now serves a default nginx page.

4.6. Secure Flag not set on Session Cookie

Vulnerability Description: Upon initial investigation of the https://www.toytalk.com website, the sessionid cookie did have the secure flag set on it. The secure flag prevents the browser from transmitting the cookie unencrypted over HTTP. The impact of this issue is increased given that the session cookie does not expire and a user can be redirected from the HTTPS version of the site to the HTTP version of the site.

Affected IPs/URLs:

- https://www.toytalk.com

Impact: An attacker might take advantage of these flaws by creating a malicious WiFi network, allowing a parent/user to connect to the network and redirecting the unsuspecting parent/user to the HTTP version of toytalk.com. This would expose the session cookie and the attacker would

be able to compromise the ToyTalk account. This would allow the attacker to listen to all the audio that was spoken to the Hello Barbie device.

Screenshots:

Figure 23: Secure Flag of sessionid Cookie Not Enabled

Status: (Fixed) This vulnerability has since been fixed and the secure flag on the session cookie is now enabled.

4.7. Session Cookies do not Expire

Vulnerability Description: The sessionid cookie does not expire and can be reused to log into the account on ToyTalk.

Affected IPs/URLs:

- https://www.toytalk.com

Impact: If an attacker were able to steal a session cookie an attacker could use that cookie to access the https://www.toytalk.com website and gain access to all of the user's audio files.

Status: (Fixed) This vulnerability has been remediated and session cookies now expire.

4.8. Hello Barbie Device Uses Unencrypted WiFi Network

Vulnerability Description: When the Hello Barbie device is put into pairing-mode (a limited mode initiated by holding two buttons down for 3 seconds on the device) the device creates an open and unencrypted WiFi network. Any WiFi compatible device can join the network.

Affected IPs/URLs:

- https://192.168.10.1

Impact: This security weakness could allow malicious actors to join the network, issue commands to the Hello Barbie device and potentially man-in-the-middle traffic. It is important to note that the WiFi network is only available when the two buttons have been held down on the device. When the device is in a normal operating mode the device does not broadcast an open WiFi network. It is conceivable that while a child is holding the device they could accidently put the device into pairing mode. If an attacker were to take advantage of this weakness, they would need to be in physical proximity of the device or have remotely compromised a neighboring WiFi device.

Status: Not Fixed

4.9. Hello Barbie Device does not Require Unique Authentication to Modify the Configuration of the Device

Vulnerability Description: The Hello Barbie Companion mobile application allows any user to configure any Hello Barbie device while the device is in pairing mode. This allows the mobile device to set the Account ID of the device and associate the doll with an account.

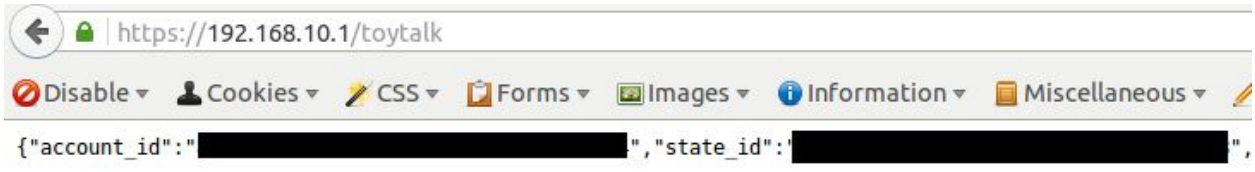
Affected IPs/URLs:

- https://192.168.10.1

Impact: If a compromised computer, compromised mobile device or attacker were in the same proximity to a Hello Barbie device when the device was put into pairing mode, the attacker could set the Barbie to use a specific account and listen to all audio conversations. In this instance an attacker could also retrieve the Account ID by quering the web service at https://192.168.10.1/toytalk. Stealing the Account ID would allow an attacker to inject malicious audio to the user's account.

Screenshots:

Figure 24: Pairing Mode Web Service



Status: Not Fixed

4.10. Password Reset Page does not Expire

Vulnerability Description: The password reset page does not expire and can be reused to reset a user's password.

Affected IPs/URLs:

- http://sendgrid.toytalk.com/wf/click?upn=
- https://www.toytalk.com/user/reset_password/

Impact: When a victim resets their password a link is sent to their email. If the user's email account was later compromised an attacker could reset the victims password for their Hello Barbie account and listen to all the stored audio.

Screenshots:

Figure 25:Sendgrid Redirect to an HTTPS Password Reset Page that does not Expire

Status: (Fixed) The URL expires and the URL becomes unavailable after the user resets his or her password

4.11. Android Mobile Application Logs Application ID to Logcat

Vulnerability Description: When the mobile application is configuring Hello Barbie device in and the Hello Barbie device is in pairing mode the mobile application logs the Application ID to logcat.

Affected IPs/URLs:

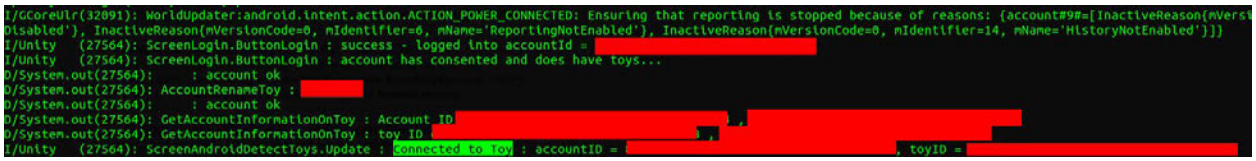
- N/A

Impact: If a malicious Android application was installed on the mobile device and the device was running Android 4.1 or below, the malicious Android application would be able to read the logcat results and read the Application ID. If an attacker is able to retrieve an Application ID they would

be able to insert malicious audio to a ToyTalk account. This vulnerability also requires that at some point the Hello Barbie Companion mobile application is used to configure a Hello Barbie Doll. It would also require that the malicious Android application have the READ_LOG permission.

Screenshots:

Figure 26: Android Logcat Output Login the Account ID



Status: (Fixed) This vulnerability has been fixed for the Hello Barbie Companion Android application (tested version 1.2). However, not all devices are supported to run this version.

4.12. Audio Files can be Accessed Without Authentication

Vulnerability Description: When a parent is administering their ToyTalk account, they can log in to https://www.toytalk.com and listen to all the conversations sent by the Hello Barbie device. These audio files are stored on Cloudfront and can be accessed without authentication. ToyTalk uses this as a feature, so that the original audio files are accessed directly when audio is shared on social media.

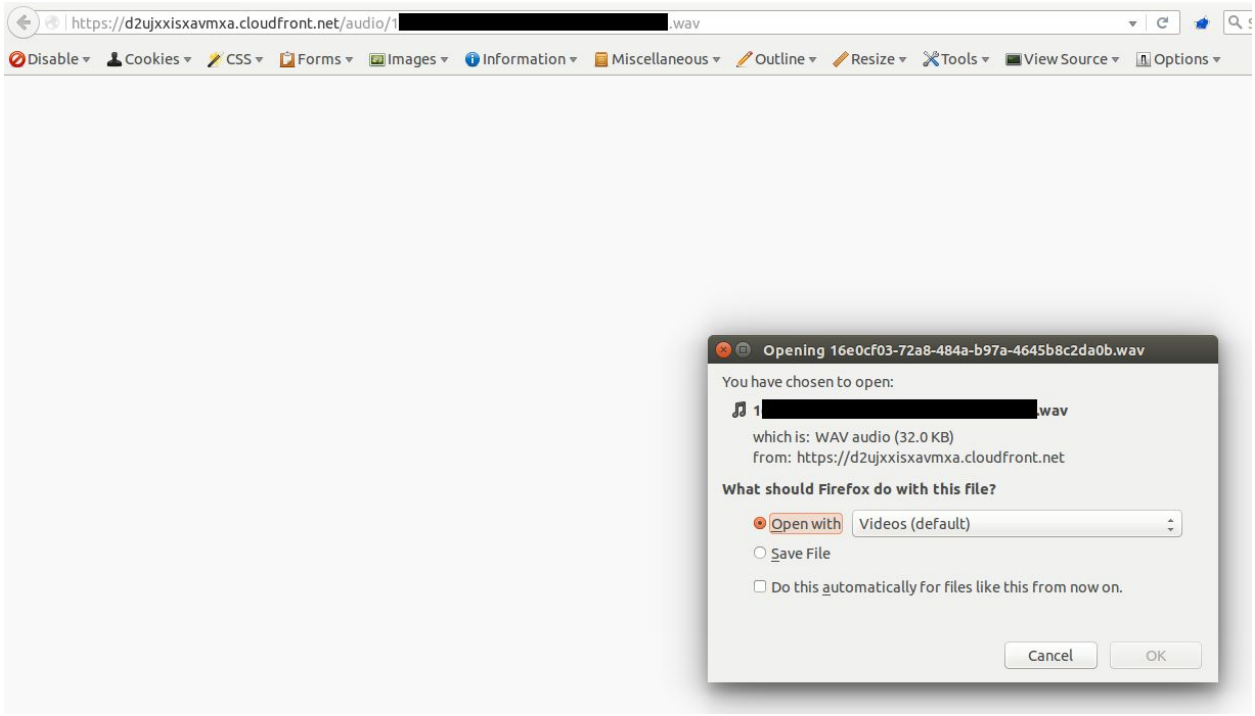
Affected IPs/URLs:

- cloudfront.net

Impact: If an attacker gains access to the URL of an audio file stored on Cloudfront they can remotely access that URL without authentication. The difficulty for an attacker accessing these files without authentication is that the URL paths to the files are somewhat random. If an attacker gains access to the URL of an audio file, the attacker will be able to access the audio file even if a victim changes their password.

Screenshots:

Figure 27: Accessing User Audio File Unauthenticated



Status: Not Fixed

4.13. Cross-Origin Resource Sharing (CORS) is Enabled and is not Restricted to Certain Sites

Vulnerability Description: It seems unnecessary to enable CORS for all origins in this instance.

Affected IPs/URLs: https://puppeteer.toytalk.com

Impact: An attacker can force a victim to make arbitrary requests to puppeteer.toytalk.com using XML HTTP Requests (XHR) and read the responses. This vulnerability could potentially be leveraged to create a more sophisticated Denial Of Service (DOS) attack against puppeteer.toytalk.com. If an attacker were able to inject Javascript in malicious advertisements, they could cause victims to make requests to the server(s) to use up network bandwidth and computing resources. Javascript could then be leveraged to read responses from the web service and tailor the attack.

Screenshots:

Figure 28: Cross Origin Resource Sharing Enabled for All Domains

Target: https://puppeteer.toytalk.com

Response

Raw | Headers | Hex

```
HTTP/1.1 201 Created
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: Content-Type, Accept, Transfer-Encoding, Content-Length
Access-Control-Allow-Methods: *
Access-Control-Allow-Origin: *
Access-Control-Expose-Headers: Location
Content-Type: application/json
```

Status: Not Fixed

4.14. Username Enumeration

Vulnerability Description: The mobile api (https://api.2.toytalk.com) is vulnerable to a username enumeration flaw. An attacker can extract the mobile application's client certificate and perform authentication requests to the server to validate usernames.

Affected IPs/URLs:

- https://api.2.toytalk.com

Impact: An attacker can validate the existence of ToyTalk user accounts by guessing valid usernames. This allowed us to validate 29 ToyTalk accounts likely used by ToyTalk employees.

Screenshots:

Figure 29: User does not Exist

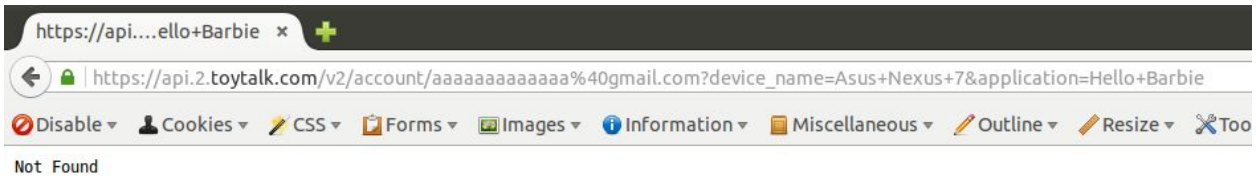
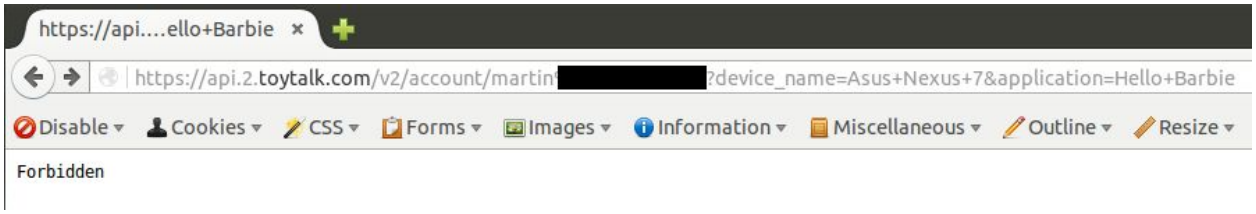


Figure 30: User does Exist



Status: Not Fixed

5. Toytalk Security Enhancements

ToyTalk has made several security enhancements to their website. Here is a list of security enhancements that we have noticed in the last several weeks:

- Fixed several of the vulnerabilities listed in this post
- Fixed the POODLE security vulnerability discovered by the security researchers at Bluebox[14]
- Added the Content-Security-Policy header to help prevent against Cross-Site Scripting attacks
- Added the Strict-Transport-Security header
- Reduced functionaity at https://tools.toytalk.com and in turn, reduced the ToyTalk attack surface
- Fixed several other vulnerabilities reported by security researchers through Hackerone's bug bounty program[15]
- ToyTalk now performs several checks on the requests made to puppeteer.toytalk.com to upload audio files

6. Recommendations

IoT devices are becoming more popular and the only way to fully mitigate the security risks that IoT devices pose is to not purchase or use them. However, since users are willing to accept the security and privacy risks, we will try to provide some security recommendations.

6.1 Hello Barbie User Recommendations

1. Use a strong password[16] for your Hello Barbie account (preferably greater than 12 characters and not a password that has been used before)
2. Try to limit the amount of time the Hello Barbie device is put into pairing mode
3. If the Hello Barbie device stops adding audio to your ToyTalk account (and a compromise to the doll's security looks likely), then delete your ToyTalk account, create another ToyTalk account and associate the new ToyTalk account with your Hello Barbie device
4. Update your Hello Barbie Companion mobile application
5. Run the Hello Barbie Companion application on the latest versions of the Android and iOS operating system
6. Only connect the Hello Barbie doll to WiFi access points that are trusted and using WPA2 with a strong password
7. If your Hello Barbie device is lost or stolen, change your WiFi network password

[14] https://bluebox.com/hello-barbie-app-hello-security-issues/
[15] https://hackerone.com/toytalk
[16] https://www.schneier.com/blog/archives/2014/03/choosing_secure_1.html

8. If someone gives you a Hello Barbie device as a gift and the device is working with no WiFi password configuration, then get a new Hello Barbie device or change the Hello Barbie account and device settings (using the mobile app)

6.2 Future IoT Lessons Learned and Device Recommendations

1. Reducing, disabling or eliminating hardware debugging features, such as serial ports, console ports, JTAG interfaces and others, on IoT devices can help protect users data when a device is lost or stolen
2. Encrypting sensitive data at rest on IoT devices and making it difficult to access the key (for instance, storing the key in protected hardware modules) can be an effective way to lower the risk of personal information being compromised when a device is lost or stolen
3. Identifying web and cloud vulnerabilities (e.g. security requirements, source code audits, vulnerability scans, pentests) before an IoT product is released can be an effective way to reduce the risk of an attack
4. Using a platform like Hackerone for bug bounties can be an effective way to communicate with security researchers and help resolve unknown security issues
5. Using small embedded operating systems, such as FreeRTOS, for IoT devices can reduce the operating system's attack surface
6. If a wireless access point is generated by an IoT device, it would serve as a good security precaution to create a unique, long and random key that the user would use to connect to a WPA2 network
7. Encrypting traffic and validating hosts that issue firmware updates can help prevent man-in-the-middle attacks against firmware update servers
8. Using certificate pinning and mutual authentication for mobile applications can make it more difficult for an attacker to intercept communications and attack web services
9. Proper cryptographic signature validation for firmware images can help mitigate attacks against firmware update mechanisms
10. The KidSafe Seal Program[17] and the Children's Online Privacy Protection Act (COPPA) should specify more strict and clear information security requirements for IoT devices, mobile applications, websites and web services

7. Conclusion

Throughout this analysis we identified 14 vulnerabilities. While there is room for improvement for the security of Hello Barbie, we felt that many security design decisions were well thought out. Since our initial discoveries, ToyTalk has quickly responded to security reports and many fixes have been implemented. It is also evident that Marvell has put a lot of consideration into their software and hardware security, providing an easy to use framework

[17] https://www.ftc.gov/system/files/attachments/press-releases/ftc-approves-kidsafe-safe-harbor-program/kidsafe_seal_program_certification_rules_ftc-approved_kidsafe_coppa_guidelines_feb_2014.pdf

with security mechanisms that were well utilized throughout the Hello Barbie firmware. In the overall data-flow, most of the user's sensitive information is kept on their mobile device and in the cloud, leaving very little on Barbie itself. This sort of architecture is becoming more standardized and will most likely become commonplace within IoT devices in the future.