ENTERPRISE MOBILE THREAT REPORT

Unsecured Firebase Databases: Exposing Sensitive Data via Thousands of Mobile Apps

Q-2 2018

apthority
MOBILE THREAT TEAM
Executive Summary

In 2017, the Appthority Mobile Threat Team (MTT) discovered the HospitalGown vulnerability.

The vulnerability, named for data leaking through backend data stores that are unsecured, results from app developers’ failure to properly secure back-end servers with firewalls and authentication, leading to data exposure. Our initial report in May 2017, revealed that weakly secured backend databases were being accessed via apps used by employees, partners and customers and resulted in numerous security risks, including extensive leaks of sensitive data, easier data access and exfiltration, and increased risks for spear phishing, social engineering, data ransom and other attacks.

This report focuses on the MTT’s latest discovery, a new variant of the HospitalGown vulnerability which occurs when app developers fail to require authentication to a Google Firebase cloud database. Firebase is one of the most popular backend database technologies for mobile apps but does not secure user data by default, provide third-party encryption tools, or alert developers to insecure data and potential vulnerabilities. To secure data properly, developers need to specifically implement user authentication on all database tables and rows, which rarely happens in practice. Moreover, it takes little effort for attackers to find open Firebase app databases and gain access to millions of private mobile data app records.

As a result, a significant mobile data vulnerability exists which has resulted in the exposure of a wide range and large amounts of sensitive data through thousands of mobile apps.

In addition to sharing our methodology and findings around apps with the Firebase vulnerability, we’ve highlighted three examples of apps with this vulnerability that exposed confidential corporate and private employee data.

Such data leaks put enterprises at risk of having their sensitive data exposed, of financial loss, of regulatory and security policy non-compliance, and of large scale breaches, data ransom and other attacks. Enterprises are advised to mitigate their risk of exposure to this vulnerability with ongoing deep app analysis.
Introduction: Uncovering New Data Privacy Vulnerabilities

Exposure of sensitive personal and other data remains a persistent and growing threat. In 2017, the number of data records compromised in publicly disclosed data breaches surpassed 2.5 billion, up 88 percent from 2016. The average cost of a data breach in the US is $7.35M. Breaches can also incur substantial fines (up to 4% of annual revenue or 20 million Euros, whichever is higher, in the case of GDPR) and risk loss of customers and revenues.

In response to ever evolving and increasing mobile threats, the Appthority Mobile Threat Team (MTT) assesses the areas of greatest risk to enterprises and constantly innovates our detection techniques to stay ahead of the risk curve.

In May of 2017, we uncovered a new type of threat that revealed a serious and significant data exposure in enterprise environments, called HospitalGown. HospitalGown exposes vast numbers of Big Data records to hackers, is difficult to detect because enterprises won’t know when a breach of an app vendor’s backend data store has occurred, and is outside the enterprises’ direct ability to prevent or resolve.

Widespread across the mobile ecosystem, HospitalGown is responsible for data leakage due, not to any code in the app, but to the app developers’ failure to properly secure their app’s backend data stores (hence the name). Developers often use backend servers to store persistent user data that programs can mine and analyze. If developers do not specifically implement external security features, for example, through an authentication plug in or API for example, the data stored is available to anyone who knows where to look.

---

2 - 2017 Cost of Data Breach Study Ponemon Institute, June 2017
THE LATEST HOSPITALGOWN THREAT

This report focuses on a new HospitalGown vulnerability that occurs when app developers fail to require authentication to a Google Firebase cloud database. Firebase is a mobile and web application development platform from Google, which MTT research found to be one of the 10 most popular datastores for mobile apps. Developers use the Firebase Realtime Database component as the back-end cloud database to store user authentication details, logs, and other data used by the app. The component is referred to as “Realtime” because it continuously syncs data between the cloud and the user’s mobile device.

The challenge for app developers is that Firebase does not provide adequate security capabilities out of the box. The only security feature available to developers is authentication and rule-based authorization. However, Firebase does not secure user data by default nor are third-party tools available to provide encryption for it. And finally, Firebase does not provide security checkup reports to identify insecure data and potential vulnerabilities. The only way to secure data in a Firebase data store is for the developer to explicitly implement user authentication on all database rows and tables.

Moreover, it takes little effort for attackers to find open Firebase app databases. Once found, cybercriminals can gain access to millions of private mobile data app records by simply appending "/.json" to the server URL, e.g., “https://appname.firebaseapp.com/.json”.

The result is a trove of data that is open to the public internet unless the developer explicitly imposes user authentication on each individual table or directory. Even when developers do implement authentication, they may not secure every database table.

The Firebase vulnerability occurs when app developers fail to require authentication to a Google Firebase cloud database.

It takes little effort for attackers to find open Firebase app databases. Once found, cybercriminals can gain access to millions of private mobile data app records.
In January of 2018, the Appthority Mobile Threat Team (MTT) began developing additional techniques to identify insecure backend servers connecting to mobile apps. We looked at various apps and identified the back-end databases to which they were sending data. Using these techniques, we discovered that Firebase was one of the top 10 most popular data stores for mobile apps.

Next, the MTT looked at the technical details related to how mobile apps access Google Firebase databases. We found that the Firebase database is an API server accessible via “https://<Firebase project name>.firebaseio.com/<database.json>.” Hackers therefore need only make a simple web request (e.g.”https://docs-examples.firebaseapp.com/.json”) to a blank “json” database to view all unprotected data hosted. In a potential threat model, a bad actor could:

• Parse strings and/or look for network connections from apps that are connected to “firebaseio.com” servers.
• Plug the “firebaseio.com” hostname into any web browser, e.g “https://vulnerableapp.firebaseio.com/.json”, quickly viewing all unprotected cloud data stored and used by all users of the app.

MTT applied this threat model to over 2.7 million iOS and Android apps found on mobile devices in enterprises, identifying apps connected to “*.firebaseio.com” Firebase database hosts. We then used the “.json” technique to make network calls that collected all unprotected data for each app, storing the data in a secure location for further analysis.

To determine the impact of the data being leaked, we evaluated the type of sensitive data exposed. We used Data Loss Prevention (DLP) techniques to identify confidential or sensitive information. These included identifying patterns that characterize sensitive data such as the format of a phone number, license plate number or credit card number or attaching “honey tokens” to sensitive data to trace which apps or databases are accessing that data. After running the analysis across all apps, we had a clear picture of the impact and scale of the vulnerability.

In the end, we uncovered a large number of vulnerable apps and variety of types of sensitive data, including PII, plaintext passwords, private access tokens, vehicle license and registration numbers, and more leaked by the vulnerable apps.

Appthority notified Google about this vulnerability and provided a list of affected apps and database servers. For the vulnerable apps mentioned in this report, we contacted the app developer according to our Security Disclosure Policy and have verified that they’ve remediated the vulnerabilities.

We uncovered a large number of vulnerable apps and variety of types of sensitive data, including PII, plaintext passwords, private access tokens, vehicle license and registration numbers, and more leaked by the vulnerable apps.

Methodology

In January of 2018, the Appthority Mobile Threat Team (MTT) began developing additional techniques to identify insecure backend servers connecting to mobile apps. We looked at various apps and identified the back-end databases to which they were sending data. Using these techniques, we discovered that Firebase was one of the top 10 most popular data stores for mobile apps.

Next, the MTT looked at the technical details related to how mobile apps access Google Firebase databases. We found that the Firebase database is an API server accessible via “https://<Firebase project name>.firebaseio.com/<database.json>.” Hackers therefore need only make a simple web request (e.g.”https://docs-examples.firebaseapp.com/.json”) to a blank “json” database to view all unprotected data hosted. In a potential threat model, a bad actor could:

• Parse strings and/or look for network connections from apps that are connected to “firebaseio.com” servers.
• Plug the “firebaseio.com” hostname into any web browser, e.g “https://vulnerableapp.firebaseapp.com/.json”, quickly viewing all unprotected cloud data stored and used by all users of the app.

MTT applied this threat model to over 2.7 million iOS and Android apps found on mobile devices in enterprises, identifying apps connected to “*.firebaseio.com” Firebase database hosts. We then used the “.json” technique to make network calls that collected all unprotected data for each app, storing the data in a secure location for further analysis.

To determine the impact of the data being leaked, we evaluated the type of sensitive data exposed. We used Data Loss Prevention (DLP) techniques to identify confidential or sensitive information. These included identifying patterns that characterize sensitive data such as the format of a phone number, license plate number or credit card number or attaching “honey tokens” to sensitive data to trace which apps or databases are accessing that data. After running the analysis across all apps, we had a clear picture of the impact and scale of the vulnerability.

In the end, we uncovered a large number of vulnerable apps and variety of types of sensitive data, including PII, plaintext passwords, private access tokens, vehicle license and registration numbers, and more leaked by the vulnerable apps.

Appthority notified Google about this vulnerability and provided a list of affected apps and database servers. For the vulnerable apps mentioned in this report, we contacted the app developer according to our Security Disclosure Policy and have verified that they’ve remediated the vulnerabilities.
Our analysis revealed that this Firebase variant of the HospitalGown vulnerability has broad ranging impacts. Organizations across industries and globally are impacted. A wide variety of data types are exposed from information subject to data privacy regulations, to sensitive intellectual property and sales information, from data that can cause potential financial fraud, to PII. As the number of apps that take advantage of Firebase continues to grow, this problem will only increase.

A GROWING PROBLEM

Our research shows that as the number of applications that rely on Firebase has grown, so too have the number of vulnerable apps:

• Between 2015 and 2016, the total number of apps using Firebase grew 2,112 percent while the number of vulnerable apps grew 1,225 percent
• Between 2016 and 2017, the number of apps using Firebase grew 271 percent while the number of vulnerable apps grew 74 percent.
• Based on the data so far in 2018, it looks like both apps using Firebase and the number of vulnerable apps will continue to grow.

<table>
<thead>
<tr>
<th>Year</th>
<th>Apps Using Firebase DB</th>
<th>Vulnerable Apps</th>
<th>Vulnerable Apps %</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>3</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>2014</td>
<td>48</td>
<td>10</td>
<td>21%</td>
</tr>
<tr>
<td>2015</td>
<td>646</td>
<td>199</td>
<td>31%</td>
</tr>
<tr>
<td>2016</td>
<td>14,288</td>
<td>2,636</td>
<td>18%</td>
</tr>
<tr>
<td>2017</td>
<td>53,010</td>
<td>4,578</td>
<td>9%</td>
</tr>
<tr>
<td>2018 YTD</td>
<td>21,250</td>
<td>1,693</td>
<td>8%</td>
</tr>
</tbody>
</table>

OUR RESEARCH SHOWS

2015/16
Apps using Firebase grew 2,112% while the number of vulnerable apps grew 1,225%

2016/17
Apps using Firebase grew 271% while the number of vulnerable apps grew 74%

2018
Both apps using Firebase and the percentage of vulnerable apps will continue to grow
SCOPE OF THE THREAT

Out of a total of 2,705,987 apps analyzed, 27,227 Android apps and 1,275 iOS apps were found to be connected to a Firebase database. Of those connected apps, we found that:

• 1 in 11 Android apps (9%) and almost half of iOS apps (47%) that connect to a Firebase database were vulnerable
• More than 3,000 apps were leaking data from 2,300 unsecured servers. Of these, 975 apps were in active customer environments.
• 1 in 10 Firebase databases (10.34%) are vulnerable
• Vulnerable Android apps alone were downloaded over 620 million times
• Over 100 million records (113 gigabytes) of data was exposed

<table>
<thead>
<tr>
<th>Description</th>
<th>Count</th>
<th>Total %</th>
<th>Android</th>
<th>iOS</th>
<th>Android %</th>
<th>iOS %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Apps with FirebaseDBs</td>
<td>28,502</td>
<td></td>
<td>27,227</td>
<td>1,275</td>
<td>95.53%</td>
<td>4.47%</td>
</tr>
<tr>
<td>Apps Vulnerable</td>
<td>3,046</td>
<td>10.69%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Vulnerable by OS</td>
<td></td>
<td>8.98%</td>
<td>47.06%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Firebase DB by OS</td>
<td>21,972</td>
<td></td>
<td>21,193</td>
<td>945</td>
<td>96.45%</td>
<td>4.30%</td>
</tr>
<tr>
<td>FirebaseDBs Vulnerable</td>
<td>2,271</td>
<td>10.34%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Vulnerable by OS</td>
<td></td>
<td>8.88%</td>
<td>46.56%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ORGANIZATIONS IMPACTED

• Most organizations are affected. We found that 62% of enterprises have at least one vulnerable app in their mobile environment.
• No type of organization is immune. Organizations with data exposed to this HospitalGown variant included banks, telecoms, postal services, ride sharing companies, hotels and educational institutions.
• Organizations worldwide were affected. Enterprises in the United States, Europe, the United Kingdom, Argentina, Brazil, Singapore, Taiwan, New Zealand, India, and China were exposed.
APPLICATIONS COMPROMISED

Applications of all categories were affected by this vulnerability, including:

- Tools
- Productivity
- Communication
- Finance and business
- Health and fitness

Among these sensitive categories, health & fitness apps had the most data records leaked. This is of particular concern because healthcare data is far more valuable to hackers than other types of data. Medical information can be worth ten times more than credit card numbers on the deep web. Fraudsters can use this data to create fake IDs to buy medical equipment or drugs, or combine a patient number with a false provider number and file fictional claims with insurers.

DATA LEAKED

Apps connected to unsecured Firebase databases have exposed more than 100 million data records, including:

- 2.6 million plain text passwords and user IDs
- 4 million+ PHI (Protected Health Information) records, including chat messages and prescription details
- 25 million GPS location records
- 50 thousand financial records including banking, payment and Bitcoin transactions
- 4.5 million+ Facebook, LinkedIn, Firebase and corporate data store user tokens

Regulated Data

Regulated data is being leaked: Some of the data leaked includes highly sensitive private information subject to regulatory requirements such as HIPAA, GDPR, and PCI. Such leaks can trigger expensive regulatory fines and breach notification requirements which can damage the company’s reputation. Exposed regulated data included:

- Medical information: Chat messages between patients and pharmaceutical sales representatives together with their prescriptions and orders leaked from a Mexico-based pharmaceutical app. Information about medical consultations, medical history, and diagnostic information was also exposed. This information is protected by HIPAA regulations and subject to breach notification requirements.
- Sensitive personal data: Email addresses, phone numbers, full names, geolocations, and Facebook OAuth tokens were all leaked in violation of data privacy protection laws such as GDPR.
- Vehicle license plate numbers: Some apps exposed vehicle license and registration numbers as well as geolocation data. California has numerous data privacy laws, some of which require companies that expose names accompanied by license plate numbers, and other automobile details to disclose these breaches.
- Credit card numbers: Credit card numbers, which were also exposed, are subject to protection in accordance with PCI DSS.
Personal Data

In some cases, leaked data may not be subject to regulatory requirements, but can nonetheless expose private or financial information. Leaked information of this type included:

- Private messages from a networking app that uses Artificial Intelligence (AI) for match making
- Voice recordings from a voice-based dating app
- Registered email addresses from an unofficial spy recording app that allows users to take videos when the app is not in the foreground
- A cryptocurrency wallet app leaked transaction history and total amount of bitcoins that users owned. This information could allow a bad actor to easily drain cryptocurrency accounts potentially containing hundreds of thousands of dollars in cryptocurrency.

Sensitive Enterprise Data

When internal company data is leaked, organizations can lose intellectual property (IP), damaging their viability and competitiveness. Forty percent of vulnerable apps installed are business related, increasing the risk for IP loss. The app leaks we found included:

- Corporate private keys and access credentials. With this information in hand, cybercriminals have free reign in a corporate network and can potentially exfiltrate sensitive intellectual property, such as patent information and plans for future products.
- Private conversations. Productivity apps leaked data about private business conversations.
- Sales info. Leaks of corporate sales details can give competitors important information about a company’s customers.

The MTT documented the exposure from three diverse examples of mobile apps in enterprise environments — Workhive, Booster Fuels, and CryptoPort — that illustrate the risk of data loss from their insecure Firebase databases. We have worked with the developers to ensure that the apps detailed in this report are no longer vulnerable.
Data Exposure Examples

WORKHIVE: TEAM CHAT

Businesses use the "Workhive: Team chat app" for iOS and Android for team communication and collaboration. The app provides real-time messaging, manages files and important documents, and syncs across devices.

The MTT found that the app did not secure most or all of the data organizations were using within the Workhive app. We found private data from 960 organizations in a sample set of 380 MB of exposed data with over 594,000 records. In addition to business and employee records, we found private and group messages containing highly sensitive business data, such as:

- Financial data and employee medical records.
- User email credentials with plaintext passwords from over 150 different corporate domains
- Infrastructure cloud credentials, including Amazon “Secret Access Keys” and more than 40 server addresses with root plaintext passwords.

As noted earlier, exposure of financial and medical records leaves organizations open to violations of regulatory requirements. Financial and medical records are also particularly vulnerable to attack because of their high potential value. Leakage of plaintext passwords is a particular risk because many individuals reuse their passwords, so this type of leakage can potentially expose data from many applications.

© 2018 Appthority. All rights reserved.
BOOSTER FUELS: ON DEMAND FUEL

The Booster Fuels app claims to “eliminate the gas station errand permanently” by coming to you to fill up your gas tank. Currently at select corporate campuses in California and Texas, users create an account using a credit card and input a time window, location, and vehicle information.

Apptivity found that the app was exposing nearly 10,000 records, including:

• The full name and mobile device geolocation of app customers
• Tanker details, including the driver’s name, geolocation, fuel levels, license plate, and vehicle identification numbers. (FIGURE 1)
• Customer fuel delivery details, including customer name, email, and telephone number in combination with their car details, including the car geolocation, make and model of car, and license plate number. (FIGURE 2)

California has numerous data privacy laws, some of which require companies that expose names accompanied with license plate numbers and other automobile details to disclose these breaches. Booster Fuels was in violation of these data privacy regulations and subject to breach notification requirements.
"XXXXXXX": {
  "USDOT": "XXXXXX",
  "_id": "XXXXXXX3",
  "chassisMake": "Isuzu",
  "chassisModel": "NPR-XD",
  "chassisYear": 2016,
  "currentDriverFullName": "John Doe",
  "currentDriverId": "XXXXXX",
  "currentFuelLevel": 600,
  "currentFuelType": "DIESEL",
  "currentServiceLocationId": "XXXXXX",
  "lastLocation": {
    "geometry": {
      "coordinates": [-122.000, 37.00],
      "type": "Point"
    },
    "properties": {
      "accuracy": 5,
      "altitude": 5,
      "course": 12345,
      "timestamp": "2018-01-15T19:03:09.000Z"
    },
    "type": "Feature"
  },
  "locationUpdatedAt": "2018-01-15T19:03:23.059Z",
  "oldId": "XXXX-XXXX",
  "plate": "XXXXXXX",
  "regionId": "111111111111111111111111",
  "shortId": "CA-XXX",
  "state": "CALIFORNIA",
  "status": "ON_TRIP",
  "tanks": {
    "XXXXX": {
      "_id": "XXXXX",
      "contents": [
        {
          "gallons": 480.25,
          "tfmId": "XXXX"
        }
      ],
      "currentFuelLevel": 480.25,
      "currentFuelType": "REGULAR",
      "lastTfmId": "XXXXXX",
      "orientation": "FRONT",
      "tankerId": "XXXXX",
      "totalCapacity": 700,
      "updatedAt": "2018-01-15T18:50:41.127Z"
    },
    "XXXXX": {
      "_id": "XXXXX",
      "contents": [
        {
          "gallons": 480.355,
          "tfmId": "XXXXX"
        }
      ],
      "currentFuelLevel": 480.355,
      "currentFuelType": "PREMIUM",
      "lastTfmId": "XXXX",
      "orientation": "BACK",
      "tankerId": "XXXX",
      "totalCapacity": 500,
      "updatedAt": "2018-01-15T18:21:37.708Z"
    }
  },
  "totalCapacity": 1200,
  "type": "TONKA",
  "updatedAt": "2018-01-15T19:03:23.060Z",
  "vin": "XXXX",
  "vinliCaseId": "XXXX",
  "vinliDeviceId": "XXXXXXXX"}}
"_id": "XXXXX",
"completeByTime": "2018-01-16T00:30:00.000Z",
"customer": {
  "_id": "XXXXX",
  "email": "user@company.com",
  "firstName": "John",
  "lastName": "Doe",
  "numCompletedRequests": 28,
  "telephone": "6505555555"
},
"customerId": "XXXXX",
"deliveryWindow": {
  "canBeScheduled": true,
  "cutoffTime": "10:00",
  "description": "Request by 10AM",
  "duration": 3600000,
  "enabled": true,
  "end": "16:30",
  "id": "anytime",
  "name": "All-Day | Delivery By 4:30PM",
  "price": 0,
  "requestStartTime": "04:00",
  "start": "08:00",
  "type": "FIXED",
  "waived": false
},
"regionId": "XXXXX",
"requestedAt": "2018-01-15T14:51:26.973Z",
"serviceLocationId": "XXXXX",
"serviceLocationZoneId": "XXXXX",
"status": "ACTIVE",
"territoryId": "XXXXX",
"vehicle": {
  "_id": "XXXXX",
  "color": "Grey",
  "customerId": "XXX",
  "fuelDispensed": 345.732,
  "fuelType": "REGULAR",
  "isVerified": true,
  "lockingFuelDoor": true,
  "make": "Honda",
  "model": "Accord",
  "mpg": {
    "city": {
      "max": 13,
      "min": 9
    },
    "combined": {
      "max": 11,
      "min": 8
    },
    "hwy": {
      "max": 8,
      "min": 7
    }
  },
  "numCompletedRequests": 28,
  "plate": "XXXXXX",
  "tankSize": {
    "max": 17.2,
    "min": 17.2
  },
  "updatedAt": "2018-01-15T14:51:27.042Z",
  "year": 2015
},
"vehicleId": "XXXXX",
"verifyGallonRange": {
  "max": 15.302010495171167,
  "min": 9.39312361971691
}
}
“CryptoPort - Coin Portfolio tracker” is a popular and highly rated Android app used to monitor and manage cryptocurrencies across close to 20 exchanges. With CryptoPort, cryptocurrency investors can use a single API key and private access token to access all their cryptocurrency accounts on multiple exchanges. They can also see details of and manage all their accounts from a single location.

The MTT found that the app stored and exposed close to 150 different API keys with private access tokens to various exchanges (FIGURE 3), allowing a bad actor to easily drain cryptocurrency accounts potentially containing hundreds of thousands of dollars in cryptocurrency.
FIGURE 3

THE MTT FOUND THAT THE APP STORED AND EXPOSED CLOSE TO 150 DIFFERENT API KEYS with private access tokens to various exchanges allowing a bad actor to easily drain cryptocurrency accounts potentially containing hundreds of thousands of dollars in cryptocurrency.
},
"THB": {
"priceBtc": {
"change": 0,
"price": 0.0167062549485353
},
"symbol": "THB",
"total": 0.01,
"totalUsd": 0.00031670625494853525
},
"XZC": {
"priceBtc": {
"change": -20.16,
"price": 0.005256658693549241
},
"priceUsd": {
"change": -11.75,
"price": 53.23357086302454
},
"symbol": "XZC",
"total": 13.46771486,
"totalUsd": 720.8373115882819
},
"credential": {
"apiKey": "XXXXXXXX",
"secret": "XXXXXXXXX"
},
"deviceId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXX",
"exchangeId": "bxinth",
"exchangeName": "BX.in.th",
"totalBtc": 0.46087983813904715,
"totalChanged24HrUsd": -535.6000315900569,
"totalChangedPercent24HrUsd": -11.414308972960292,
"totalUsd": 4692.35617205436,
"walletTitle": "BX.in.th",
"walletType": "EXCHANGE"}
Conclusion

Our research shows that apps with open backend Firebase database servers are a common and serious data leakage threat, exposing many types of personal and enterprise data from thousands of iOS and Android applications and millions of users.

The impacts to enterprises of data exposures such as those related to mobile apps using unsecured Firebase databases can be huge, including bad publicity, potential fraud, theft of intellectual property and other corporate and personal data, financial losses, regulatory violations/penalties, larger breach enablement and much more.

To date, Apthority is the only security vendor researching and protecting against these large scale back-end data exposures. This is the third time within a year that we have uncovered a massive mobile data breach with sensitive personal and enterprise information. In our two previous Mobile Threat Reports in 2017: "Q2-2017 HospitalGown: The Backend Exposure Putting Enterprise Data at Risk" and "Q4-2017 Eavesdropper: The Mobile Vulnerability Exposing Millions of Conversations", we detailed other large-scale data exposures that threaten enterprises.
Recommendations

The Firebase vulnerability is the latest example of a security risk to the enterprise posed by the HospitalGown threat. These ongoing data breaches show that enterprises need visibility into vulnerabilities such as these to avoid non-compliance and leakage of private and sensitive data. To effectively mitigate your enterprise risk of wide-scale sensitive data exposure, we recommend the following strategy:

Ensure that your EMM published and company branded apps are secure

Most importantly, verify that both the private and public apps available to your employees through your EMM App Store, which often handles sensitive corporate and personal data, are secured against the HospitalGown threat and this Firebase variant. You’ll need to perform a thorough security review of internal apps developed by third parties, in-house developed apps, and public apps available for employee productivity. You may have difficulty achieving visibility into data exposed by this threat in EMM published enterprise and public apps without an automated MTD solution focused on app threats and backend vulnerabilities, such as Appthority Mobile Threat Protection.

Understand your organization’s exposure via personally downloaded apps on COPE and BYO devices and act accordingly

If you’re like many enterprises, employees may be free to download public apps onto their corporate-owned or BYO devices. Security issues on personally downloaded public apps often unknowingly introduce risk to corporate and personal data. Because so many employees use personally downloaded apps for business and productivity purposes, we highly recommend that you understand which personally downloaded apps present enterprise exposure from the HospitalGown threat and its variants and take actions to prohibit those deemed an enterprise risk due to the nature of the app (i.e. productivity or business app). Again, you can accomplish this most effectively with an automated MTD solution focused on app threats and backend vulnerabilities, such as Appthority Mobile Threat Protection.

Verify that both the private and public apps available to your employees through your EMM App Store are secured against the HospitalGown threat and this Firebase variant

Understand which personally downloaded apps present enterprise exposure from this threat and take actions to prohibit those deemed an enterprise risk due to the nature of the app (i.e. productivity or business app).
ABOUT APPTHORITY

Appthority is a pioneer in enterprise mobile security and the leader in the Mobile Threat Defense category. The comprehensive Appthority Mobile Threat Protection (MTP) solution helps customers keep their data private and secure from mobile device, app and network threats. More Fortune 1000 companies trust Appthority to secure their enterprises from mobile threats because Appthority delivers best-in-class mobile threat protection and unparalleled enterprise visibility and control of mobile risks. With Appthority, security teams are informed, employees are productive, and enterprise data is kept private and secure.

ABOUT THE APPTHORITY MOBILE THREAT TEAM (MTT)

The Appthority Mobile Threat Team (MTT) monitors and investigates mobile risks that pose a direct threat to mobile enterprises. Its goal is to provide research that educates and informs enterprises looking to protect their people, data, devices, apps, and networks from mobile risks. The MTT is comprised of top mobile security researchers and threat analytics managers who use their experience and expertise to develop best-in-class research insights. The team prides itself on delivering unique, accurate and practical perspectives, as well as security solutions, that help our enterprise audience understand and address the most impactful mobile threats.

ABOUT THE AUTHORS

Appthority Enterprise Mobile Threat Team members Mr. Kevin Watkins and Dr. Su Mon Kywe and were the lead researchers on this report.

Mr. Watkins, an Appthority Co-founder, is a mobile security specialist and heads the Data Science function at Appthority. He previously served as Research Architect at McAfee Labs where he designed mobile security research tools and is a frequent speaker at security conferences.

Dr. Kywe is a Research Scientist at Appthority and holds an Information Systems Ph.D. from Singapore Management University, among other academic credentials. She has published regularly in peer-review journals and is a frequent speaker at security conferences.