{* RESEARCH *}

Software developer cracks Hyundai car security with Google search

Top tip: Your RSA private key should not be copied from a public code tutorial

Thomas Claburn
Wed 17 Aug 2022 // 20:19 UTC

79

A developer says it was possible to run their own software on the car infotainment hardware after discovering the vehicle's manufacturer had secured its system using keys that were not only publicly known but had been lifted from programming examples.

An unidentified developer posting under the name "greenluigi1" wanted to modify the in-vehicle infotainment (IVI) system in his 2021 Hyundai Ioniq SEL.

To do so, he would have to figure out how to connect to the device and bypass its security.

After trying to figure out how to customize firmware updates for the IVI's D-Audio2 system, made by the car company's mobility platform subsidiary Hyundai Mobis, and have them accepted by the IVI, the unidentified car hacker found an unexpected way – through Google.

The IVI accepts firmware updates in the form of password-protected ZIP archives. This led to downloading an update ZIP from Hyundai's website and was able to bypass the simple password protection on the archive to access its contents, which included encrypted firmware images for various parts of the IVI.

The goal then became creating his own firmware images and encrypt them in a way within a ZIP that the car would accept, install, and run, thus allowing control of the hardware from the hacker's own supplied code.

As luck would have it, "greenluigi1" found on Mobis's website a Linux setup script that created a suitable ZIP file for performing a system update.

Turns out the encryption key in that script is the first AES 128-bit CBC example key listed in a NIST document

The script included the necessary ZIP password for the system update archives, along with an AES symmetric Cipher-Block-Chaining (CBC) encryption key (a single key rather than an RSA asymmetric public/private key pair) and the IV (initialization vector) value to encrypt the firmware images.

This information could also be used to decrypt the images.

That meant "greenluigi1" could use the AES key to decrypt the firmware images, modify them, and then use the script to re-encrypt the images using the AES key and package it all up into a password-protected ZIP for Hyundai's IVI update system to ingest.

But it wasn't going to be that easy: some part of the supplied data, at least, would need to be cryptographically signed using an RSA private key, and the car hacker didn't have it. The updater would use the private key's corresponding RSA public key to check the data was signed using the correct secret private key.

"greenluigi1" thus needed to find the RSA private key to get any further.

"The script hinted at RSA signing being used, but unfortunately the key used for that was not in the source code," the anonymous poster explained in a blog post back in May that was brought to our attention by a reader this week.

"Turns out the [AES] encryption key in that script is the first AES 128-bit CBC example key listed in the NIST document SP800-38A [PDF]".

As an aside, the consensus in the crypto community appears to be that CBC is difficult to use properly and other approaches are recommended. Microsoft last year warned that the use of AES CBC with padding may not be secure.

"Microsoft believes it's no longer safe to decrypt data encrypted with the Cipher-Block-Chaining (CBC) mode of symmetric encryption when verifiable padding has been applied without first ensuring the integrity of the ciphertext, except for very specific circumstances," the company said. "This judgment is based on currently known cryptographic research."

But Hyundai's fault isn't misimplementing AES CBC; it's using yet another key published online as a secret.

- Hyundai reveals the robotaxi it built for Lyft, and a version for its very own metaverse
- Driverless car first: Chinese biz recalls faulty AI
- Toyota, Subaru recall EVs because tires might literally fall off
- Car makers lock in long-term deals with chip giants for future autonomous vehicles

With that symmetric key, "greenluigi1" was able to extract the contents of one of the encrypted firmware image files within the update ZIP.

And within the extracted files, he found the software that handles IVI updates, a binary called updateAgent. In other words, the hacker was now looking at the very code – the update tool in the IVI in his car – that he or she needed to defeat, which presented a sporting chance to beat it.

"Since I already had the zip password and the encryption key, I decided to look for the signing key," explained "greenluigi1". "With any luck they left not only the public key but the private key too."

Luck held out, in a way. "Greenluigi1" found within the firmware image the RSA public key used by the updater, and searched online for a portion of that key. The search results pointed to a common public key that shows up in

online tutorials like "RSA Encryption & Decryption Example with OpenSSL in C."

That tutorial and other projects implementing OpenSSL include within their source code that public key and the corresponding RSA private key.

This means Hyundai used a public-private key pair from a tutorial, and placed the public key in its code, allowing "greenluigi1" to track down the private key. Thus he was able to sign Hyundai's files and have them accepted by the updater.

At that point, there was an opportunity to create custom firmware for the car's IVI, as described in two subsequent online posts. "greenluigi1" also has summarized his approach for other Hyundai owners.

Hyundai has not responded to a request for comment. ®

Similar topics

Automotive | Encryption | Security | Software

Narrower topics

2FA | AdBlock Plus | Advanced persistent threat | App | Audacity | Authentication | Black Hat | BMW | Bug Bounty | Common Vulnerability Scoring System | Cybercrime | Cybersecurity | Cybersecurity and Infrastructure Security Agency | Cybersecurity Information Sharing Act | Database | Data Breach | Data Protection | Data Theft | DDoS | Digital certificate | Electric vehicle | Exploit | Firewall | Ford | Hacker | Hacking | IDE | Identity Theft | Infosec | Jenkins | Kenna Security | Let's Encrypt | LibreOffice | Map | Microsoft 365 | Microsoft Office | Microsoft Teams | NCSC | OpenOffice | Palo Alto Networks | Password | Phishing | Programming Language | Quantum key distribution | Ransomware | Remote Access Trojan | Retro computing | REvil | RSA Conference | Search Engine | Signal | Software bug | Software License | Spamming | Spyware | Surveillance | TLS | Trojan | Trusted Platform Module | Visual Studio | Visual Studio Code | Vulnerability | Wannacry | Web Browser | Zero trust