Screenshots:

1. Amazon AWS ubuntu connection

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

2.Repackaging command execution procedure and date/time of the new packaging file

RSA Keypair Generation and signing the app:

A screenshot of a computer

AI-generated content may be incorrect.

Tried again and restarted signing setup using Oracle bash instead of Amazon AWS-

A screenshot of a computer

AI-generated content may be incorrect.

Signing:

A screenshot of a computer

AI-generated content may be incorrect.

3. Contacts before and after the reboot:

A screenshot of a computer

AI-generated content may be incorrect.

Before:

A screenshot of a computer

AI-generated content may be incorrect.

After the reboot:

A screenshot of a computer

AI-generated content may be incorrect.

**Questions:**

1. **In this project, the malicious APP deletes all contacts. Do you think an attacker can insert or substitute persons/phone number in the contact list? How can he do it? Show an example:**

Yes, if a malicious app is granted permission to access the contact list, it can easily insert fake entries or overwrite existing ones. This allows an attacker to impersonate someone the user trusts by swapping their contact info with the attacker’s phone number.

For instance, the attacker could use a repackaged app that secretly modifies contacts once installed. This could be used for phishing or tricking the victim into sharing private information with what appears to be a familiar contact. To do this, the attacker must ensure the app requests permissions like “READ\_CONTACTS” and “WRITE\_CONTACTS” in the AndroidManifest.xml. They can then write Java or Smali code that manipulates the contact list programmatically when the app runs.

Proof of modification: A screenshot of a computer screen

AI-generated content may be incorrect.

1. **Self-signing of the package actually contributes to the distribution of malicious apps. Do you think prevention of self-signing can eliminate the threat of malicious apps? Explain.**

Preventing developers from self-signing apps can reduce some risks, but it won’t completely eliminate the threat of malware. Self-signing allows anyone to publish an app without validation, which is how repackaged apps can be sideloaded onto devices.

However, even without self-signing, attackers can still find ways to spread malware, like using stolen signing keys or injecting code into trusted apps. Some devices also disable signature checks entirely, especially rooted or test setups.

Instead of banning self-signing, it’s more effective to combine approaches: enforcing app store checks, analyzing runtime behavior, and using tools like Google Play Protect or mobile device management. Educating users about sideloading risks is also key to better protection.

1. **To compromise a smart home network, an attacker can try to take over the control of either your cellphone or the Smart Hub. What are the advantages and disadvantages of these two approaches for the attackers?**

Attackers can target either a smartphone or a smart hub to take control of a smart home. Both approaches have unique benefits and drawbacks.

If the attack is on a smartphone:

* **Advantages**: Phones hold private data like messages, contacts, and banking info. It’s easier to attack them remotely through phishing or fake apps.
* **Disadvantages**: Phones aren’t always connected to the home network, and strong protections like biometrics and app permissions make attacks harder.

If the attack is on a smart hub:

* **Advantages**: Hubs are always on and connected, allowing long-term access. They can control all connected IoT devices quietly.
* **Disadvantages**: Hubs usually don’t store personal data and are designed with tighter firmware security. Also, they may need physical access to pair.

**Conclusion**  
Smartphones are typically easier to exploit and offer more personal data, so they’re the more common target. But as smart homes grow, both devices need strong security to stay protected. smartphones provide richer rewards (data and credentials) but are harder to maintain persistent access to. Smart hubs, while offering a quieter, more stable foothold in the network, are typically harder to exploit and provide narrower data scope.