Initial Access, Tactic TA0027 - Mobile

Archived: 2026-04-05 15:33:41 UTC

The adversary is trying to get into your device.

The initial access tactic represents the vectors adversaries use to gain an initial foothold onto a mobile device.

ID: TA0027

Created: 17 October 2018

Last Modified: 25 April 2025

Techniques

Techniques: 8

ID	Name	Description
T1661	Application Versioning	An adversary may push an update to a previously benign application to add malicious code. This can be accomplished by pushing an initially benign, functional application to a trusted application store, such as the Google Play Store or the Apple App Store. This allows the adversary to establish a trusted userbase that may grant permissions to the application prior to the introduction of malicious code. Then, an application update could be pushed to introduce malicious code.
T1456	Drive-By Compromise	Adversaries may gain access to a system through a user visiting a website over the normal course of browsing. With this technique, the user's web browser is typically targeted for exploitation, but adversaries may also use compromised websites for non-exploitation behavior such as acquiring an Application Access Token.
T1664	Exploitation for Initial Access	Adversaries may exploit software vulnerabilities to gain initial access to a mobile device.
T1461	Lockscreen Bypass	An adversary with physical access to a mobile device may seek to bypass the device’s lockscreen. Several methods exist to accomplish this, including:
T1660	Phishing	Adversaries may send malicious content to users in order to gain access to their mobile devices. All forms of phishing are electronically delivered social engineering. Adversaries can conduct both non-targeted phishing, such as in mass malware spam campaigns, as well as more targeted phishing tailored for a specific individual, company, or industry, known as

ID		Name	Description
			"spearphishing." Phishing often involves social engineering techniques, such as posing as a trusted source, as well as evasion techniques, such as removing or manipulating emails or metadata/headers from compromised accounts being abused to send messages.
T1458		Replication Through Removable Media	Adversaries may move onto devices by exploiting or copying malware to devices connected via USB. In the case of Lateral Movement, adversaries may utilize the physical connection of a device to a compromised or malicious charging station or PC to bypass application store requirements and install malicious applications directly. In the case of Initial Access, adversaries may attempt to exploit the device via the connection to gain access to data stored on the device. Examples of this include:
T1451		SIM Card Swap	Adversaries may gain access to mobile devices through transfers or swaps from victims' phone numbers to adversary-controlled SIM cards and mobile devices.
T1474		Supply Chain Compromise	Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise.
	.001	Compromise Software Dependencies and Development Tools	Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise. Applications often depend on external software to function properly. Popular open source projects that are used as dependencies in many applications may be targeted as a means to add malicious code to users of the dependency.
	.002	Compromise Hardware Supply Chain	Adversaries may manipulate hardware components in products prior to receipt by a final consumer for the purpose of data or system compromise. By modifying hardware or firmware in the supply chain, adversaries can insert a backdoor into consumer networks that may be difficult to detect and give the adversary a high degree of control over the system.
	.003	Compromise Software Supply Chain	Adversaries may manipulate application software prior to receipt by a final consumer for the purpose of data or system compromise. Supply chain compromise of software can take place in a number of ways, including manipulation of the application source code, manipulation of the update/distribution mechanism for that software, or replacing compiled releases with a modified version.