Rooting Malware Makes a Comeback: Lookout Discovers Global Campaign

blog.lookout.com/lookout-discovers-global-rooting-malware-campaign

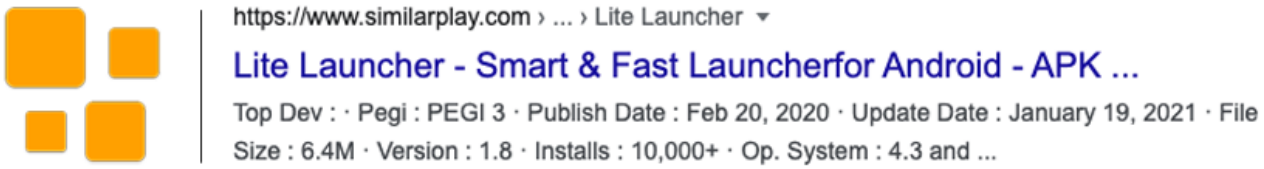
By Kristina Balaam, Paul Shunk

Security researchers at the Lookout Threat Lab have identified a new rooting malware distributed on Google Play and prominent third-party stores such as the Amazon Appstore and the Samsung Galaxy Store.

We named the malware "AbstractEmu" after its use of code abstraction and anti-emulation checks to avoid running while under analysis. A total of 19 related applications were uncovered, seven of which contain rooting functionality, including one on Play that had more than 10,000 downloads. To protect Android users, Google promptly removed the app as soon as we notified them of the malware.

This is a significant discovery because widely-distributed malware with root capabilities have become rare over the past five years. As the Android ecosystem matures there are fewer exploits that affect a large number of devices, making them less useful for threat actors.

While rare, rooting malware is very dangerous. By using the rooting process to gain privileged access to the Android operating system, the threat actor can silently grant themselves dangerous permissions or install additional malware — steps that would normally require user interaction. Elevated privileges also give the malware access to other apps' sensitive data, something not possible under normal circumstances.



"Lite Launcher," an app launcher replacement, is one of the AbstractEmu apps that appeared on Google Play. It had more than 10,000 downloads.

Who is the threat actor and what do they want?

While we don't know exactly who is behind AbstractEmu, we think the actors are a well-resourced group with financial motivation. Their code-base and evasion techniques — such as the use of burner emails, names, phone numbers and pseudonyms — are quite

sophisticated. We also found parallels between the malware and banking trojans, such as the untargeted distribution of their apps and the permissions they seek.

AbstractEmu disguised itself as a number of different apps: including utility apps, such as password managers, and system tools like app launchers or data savers. From left to right: Anti-ads Browser, Data Saver, Lite Launcher, My Phone, Night Light, All Passwords, Phone Plus.

Indiscriminate targeting

One of the major clues as to the threat actors behind AbstractEmu is based on the widespread, untargeted distribution of the apps. Of the 19 apps we found related to the malware, most of them were disguised as utility apps such as password or money managers, and system tools like file managers and app launchers. All of them appeared to be functional to the users. This includes “Lite Launcher” which had more than 10,000 downloads before it was taken off Play.

The types of vulnerabilities AbstractEmu takes advantage of also point to a goal of targeting as many users as possible, as very contemporary vulnerabilities from 2019 and 2020 are leveraged. One of the exploits used CVE-2020-0041, a vulnerability not previously seen exploited in the wild by Android apps. Another exploit targeted CVE-2020-0069, a vulnerability found in MediaTek chips used by dozens of smartphone manufacturers that have collectively sold millions of devices. As a hint to the threat actor’s technical abilities, they also modified publicly available exploit code for CVE-2019-2215 and CVE-2020-0041 in order to add support for more targets.

The way the AbstractEmu threat actor distributes these apps is also indiscriminate. In addition to Google Play, Amazon Appstore and Samsung Galaxy Store, we found them on Aptoide, APKPure and other lesser known app stores and marketplaces. In terms of promotions, we uncovered advertisements on social media and Android-related forums. While most were written in English, we did find one instance where the malware was promoted in Vietnamese. Though our telemetry showed that people in the United States were the most impacted, people from a total of 17 countries were victimized by AbstractEmu.

Parallels to banking trojans

In addition to the untargeted distribution of the app, the extensive permissions granted through root access align with other financially motivated threats we have observed before. This includes common permissions banking trojans request that provide them the ability to receive any two-factor authentication codes sent via SMS, or run in the background and launch phishing attacks. There are also permissions that allow for remote interactions with the device, such as capturing content on the screen and accessing accessibility services, which enables threat actors to interact with other apps on the device, including finance apps. Both of these are similar to the permissions requested by the Anatsa and Vultur malware families.

Beyond these, Mandrake was another financially motivated threat which had extensive spyware capabilities similar to those seen with AbstractEmu. By having complete insight into the device and its activity, the actors can tailor their attacks to the specific target and increase the likelihood of success.

Multilayer malicious flow

The threat actor behind AbstractEmu takes great lengths to ensure they evade detection — from the initial infection to the third stage of the infection. Each of the techniques aren't unique on their own, but when deployed as part of a campaign they indicate just how well-resourced the threat actor is.

AbstractEmu does not have any sophisticated zero-click remote exploit functionality used in advanced APT-style threats, it is activated simply by the user having opened the app. As the malware is disguised as functional apps, most users will likely interact with them shortly after downloading.

Initial infection: anti-emulation and device inspection

Beyond the legitimate functionalities of the trojanized apps lies a series of steps taken to ensure AbstractEmu isn't detected, which are activated as soon as the user opens the app. The first step is to check whether the infected device is a real device or is emulated. Similar to checks seen in an open source library EmulatorDetector, the malware will look at the device's system properties, list of installed applications and filesystem.

Once the device passes that initial analysis, the app will begin communicating with its command and control (C2) server via HTTP, expecting to receive a series of JSON commands to execute. Each app contains hard-coded commands that it supports. To decide which command to execute, the app will send a large amount of data to the C2 server, including both the commands it has support for, and device data such as the device's manufacturer, model, version and serial number, telephone number and IP address.

• Manufacturer, model, version, serial • IP Address • Wi-Fi/Bluetooth MAC addresses • Package name of app • Status of risky permissions/capabilities granted to the app	• SIM information: carrier name, number, IMEI/device ID • Timezone • Account information • App process ID • Command numbers supported by the app • Root status • Package name of installer of app

To decide on what further actions to take, AbstractEmu apps send a large amount of data to the C2 server.

Other information AbstractEmu's C2 server checks include whether the app has root access, which app was used to install the malicious app and whether the requested permissions and capabilities have been granted.

In total we found four supported commands embedded within these apps, though not all of the apps offer the same capabilities.

Command Number	Function
527	• Collect targeted files based on path information, regex patterns and blocklists sent by the C2 server • Track modified times of files and only collect new or updated content
576	• Collect contact information from the device, including name and number
668	• Collect device location, or as a fallback, Wi-Fi information
955	• Execute embedded root exploits against the device • Use root access to install a new app, grant it permissions and launch

We saw a total of four different types of JSON commands sent from AbstractEmu's C2 server, which are listed above.

The rooting process: the heart of the malicious flow

At the center of AbstractEmu's infection flow is getting root access to the Android device. By rooting the device, the malware is able to silently modify the device in ways that would otherwise require user interaction and access data of other apps on the device.

To ensure the process goes smoothly, the apps are embedded with hidden, encoded files used during and after the rooting process — including exploit binaries targeting different vulnerabilities. By default, these binaries are executed in a specific order, although the C2 server can change that order based on how the device is configured.

Exploit #	Architecture	Notes
1500	32-bit	iovyroot (CVE-2015-1805)
1501	64-bit	iovyroot (CVE-2015-1805)
2002	64-bit	CVE-2020-0069
2001	64-bit	CVE-2020-0041
1901	64-bit	Qu1ckr00t (CVE-2019-2215)
1502	32-bit	PingPongRoot (CVE-2015-3636)

By default, AbstractEmu malware attempts to execute these exploits in the order they are shown in this table. The C2 server can change that order based on the device's configuration.

In addition to these binaries, the apps also contain three encoded shell scripts and two encoded binaries copied from Magisk that are used during and after the rooting process. Magisk is a tool that allows Android users to acquire root access on their devices.

Two of the shell scripts are used to execute the exploit binary, gain root and then use elevated privileges to install the Magisk components for further root access. The newly installed Magisk components are used to execute the final shell script which first extracts an APK embedded in a binary to the device.

Then the package manager is used to silently install a new app and grant it a number of intrusive permissions, such as access to contacts, call logs, SMS messages, location, camera and microphone. In addition, the app will modify settings to grant itself risky capabilities or

reduce the device's security. With these capabilities the app can be used to conduct phishing attacks and provide the actor with all the information needed for illicit access to user accounts.

Risky permissions that grant access to:	Settings are modified to give app ability to:
• Contacts • Call logs • SMS messages • GPS • Camera • Microphone	• Reset the device password, or lock the device, through device admin • Draw over other windows • Install other packages • Access accessibility services • Ignore battery optimization • Monitor notifications • Capture screenshots • Record device screen • Disable Google Play Protect

The malware changes the device's settings and grants itself risky permissions, both of which make the device easier to target.

The "Settings Storage" App

The silently installed app is disguised as "Settings Storage" on the Android device. If the user tries to run the app, it will exit and open the legitimate settings app. The app itself does not contain any malicious functionality, which makes it harder to detect. Instead, it depends entirely on the files that its C2 server provides during execution.

At the time of discovery, the threat actor behind AbstractEmu had already disabled the endpoints necessary to retrieve this additional payload from C2, which has prevented us from learning the ultimate aim of the attackers.

Rare or not, always keep your OS up to date

While we weren't able to discover the purpose of AbstractEmu, we gained valuable insights into a modern, mass distributed rooting malware campaign, which has become rare as the Android platform matures.

Rooting Android or jailbreaking iOS devices are still the most invasive ways to fully compromise a mobile device. What we need to keep in mind — whether you're an IT professional or a consumer — is that mobile devices are perfect tools for cyber criminals to exploit, as they have countless functionalities and hold an immense amount of sensitive data.

To ensure you or your organization stay secure, we recommend diligently keeping your operating system up to date. Additionally, we recommend downloading apps from official stores only, as malware taken down from these stores may still be available elsewhere. Regardless of which store you use, always exercise caution when installing unknown apps.

Of course, you should also have dedicated mobile security software to secure against all mobile threats, including phishing, OS and app vulnerabilities, malware and network threats.

Indicator of Compromise

AbstractEmu APKs

(Download CSV file here)

title	package_name	sha1
		311e4c2b1d4b90664c56d8caa0d32035dde68cc6
		8716359ca3b4b7ed707e94b280e6e1e4c106035a
		0dddea2fc5d4d9e819d3f45b2673347a927e7cef
		60b9655d98d9dd697184e9b7d4026ef9ebc0bf05
All Passwords	com.mobilesoft.security.password	b3320a3b34fea23f7d402dc451667fb66214fb9f
		7e4c93c228d63f175b8b7232ab826b97dfbbd6b5
		7e263ba23e997ce5f4420f1e7de87305dc5eca6d
		84bef7fba1562df4aefcd552fd2b53b47c544427
		844e1de8d50cce29285d7a661141f8d93368702b
		935c7ee3dd5a0927352fde3cb91a2f1bf69719e3
Anti-ads Browser	com.zooitlab.antiadsbrowser	9caee5c9078cbcdcc2f5dcceb3cc60f8f57b94db
Data Saver	com.smarttool.backup.smscontacts	78820fdf4d81ecd2ac869be50211446257e17b66
		663f9102ce0e7b6d041efc9010a3afa70d8c1aaa
		99b7edc2af4e1c8dae3ee6f505ee771218e638fc
		96a207e41bdaac5fd5e74298a357f33fe343d93d
		c7d5b2cac0c9f65d40a7f8ed3f12b891fe21c5ed
		0afa18ff39419db788d0d6290f490e66513cf139
		d9eae350eb07f7f43e69f3c6c6dddc5d952e9de8
		2e074fa0c6de7092181c7b9284aa92c8c732d32a
		72b127983d70f79e366a2a1bc0b2d95af9e58d3f
Lite Launcher	com.st.launcher.lite	3e3eb8d0dfc57374e689fa7d24a0490be0aab3d1
My Phone	com.dentonix.myphone	43a910c44909583f0c0d690f3a24cba302e03432
		8108bcda08173ff6ee82a7b1ea1cd781364493d8
		50c98698c1af133a49eb7b2482246519913051ba
		44f705ac7f360671ba80232420dac81299c00394
Night Light	com.nightlight.app	e8e0905f98782027800e6ead9c0c6130d8822dac
Phone Plus	com.phoneplusapp	0ec2af45649b49a1bb807ae11d1db4b551a93d82

File hashes - Exploit Files

(Download CSV file here)

97f1b024317e6055817fec18e6435838ddda70cb 1eb7195b150b04e8629a2798654f2f3bdfb93309 a7d7d1e9fc0eabb575f6fd704c1e183e3268c7f8	iovyroot 32bit (CVE-2015-1805)
1bb0336f37efcc78334dac5544d20839445c524b aa1aa6cfbe8d6ea5e63ea8c2ed8fdbe917bab5c7 5eacf19bec89625fb6d2ceb1759a82bd36a06b1a	iovyroot 64bit
41bb8825197878154c885ac8360edecbd6f9962e 3f2fd895b16a82916c7632128d469db12f73b8a2 a5a844a6fa0e5fcedbe98220caedef8ffc5dab07	PingPongRoot (CVE-2015-3636)
afd49c170fe2bfa9fb1c99cf63e5c15861472e13 f60c4fc3c6f07b4810aae0c7c6ad17bbea8cccad 4ee775e43c200c87da05086607e64576daa0ae1a	Qu1ckr00t (CVE-2019-2215)
fb96f4bfcc758baf270081effcb6640fb680f602 9c3b3282ac4a19fa7ff51cdc03a18c3499f6ce67 3dbd3d933968b7626f2f8979a48807c75d370521	CVE-2020-0041
5e6b96ebc741ed41952c4c9e202d31951fe64240 c389e57890c9486ec532e0b295b4e0e52ac68677 fd965fb57da2ba6722e6ddce9c8be668bd14a903	CVE-2020-0069

File hashes - Rooting Tools

(Download CSV file here)

e5685281501b5376fbaf7be5a9143e56c3069390 9d1c997b9e59c71d0f481f03f3f1cd23cc387e48 bbaa0f0de247775db6fec005ce7b67ccff768c15 c930202c28dcf8efca7dc8b9fd37af3c7c2e3f90 fa3a330b19d6a8dd36df1f5136a7acf276aa3e8a	magiskinit binary
ca69bfd4619aa37b67fadffe05024dffda345795	"run" shell script
6ed71d247520745801e4c59158641f784ca4bf2d	"root" shell script
4cd5b1fe064223a913977e09c71f645b57edc1ba	final post-root script

Network IOCs

(Download CSV file here)

jobs[.]illaewinstralinc[.]com
outline[.]abunddhighett[.]com
tags[.]illaryboucnc[.]com
cloud[.]nathompsstra[.]com
store[.]dianmpsoathom[.]com
fluency[.]ryboucoathom[.]com
csa[.]naaronegya[.]com
tips[.]ghetaldhighe[.]com
color[.]joarteauxelb[.]com