Analysis of Recent Malware Incidents

1. Exposed Docker API Breaches Using the Tor Network

Overview:
A coordinated campaign is currently actively attacking publicly accessible and misconfigured Docker Remote API endpoints. The attack has matured from plain resource abuse to the basis of a sophisticated, anonymized botnet.

Attack Method:
The attack starts with an initial compromise in which attackers scan the internet for open Docker API ports (usually 2375). When discovered, they initiate a container creation request with a compromised Alpine Linux image that has a malicious shell command. For C2 communications and anonymity purposes, the malware container installs and starts a Tor daemon and directs all its communication through the Tor network for hiding the attackers' IP address and C2 server addresses. After starting Tor, the container downloads a second-stage script for deploying the payload through it. This script creates persistence, spreads the malware, and installs the main payload, which has evolved from cryptominers to more sophisticated botnet agents. The malware gains persistence and entrenchment through the addition of an attacker-controlled public SSH key to the authorized_keys file of the host and the installation of a cron job that denies external access to the Docker API port, shutting out rivals. Lastly, for propagation, the malware also contains utilities such as masscan to scan for other exposed Docker hosts and copies itself to increase the coverage of the botnet.

Mitigation and Resolution:
To prevent this, the first line of defense is securing API endpoints by ensuring that Docker API endpoints are not exposed to the public internet; if remote access is needed, it must be encrypted using TLS and strong authentication. Ensuring proper configuration of the firewall is also paramount, applying strict rules to restrict access to the Docker API port to trusted and authorized IP addresses only. The principle of least privilege must be enforced, using Docker containers with the fewest privileges necessary to minimize potential damage. Lastly, ongoing system monitoring is essential to monitor for unusual container activity, unfamiliar network traffic (particularly to Tor nodes), and extremely high CPU usage.

2. Phishing and Malware Distribution through Malicious SVG Files

Overview:
Cyber attackers are exploiting Scalable Vector Graphics (SVG) image files as a vector for malware spreading and phishing. By including scripts inside seemingly innocuous images, this

attack method is intended to evade standard security filters.
Attack Method:
This is an attack technique based on security scanner evasion since SVG images, being XML-based, are typically dealt with as plain images by email security gateways and antivirus software, enabling them to end up in the user inboxes without detection. Attackers embed obfuscated JavaScript inside the SVG XML architecture for the execution of embedded JavaScript. When a user opens the file using a contemporary web browser, the script runs automatically, resulting in phishing as well as payload delivery. The script can load a Base64-encoded HTML phishing page, like in an attack imitating the Colombian justice system, or initiate the background download of a malicious ZIP file. The attackers also employ polymorphism by slightly modifying each SVG document to evade hash-based detection signatures.
Mitigation and Resolution:
Proper mitigation and resolution involve a multi-layered strategy. User education training. Important to instruct users that SVGs may contain code which is executable and needs to be handled with utmost care. Organizations. Use sophisticated file analysis solutions with deep content inspection capability, such as VirusTotal's AI-driven Code Insight, to scan code in SVGs. Using file sanitization and Content Disarm and Reconstruction (CDR) technologies can remove active content. Such as scripts. From files prior to delivery. Finally, enforcing browser security policies like Content Security Policies (CSP) can limit or prevent the execution of inline scripts wherever possible.

3. Brokewell Android Malware Spammed through Fake Advertisements

Overview:
Brokewell is a full-featured Android banking trojan being spread via malvertising campaigns on popular social media networks. The malware provides attackers with comprehensive data-stealing capabilities and complete remote access to the infected device.
Attack Method:
The assault starts with a malvertising bait, when attackers place advertisements on Meta (Facebook) platforms that mimic legitimate, well-known apps. The campaign spoofed users with advertisements for a scam "free premium" version of the TradingView financial application. Clicking on the ad will send the users to a professionally replicated sideloading website, where they are required to install and download a malicious APK file, evading the protection of the genuine Google Play Store. The malware then resorts to accessibility service abuse, repetitively asking for access to this feature-rich Android feature. After permission, it exploits these permissions to auto-grant more permissions, thus ensuring its removal becomes challenging. The malware possesses wide-ranging malicious features, such as executing overlay attacks to extract credentials using misleading login windows, data exfiltration to extract session cookies, call logs, and device location, spyware features to record screenshots and audio, and Remote Access Trojan (RAT) features to give attackers a

real-time feed of the device screen and the capability to execute clicks, swipes, and other touch events remotely.

Mitigation and Resolution:

Mitigation practices target user actions and technical measures. Users have to be taught to download programs only from secure places such as the Google Play Store or the Apple App Store, and sideloading from websites must be prevented. It is also important to review app permissions; users need to be educated to thoughtfully check the permissions an app needs, with high-risk permissions for the Accessibility Service being a significant concern unless overtly necessitated. Technically, installing an established mobile endpoint security or Mobile Threat Defense (MTD) solution can identify and stop known malware. Last but not least, making sure that the Android operating system and all applications are updated with the most current security patches is crucial since native defenses such as Google Play Protect can hinder known copies of this threat.

Source:

https://www.bleepingcomputer.com/news/security/hackers-hide-behind-tor-in-exposed-docker-api-breaches/
https://www.bleepingcomputer.com/news/security/virustotal-finds-hidden-malware-phishing-campaign-in-svg-files/
https://www.bleepingcomputer.com/news/security/brokewell-android-malware-delivered-through-fake-tradingview-ads/