Incident Response: Investigating a Remote-Access Compromise

1. Overview

I responded to a compromised Windows PC that was the victim of a remote access scam. The attacker used a legitimate remote access tool, AnyDesk, to gain initial access and then installed a persistent malware toolkit that would survive a reboot and run with administrator privileges. I treated this as an incident response and forensics scenario. I identified the malicious files, authenticated those files, removed them, and related my actions to the NIST SP 800-61 incident response lifecycle.

The malware did not run with administrator privileges and the incident was resolved.

2. What Happened

The initial attack involved a remote access scam using AnyDesk to gain access to the computer. Once inside the computer, the attacker installed three malicious files including an executable, a key file, and a DLL. These files were configured to run again upon a reboot. The next step taken by the attacker was attempting to gain administrator privileges. This was presented as a series of User Account Control (UAC) prompts that were repeatedly displayed.

The repeated appearance of these UAC prompts was the first indication that something was wrong. I used these prompts as the indicator of compromise and built from there.

2.1 The Malware Toolkit

During my investigation I discovered one folder with three components of what appears to be a coordinated malware toolkit.

File	Type	Purpose and risk
setup.exe	Application	The primary launcher. This file is set to run on boot and attempts to gain administrator privileges, which presents as the UAC prompts
api_hook.dll	Application extension (DLL)	Used for API hooking. API hooking allows the malware to intercept information that passes through the operating system. This technique is often used to record keystrokes, collect login credentials, and monitor activity in browsers, such as when accessing banking services
setup.key	Key or configuration file	Contains the keys, instructions or settings required for the primary program to perform its intended functions

2.2 Evidence of timestomping

Upon examining the file properties I noticed an anomaly. The created date was the same as the date of the initial attack but the modified date indicated that the file was last modified three years prior. This is an example of timestomping (MITRE ATT&CK T1070.006). The attacker manipulated the metadata so that the files would appear to be old legitimate system files and therefore be able to evade antivirus and a quick user inspection of the files.

3. How I Responded and Investigated

- **Isolated the machine.** I isolated the host in order to prevent continued use by the attacker.
- **Identified the path.** The UAC warning included the path to the hidden malware.
- **Disabling persistence.** In the Startup Apps tab of the Task Manager I changed the startup entry for the malicious program from Enabled to Disabled. This prevented the program from automatically running on each reboot.
- **Authenticated the files.** I opened the folder containing the files and compared the date the program was created. I discovered the timestomping and confirmed that the files were malicious.
- **Checked the logs.** I accessed the raw MPLog logs in Microsoft Defender and analyzed the hexadecimal flags and result codes. For example a result of 0 (0x0) was interpreted as confirming the integrity of the system.
- **Removed the malware.** I removed the three files (setup.exe, api_hook.dll, and setup.key) and also emptied the Recycle Bin. Upon reboot the startup entry was gone and the UAC prompts had stopped. I then conducted full Microsoft Defender scans which returned no threats.

4. Impact Assessment

- **Active execution: blocked.** The user repeatedly denied (by clicking No) or closed the UAC prompt. This denied the program administrator privileges to deeper portions of the operating system.
- **Data theft by the malware: highly unlikely.** The malware was not initialized and remained stuck before execution on the hard drive. While the attacker did have live remote access for a period of time, I was unable to rule out other activity during this time. This is why the follow-up actions below are important.
- **The risk if it had run.** If someone had clicked Yes at any point during a reboot the DLL could have been injected into running processes and would have provided the attacker with a persistent backdoor and keylogging capabilities.

5. Framework Mapping

I related my actions to the NIST SP 800-61 incident response lifecycle.

Phase	What I did
Preparation and detection	Recognized the use of remote access software and used the repeating UAC prompt as the indicator of compromise
Containment	Contained the host
Eradication	Disabled the persistence trigger and removed the malicious files
Lessons learned	Documented the event and hardening steps

6. Why the Attack Failed

This attack used a technique known as Living off the Land. Rather than using an exploit the attacker used legitimate administrative tools, AnyDesk, to bypass perimeter security. Antivirus did not prevent this attack.

UAC prevented this attack. Since the administrator prompts were repeatedly denied the malware did not gain active execution. This is an example of defense in depth. When one measure fails another can still prevent the attack.

7. Outcome

The incident was resolved. The malware did not gain active execution because the administrator prompts were repeatedly denied. All three components were removed from the system and subsequent full Microsoft Defender scans returned no threats.

8. Going Further

Full Microsoft Defender scans returned no threats. To extend the investigation and be more confident that the system is completely clean the following steps would be taken:

1. **Trace the connection.** AnyDesk maintains a connection log. The trace file would contain information about when the scammer connected and for how long.
2. **Fingerprint the malware.** If a file hash exists in the Windows Defender history for the day the malware was installed searching that hash on VirusTotal would identify which malware family it belongs to.
3. **Run an offline scan.** A Microsoft Defender Offline scan examines the drive from outside of Windows. This makes it more difficult for stealth malware to hide from detection. It could also detect a second malware payload that had been dropped elsewhere.

4. **Verify system files.** Running sfc /scannow compares system files to known-good copies and repairs any that have been altered.

9. What I Would Do Differently

During the investigation I did not take screenshots. This write up relies on my notes and my incident report instead of screenshots. In forensics practice it is important to document a system before altering it. Once a system has been cleaned up there is no evidence left. I would:

- Take a screenshot of the UAC prompt, the startup entry and the file properties before altering them
- Record file hashes and timestamps so that they can be used later to verify integrity
- Save the AnyDesk connection log and the Windows Security protection history
- Quarantine the files instead of removing them immediately so they can be analyzed

10. What I Learned

The most important thing that I learned is how much a small detail can tell you. A repeating prompt led me to the path of the files and the timestamps on the files revealed that they had been manipulated. I also learned the importance of layered security since one control caught something that another did not. I documented the incident and hardening steps so that the next incident can be more easily detected and prevented.