Case Study

Threat Hunter finds renamed system utilities by file hash to uncover multiple attacks

Challenge

National architecture and engineering firm wanted to strengthen their security defense with Threat Hunting to proactively search for hidden threats and identify weakness in their posture

Results

Binary Defense Threat Hunting provides an additional layer of security by acting as an extension of the client's security team.

Summary:

Threat hunting is a vital but often misunderstood practice for organizations and security teams. Curiosity is one of the driving forces behind any successful threat hunt. This curiosity leads to a Threat Hunter creating a set of hypotheses that serve as the hunt's purpose and methodology. In this case, a threat hunters hypothesis leveraging file survey data, KQL queries on Microsoft Sentinel, and process file hashes uncovers anomalous events in the environment of a national architecture and engineering firm.

The Hypothesis:

Threat actors often attempt to disguise the execution of their malicious payloads by using built-in Windows system utilities rather than running a suspicious, never-before-seen program. Sometimes, the threat actor further disguises their attempts by re-naming the system utility to avoid detection by EDR tools that look for those utilities by name. However, even when a file is re-named, if the contents of the file have not changed, the file still has the same hash value. Hashes are mathematical representations of a summary of the file contents – if the bytes inside the file change even slightly, the file will have a completely different hash value.

This hypothesis aimed to find renamed system utilities, particularly regsvr32.exe and rundll32.exe, that are often used by threat actors to run malicious DLL files.

The Hunt:

With the hypothesis identified, the hunt begins. Making use of file survey data, the Threat Hunters built a set of KQL queries to identify process start events from files with the same hashes as system utilities, but different file names. During the investigation into these file hashes the Threat Hunters identified a suspicious execution event on an individual workstation. With a thread to pull on, the Threat Hunters began their investigation and discovered a series of related suspicious events to tell the complete story.

Retracing the event data on the compromised workstation, the Threat Hunters found that a user downloaded a legitimate looking .zip file and extracted compressed files. Within the .zip file was an .ISO disk image file containing a shortcut file to execute. When the user executed the shortcut, a batch script ran that made a copy of the legitimate system utility regsvr32.exe, renamed it, and deployed its payload as a .DLL file renamed to have a ".DAT" file extension. To the unsuspecting user of the

workstation, it appears that nothing had happened at all.

At this point in the hunt, the client was informed about the compromised workstation and began their own investigation. The client found that Microsoft Defender for Endpoint had not alerted on the rogue execution and would not have discovered the compromise without the hunt activity.

The Action:

With the hunt completed and an incident identified, the client began their internal Incident Response process with assistance from the larger Binary Defense Threat Hunting team. During the IR process, Binary Defense Threat Hunters identified another incident a few days earlier on the same system, following similar but not identical methods. The Threat Hunting team provided a thorough report in real-time as the investigation unfolded, answering the key questions of what happened, how it happened and what was affected. Additionally, the team wrote a query for Microsoft Defender for Endpoint to identify when .ISO files are extracted from .zip files.

The Takeaway:

Organizations like this national architecture and engineering firm turn to Binary Defense Threat Hunters for this exact expertise. Binary Defense Threat Hunters are constantly researching new potential attack vectors and vulnerabilities in common environments and utilizing this research to form threat hypotheses. These hunts help provide visibility to unknown environments and processes but allow Threat Hunters to build detections for weak spots in customers' detection capabilities – even if an active threat isn't necessarily found. In this hunt, the team uncovered a potential threat incident in our client's environment and helped mitigate a potential attack – a huge win for both Threat Hunters and our client.