Threat Hunting Case Study: Tracking Down GootLoader

intel471.com/blog/threat-hunting-case-study-tracking-down-gootloader

One of the ways users become infected with malware is through search engine optimization (SEO) poisoning. This technique is a net that entraps groups of users searching for certain terms. By either seeding legitimate sites with malware or creating misleading websites, attackers can lure people into downloading malicious code. One group of threat actors that has effectively used SEO poisoning to infect computers is behind the malware known as GootLoader. GootLoader is loader malware that appeared in late 2020. In its early days, GootLoader delivered the Gootkit banking trojan to facilitate account takeover (ATO). However, Gootkit is seldom seen anymore. GootLoader's mission has diversified, and its operators have shifted to an initial access broker (IAB) model. Access brokering is part of cybercrime-as-a-service, featuring threat actors who specialize in gaining access to systems and who then sell that access on to other threat actors. Those access buyers then exploit those computers, whether for data theft, ransomware or other schemes. GootLoader's access has also been used to install tools for reconnaissance and lateral movement, including Rubeus, SharpHound, SystemBC and Cobalt Strike, a legitimate penetration testing framework that is abused by threat actors.

In early 2023, GootLoader was distributed on sites that would show up in search results when looking for terms such as “agreement,” “contract,” “form,” “law,” “license” and “template.” To create these malicious links, GootLoader's operators would look to leverage vulnerabilities in websites and forums using the WordPress content management system (CMS). This allowed the threat actors to add new servers to their network, increasing their distribution and chances that users will encounter a GootLoader-seeded site in a search.

GootLoader remains a pervasive threat to organizations. Due to its stealthiness, effectiveness and exploitation in the wild by a number of ransomware campaigns, it is important that teams assess and prepare for this loader’s capabilities. Early detection and removal of a GootLoader infection can mean avoiding a data breach or ransomware attack. This post will discuss how to use Intel 471’s HUNTER platform to look for clues of GootLoader infections.

To begin threat hunting, we first need to collect current tactics, techniques and procedures (TTPs) GootLoader uses. These TTPs can originate from a variety of sources, such as vendor reports and data shared by independent researchers. The TTPs comprise current behaviors the malware is using and are less likely to be changed by the malware’s operators.

For this threat hunting example, we will collect TTPs from the DFIR Report, a group of researchers who publish about their investigations into malware infections, ransomware incidents and data breaches. In February 2024, the DFIR Report published “SEO Poisoning to Domain Control: The GootLoader Saga Continues.” The post discusses an incident at an organization that originated with a user conducting a search for “Implied Employment Agreement.” One of the sites returned was a compromised website mimicking a forum that hosted GootLoader. The person followed a link and downloaded what purported to be an employment agreement but was actually GootLoader.

The DFIR Report contains details about GootLoader’s execution:

Scrolling down to the “Persistence” section of the DFIR Report gives more information about scheduled tasks, which is one of GootLoader’s recurrent behaviors, and the evidence that it leaves behind. We can also identify the users who executed GootLoader and the trigger that the scheduled task was scheduled to execute on. If we look under LogonTrigger, we can see it is enabled and can conclude that the scheduled task will trigger when a particular user logs on to the machine (the UserID has been redacted):

What other TTPs can we find? We can see GootLoader runs other actions or commands, including scheduled tasks in unexpected locations. Adversaries love to tuck malware into unlikely locations where users pay little attention. When was the last time you stored an important document in the appdata/roaming directory, which is normally a hidden folder? This is an area to focus on.

We can pay less attention to arguments or file names used by an adversary. If an adversary is smart, the same file name not will not be repeated, as it’s an indicator of compromise (IoC) that can be easily changed for a new attack on a new target. By focusing on the behavior (scheduling a task) and the location (an odd directory), it’s possible to repeatedly catch GootLoader as opposed to concentrating detection efforts using an atomic or point-in-time artifact, such as a file name, hash, IP address, etc.

We don’t have to rely on external open source reports, however, to understand the TTPs of GootLoader. GootLoader is one of dozens of malware families tracked by Intel 471’s Malware Intelligence team. The team tracks malware families and infection campaigns and emulates malware to understand new behaviors. This intelligence can then be applied to threat hunting. Since Intel 471 acquired Cyborg Security in May 2024, we have been using data collected about malware such as GootLoader to write threat hunting packages. These packages, which are part of the HUNTER platform, are pre-written queries that are applicable to searching for possible signs of an infection within a wide range of endpoint detection and response (EDR), extended detection and response (XDR), logging and security information and event management (SIEM) platforms.

The advantage of the pre-written aspect for security teams is it allows threat hunters to focus on executing current hunts based on fresh intelligence from recent malware campaigns rather than researching and writing the queries. Since we have access to Malware Intelligence, we can improve the relevancy of our threat hunt packages to more closely match the current threat environment and lessen dependence on aging open source intelligence. This is not to slight or discount the value of open source intelligence. It’s extremely valuable, and sharing by the threat intelligence community strengthens the defenses of all. However, open source intelligence is also available to threat actors who may read the same blogs, social media feeds and vendor reports. Some actors may subsequently make changes to avoid detection.

For those who are not Intel 471 customers yet, we’ve made a GootLoader hunt query available as part of the Community Edition of HUNTER, which is free.

The threat hunt titled “Scheduled Task Executing from Abnormal Location” is available in the HUNTER Community Edition.

Earlier, we referred to GootLoader’s tendency to hide scheduled tasks in abnormal locations. One of the threat hunts included in the HUNTER Community Edition is titled “Scheduled Task with Abnormal Location in Details.” If we click on the hunt package, we can see that this type of threat hunt is not only applicable to GootLoader, but also other types of malware, including the Spectre remote access trojan (RAT), the Lumma information stealer and various ransomware strains including MedusaLocker, Nokoyawa, Quantum and LockBit 3.0.

If we scroll down to the query logic table, we can identify the field-value relationships that we will be analyzing in certain fields (see: screenshot below). First, we have a scheduled task, which is event_id 4698. This is the native Windows logging event ID that captures scheduled tasks.

Next we can look at the message field, which captures information in the native Windows event log. These are locations where GootLoader potentially may be active. Going further down, we have excluded \Windows Defender\ and \Microsft\Windows\Applications. Those are locations that ScheduledTasks often reference, which could result in an overwhelming number of false positives.

To see how this hunt works in practice, let's switch to Splunk. The data that is visible in the screenshot below reflects the logic of the query.

Here we have task arguments containing users, which checks off in the task arguments field that we created by using some regular expression. We also see the users and the event code 4689 specifically. Additionally, we see the task arguments contain a batch (.bat) file that's running in the \Users\James Murphy\AppData\Local\Temp directory. We see the command contains command.exe, and the task name is \DailyBackup.

There are multiple routes that can be taken next based on this data. If this is completely abnormal in your environment, it may be time to alert the incident response and digital forensics team and let them know a computer is likely compromised. Another route would be to pivot from the scheduled task to process creation and figure out if the file executed. It would be possible to write another query for process creation events to see if AutoLogoff.bat exists in any command-line arguments or parent command-line arguments.

We hope this post furthers an understanding of GootLoader and how this malware can be proactively hunted for in an environment. Sign up for a free HUNTER Community Edition account to maintain an edge in threat hunting activities. For more information about HUNTER, please contact Intel 471. Stay safe and as always, happy hunting.