Incident Report: Gootloader

Summary:

This report details a Gootloader payload distribution attack, as documented by the DFIR Report. The attackers compromised websites to host malware and used Search Engine Optimization (SEO) poisoning techniques to lure victims into downloading and executing malicious files. The attack involved the use of JavaScript, PowerShell, and Cobalt Strike payloads to establish persistence, perform reconnaissance, and move laterally within the compromised network.

Attack Details:

The attack began with the threat actors compromising websites to host malware and launching an SEO poisoning attack to move the compromised websites hosting malware to the top of search results. When a user searched for specific keywords and clicked on a compromised website, a forum-looking page was displayed. The user then clicked on a link, which downloaded a malicious zip file containing a JavaScript file.

Upon executing the JavaScript file, a Gootloader DLL was dropped on the system. The DLL was then executed using rundll32.exe, which made a Windows Management Instrumentation (WMI) call for process injection.

The malware created two registry keys, one populated with an encoded Cobalt Strike payload and the other storing a .NET loader named "powershell.dll." A PowerShell command was issued to extract the .NET loader from the registry and execute the code in memory via Assembly.Load().

The encoded PowerShell script also created a scheduled task that executed when the user logged on to the computer. The .NET loader decoded a Cobalt Strike payload from the registry and loaded and executed the payload using rundll32.

The Cobalt Strike payload performed various reconnaissance activities, including:

- Account Discovery using Bloodhound
- Domain Trust Discovery using Bloodhound
- Group Policy Discovery using Bloodhound
- Security Software Discovery using WMI

The attackers then used PowerShell to pivot to a nearby workstation and deployed a second Cobalt Strike payload to initiate a second C2 server. They disabled Windows Defender logging on the workstation and used PowerShell to drop additional malware, including a script named "mi.ps1," which is a renamed version of Mimikatz used for credential harvesting.

Using the harvested credentials, the attackers moved laterally to the Domain Controller and the File Share Server using RDP. They executed LaZagne to retrieve saved credentials and utilized SMB to transfer Cobalt Strike payload executables to three additional workstations.

On the Domain Controller, the attackers ran an Advanced IP Scanner to perform network service discovery and accessed file server SMB shares. They then used RDP to connect to the Backup Server and File Server to search for additional files of interest.