SOPHOS NEWS

Atom Silo ransomware actors use Confluence exploit, DLL side-load for stealthy attack

A new ransomware operator uses stealthy techniques, but borrows heavily from other players.

Written by Sean Gallagher, Vikas Singh

OCTOBER 04, 2021

SOPHOSLABS UNCUT THREAT RESEARCH ATOM SILO FEATURED LOCKFILE RANSOMWARE

Sophos' MTR Rapid Response team recently investigated a ransomware attack by a recently emerged threat actor group called Atom Silo. The sophisticated attack, which took place over two days , was made possible by an earlier initial access leveraging a recently revealed vulnerability in Atlassian's Confluence collaboration software.

While the ransomware itself is virtually identical to LockFile, the intrusion that made the ransomware attack possible made use of several novel techniques that made it extremely difficult to investigate, including the side-loading of malicious dynamic-link libraries tailored to disrupt endpoint protection software.

The incident offers evidence of how dangerous publicly disclosed security vulnerabilities in Internet-facing software packages can be when left unpatched even for a relatively short period. Concurrent with the ransomware attack, Sophos responders found that the Confluence vulnerability had also been exploited by a crypto miner.

The investigation by the Rapid Response team offered us an opportunity to perform the first in-depth look at the tools, techniques and practices of the Atom Silo group. The actors deploying the ransomware used well-worn techniques in new ways, and made significant efforts to evade detection prior to launching the ransomware.

Compromise and lateral movement

The first stage of the intrusion took place on September 13, 11 days before the ransomware attack unfolded. The intruder (either the Atom Silo actors themselves, an affiliate or an initial access broker) gained initial access through a Confluence server via an Object-Graph Navigation Language (OGNL) injection attack. This code injection on the Confluence server provided a backdoor, via which the attacker was able to drop and execute files for another, stealthy backdoor.

The payload dropped for the second backdoor consisted of three files. One of them was a legitimate, signed executable from a third-party software provider that is vulnerable to an unsigned DLL sideload attack.

The malicious DLL spoofs a library required by the executable and is placed in the same folder on the targeted server as the vulnerable .exe. This attack technique, known as DLL search order hijacking (ATT&CK T1574.001), is a well-worn technique recently observed in LockFile ransomware attacks leveraging the ProxyShell vulnerability.

The DLL's main role is decrypting and loading the backdoor from the third file, **mfc.ini**. The loaded code then connects to one of several stored hostnames (in this case, update.ajaxrenew[.]com) over TCP/IP port 80. The code appears similar to that of a Cobalt Strike Beacon. Once loaded, the backdoor allowed for remote execution of Windows shell commands through the Windows Management Interface (WMI), in the style of SecureAuth Corp.'s WMIexec penetration testing tool.

From this point, the intruder began lateral movement. Within five hours, they had compromised several additional servers. Using a single compromised administrative account, the attackers copied and executed the backdoor binaries using WMI.

At least some of the work was done using a self-deleting Windows batch file named **howtorun.bat**. The backdoor files were copied and launched on each additional server; in at least one case, the backdoor was then installed on the targeted servers as a service named "WindowsUpdate," executed using the vulnerable legitimate executable, to provide persistence. But otherwise, backdoors were executed directly from WMI.

On the third day of the intrusion, via a batch file named logs.bat, the intruder did some additional discovery, fetching information from the security logs to check for user logons and logoffs, account lockouts, the assignment of special privileges to a logon, and use of sensitive privileges. They also collected more information about the local network, compressing it into a zipped file for exfiltration.

During this time, the unrelated coin miner malware was implanted by another actor using the Confluence vulnerability.

Exfiltration and effect

On September 24, the ransomware actors began their own discovery and exfiltration efforts, checking the local volumes attached to an important server and then checking its history of Remote Desktop sessions. Using RDP, the ransomware gang then went hands-on-keyboard, dropping and executing the RClone utility to copy data off the server to a Dropbox account from several directories. The process was repeated on another server.

Soon after the exfiltration was complete, the intruders connected to the domain controller and dropped their all-in-one attack executable. There were two variants of the attack executable used in the attack; both executables carry the following files packed as resources:

- autoupdate.exe (the ransomware, detected as Troj/Ransom-GKL).
- autologin.exe, a Kernel Driver Utility hacktool (detected by Sophos as ATK/KDUtil-A).
- autologin.sys, a driver targeting Sophos services, including the file scanning service (now detected as Troj/KillAV-IT).
- drv64.dll, a Kernel Driver Utility hacktool database (detected by Sophos as ATK/KDUtil-A), previously reported as part of a LockFile ransomware attack using the PetitPotam exploit.

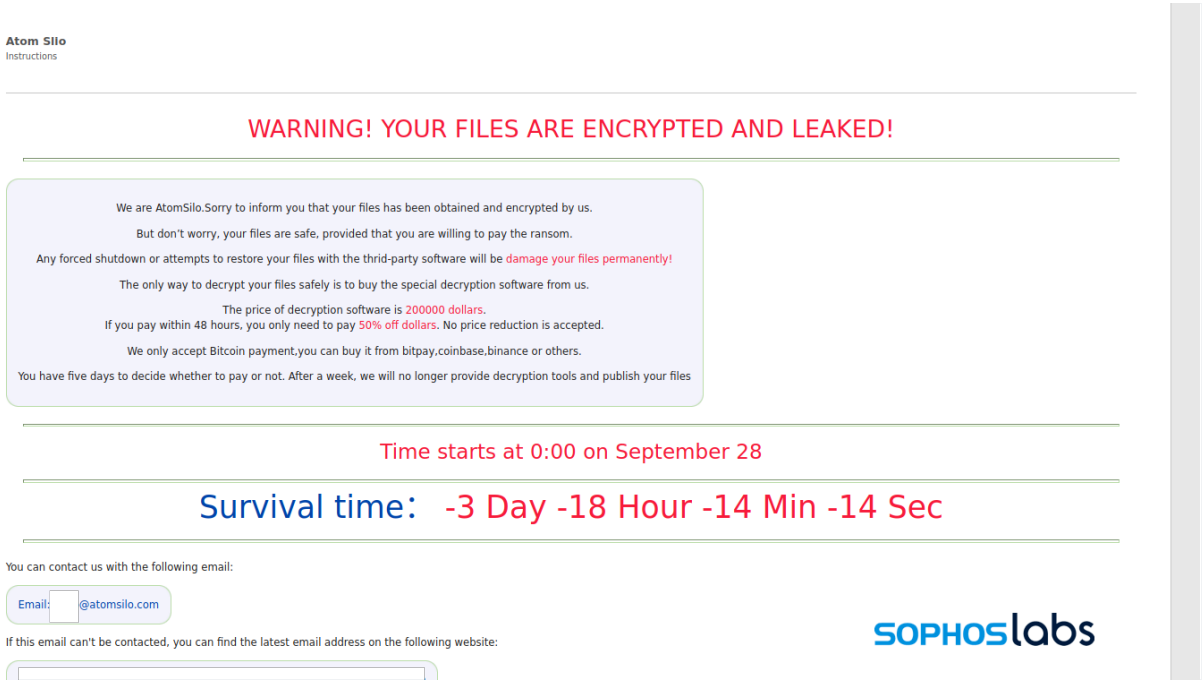
The attackers used autologin.exe to “map” the autologin.sys driver to the kernel, using the database of vulnerable drivers to find an available exploit. Once loaded, autologin.sys was able to bypass protections against shutting down endpoint protection services.

The attack executables performed a search for all the domain controllers on a network. They then push the four payload files to the \netlogon folder of each domain server, and create a batch file (autologin.bat or autologin1.bat, in this case). The main difference between the two is how that batch file is deployed — the first used in the attack (named **FuckGPO.exe** in this case) created a scheduled task XML file to carry out the attack, while the second (**2.exe**) creates a service that establishes persistence. The first version is executed on the domain controller; the second is launched as a service called "Update" on all domain-connected systems via a Global Policy Object.

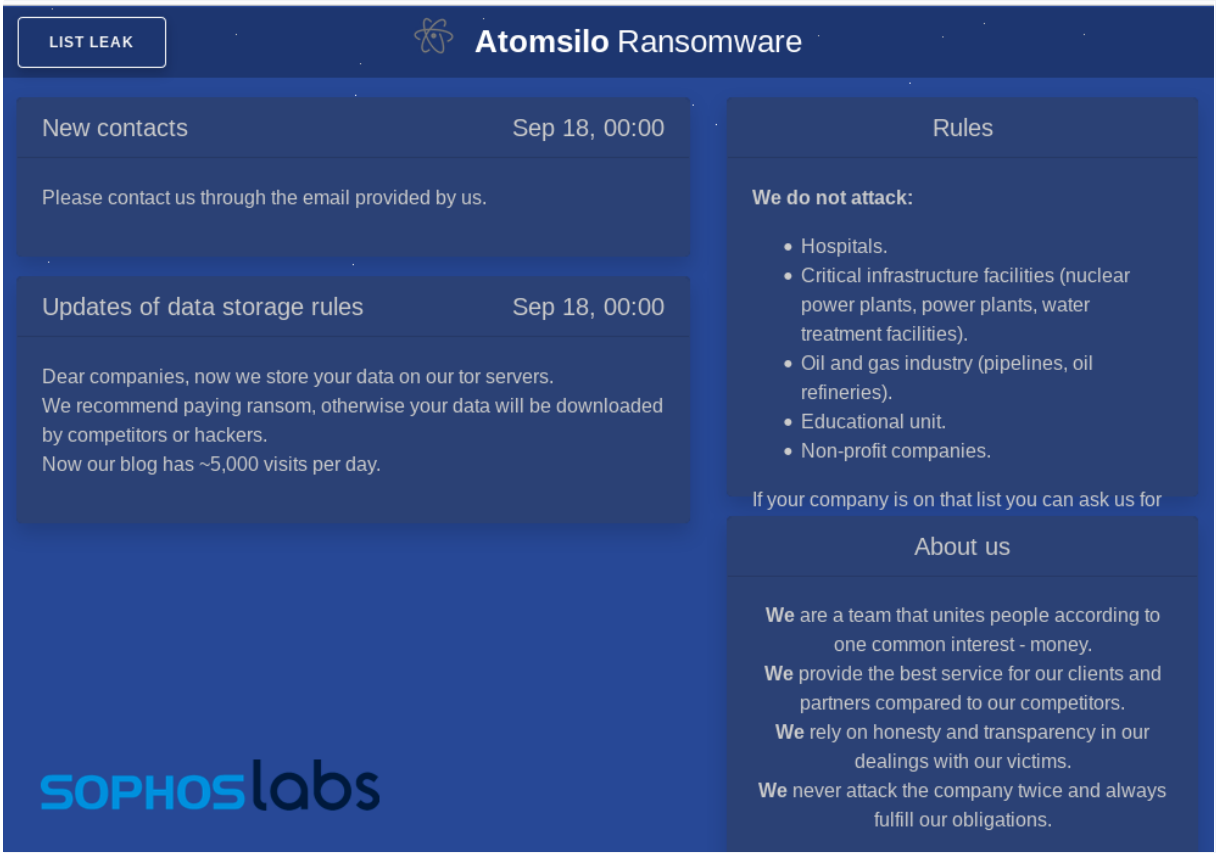
The batch file executed the kernel driver to disrupt endopoint protection, and then launched the ransomware.While the attack file triggered alerts when dropped on the domain server, the ransomware actors used the kernel hacktool to disable Sophos' file scanning service. This generated an alert as well, but further detection and quarantining of files was disabled.

Intercept X's CryptoGuard detected the ransomware , but the attackers then subsequently used the secondary attack executable, dropped into the Desktop folder, to disrupt the protection and again launch the ransomware with an updated GPO.

The ransomware executable itself connected to a remote URL (hxxp://139[.]180[.]184[.]147:45532/fake.php), and started encrypting files in a similar fashion to LockFile, adding a .ATOMSILO extension to encrypted files. The ransomware dropped a ransom note formatted in HTML, with instructions on how to contact Atom Silo's operators.



The Atom Silo ransom note.



The Atom Silo “blog.” The group has “rules” similar to DarkSide and BlackMatter.

All communications with the operators are over email through an account on atomsilo.com. That domain is registered through NameSilo LLC; the mail server is hosted in Hong Kong by the hosting provider Dataplugs.

The patch pacing problem

The initial point of compromise in this attack was a vulnerability that was only public for about three weeks at the time. For many organizations, keeping up with the pace of patching can be a challenge in the best of times—and the effects of lock-down and other recent stressors affecting staff availability are only making keeping up with patches more difficult.

Ransomware operators and other malware developers are becoming very adept at taking advantage of these gaps, jumping on published proof of concept exploits for newly-revealed vulnerabilities and weaponizing them rapidly to profit off them—as demonstrated by the evidence of two separate threat actors finding and exploiting the vulnerable Confluence server involved in this incident. If the ransomware attack had not been discovered, the cryptocurrency miner on the server may have gone undiscovered.

To reduce the threat, organizations need to both ensure that they have robust ransomware and malware protection in place, and are vigilant about emerging vulnerabilities on Internet-facing software products they operate on their networks. Shifting some products to vendor-hosted software-as-a-service may mitigate some of these risks, as vendors typically patch vulnerabilities in their own deployments of software faster than they can be deployed by on-premises customers, but this is not a panacea.

Additionally, abuse of legitimate but vulnerable software components through DLL side-loading and other methods has long been a technique used by attackers with a wide range of capabilities, and it has filtered down to the affiliates of ransomware operators and other cybercriminals. While abuse of some of these legitimate, signed components is well-enough known to defend against, the supply of alternative vulnerable executables is likely deep. Spotting legitimate executables that exist outside of the context of the products they are supposed to be part of requires vigilance—and vulnerability disclosure by the vendors they come from.

Under these conditions, organizations' best defenses include full deployment of malware protection on servers and endpoint devices, and that products are monitored to catch attacks that trigger detections or alerts before an attacker with administrative access can defeat protections. (Sophos products, including Intercept X, provide a number of protections against these types of attacks.) Organizations are also encouraged to have effective data backup practices and business continuity plans, regardless of their size, to ensure that they can survive attacks that leverage rapidly exploited vulnerabilities such as those revealed in Confluence and Microsoft Exchange this year.

A list of indicators of compromise for this attack is available on the SophosLabs Github page.

SophosLabs would like to acknowledge Bill Kearny of Sophos MTR's Rapid Response team, and Kajal Katiyar, Chaitanya Ghorpade and Rahil Shah of SophosLabs for their contributions to this report.