THREAT BULLETIN

# AR20230124-002 [Advisory Report] Vice Society Threat Group Wreaks Havoc In Education And Rail Sectors

TLP White

DESCRIPTION

## Summary

Vice Society, which first appeared in summer 2021, is a hacking group that has been targeting school systems in the US and UK and most recently the Bay Area Rapid Transit (BART) rail system in San Francisco. Vice Society has posted documents and files from these public sector organizations they have obtained through their activities. Some of the most recent leaked information includes contract data, W 9 forms, passport photos, and SSNs from the school systems as well as sensitive information from cases that the BART transit police department is investigating. Vice Society has used numerous ransomware variants as well as exploiting known vulnerabilities for access to systems and networks.

## Technical Details

Vice Society is an intrusion, exfiltration and extortion hacking group most known for ransomware attacks on educational and healthcare organizations. These threat actors are believed to be Russian-speaking and have used versions of Hello Kitty and Zeppelin ransomware, but are likely to deploy other variants in the future. They also use known vulnerabilities that have not been patched by system owners to obtain access and persistence.

Vice Society threat actors likely used compromised account credentials obtained through phishing campaigns or purchased from the dark web to exploit internet-facing applications that these public sector organizations use. Once access is gained to the targeted system or network, the attackers are looking for ways to escalate their privileges to access larger amounts of sensitive data which they plan to publicly release if the victim refuses to pay the ransom. If escalation is successful, the attackers can then access or create domain administrator accounts, change passwords, and design and run scripts which prevent victim remediation. Moving laterally in the targeted networks is also amongst the goals of these attackers. Vice Society has been observed using well known tools like PowerShell, SystemBC and CobaltStrike for lateral movement.

Vice Society has also been observed exploiting the PrintNightmare vulnerability for privilege escalation. Numerous Tactics, Techniques and Procedures (TTPs) are utilized by these threat actors to maintain persistence including leveraging scheduled tasks and creating undocumented autostart Registry Keys so that their access remains. While most organizations have software or tools that try to find this type of activity, threat actors use evasion techniques. These techniques include actions to defeat dynamic analysis, masquerading their tools and malware to make them look like legitimate files and activity, as well as malicious code injection.

Vice Society and other similar organizations favor striking these data rich and generally resource poor public sectors at such times when they know security staff will be limited such as over holiday periods, weekends or overnight hours as they know response times are slower during these times

### Tactics and Techniques

See the ATT&CK® for Enterprise framework for all referenced threat actor tactics and techniques.

- https://attack.mitre.org/versions/v11/techniques/T1190/ - Exploit Public-Facing Application
- https://attack.mitre.org/versions/v11/techniques/T1078/ - Valid Accounts
- https://attack.mitre.org/versions/v11/techniques/T1486/ - Data Encrypted for Impact
- https://attack.mitre.org/versions/v11/techniques/T1068/ - Exploitation for Privilege Escalation
- https://attack.mitre.org/versions/v11/tactics/TA0010/ - Exfiltration
- https://attack.mitre.org/versions/v11/techniques/T1053/ - Scheduled Task/Job
- https://attack.mitre.org/versions/v11/techniques/T1047/ - Windows Management Instrumentation
- https://attack.mitre.org/versions/v11/techniques/T1543/003/ - Modify System Process: Windows Service
- https://attack.mitre.org/versions/v11/techniques/T1547/001/ - Boot or Logon AutoStart Execution: Registry Run Keys / Startup Folder
- https://attack.mitre.org/versions/v11/techniques/T1547/002/ - Boot of Logon AutoStart Execution: Authentication Package
- https://attack.mitre.org/versions/v11/techniques/T1036/ - Masquerading
- https://attack.mitre.org/versions/v11/techniques/T1055/ - Process Injection

### Detections

- Exploit Public-Facing Application
  Application Log Content - Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Web Application Firewalls may detect improper inputs attempting exploitation.
  Network Traffic Content - Use deep packet inspection to look for artifacts of common exploit traffic, such as SQL injection strings or known payloads.
- Valid Accounts
  Logon Session Creation - Monitor for newly constructed logon behavior that may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Correlate other security systems with login information (e.g., a user has an active login session but has not entered the building or does not have VPN access).
  Logon Session Metadata - Look for suspicious account behavior across systems that share accounts, either user, admin, or service accounts. Examples: one account logged into multiple systems simultaneously; multiple accounts logged into the same machine simultaneously; accounts logged in at odd times or outside of business hours. Activity may be from interactive login sessions or process ownership from accounts being used to execute binaries on a remote system as a particular account.
  User Account Authentication - Monitor for an attempt by a user that may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion.
- Data Encrypted for Impact
  Command Execution - Monitor executed commands and arguments for actions involved in data destruction activity, such as vssadmin, wbadmin, and bcdedit
  File Creation - Monitor for newly constructed files in user directories.
  File Modification - Monitor for changes made to files in user directories.
  Process Creation - Monitor for newly constructed processes and/or command-lines involved in data destruction activity, such as vssadmin, wbadmin, and bcdedit.
- Exploitation for Privilege Escalation
  Driver Load - Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of the processes. This could include suspicious files written to disk, evidence of Process Injection for attempts to hide execution or evidence of Discovery. Consider monitoring for the presence or loading (ex: Sysmon Event ID 6) of known vulnerable drivers that adversaries may drop and exploit to execute code in kernel mode. Higher privileges are often necessary to perform additional actions such as some methods of OS Credential Dumping. Look for additional activity that may indicate an adversary has gained higher privileges.
- Scheduled Task/Job
  Command Execution - Monitor executed commands and arguments that may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code.
  Container Creation - Monitor for newly constructed containers that may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code
  Scheduled Job Creation - Monitor newly constructed scheduled jobs that may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code.
- Windows Management Instrumentation
  Network Connection Creation - Monitor network traffic for WMI connections; the use of WMI in environments that do not typically use WMI may be suspect.
- Modify System Process: WIndows Service
  OS API Execution - Monitor for API calls that may create or modify Windows services (ex: CreateServiceW()) to repeatedly execute malicious payloads as part of persistence.
  Service Creation - Creation of new services may generate an alterable event (ex: Event ID 4697 and/or 7045), especially those associated with unknown/abnormal drivers. New, benign services may be created during installation of new software.
  Service Modification - Monitor for changes made to Windows services to repeatedly execute malicious payloads as part of persistence.
  Windows Registry Key Creation - Monitor for new constructed windows registry keys that may create or modify Windows services to repeatedly execute malicious payloads as part of persistence.
  Windows Registry Key Modification - Look for changes to service Registry entries that do not correlate with known software, patch cycles, etc. Service information is stored in the Registry at HKLM\SYSTEM\CurrentControlSet\Services. Changes to the binary path and the service startup type changed from manual or disabled to automatic, if it does not typically do so, may be suspicious. Tools such as Sysinternals Autoruns may also be used to detect system service changes that could be attempts at persistence.
- Boot or Logon AutoStart Execution: Registry Run Keys / Startup Folder
  Command Execution - Monitor executed commands and arguments that may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key.
  File Modification - Monitor the start folder for additions or changes. Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including the startup folders.
- Boot of Logon AutoStart Execution: Authentication Package
  Module Load - Monitor the LSA process for DLL loads. Windows 8.1 and Windows Server 2012 R2 may generate events when unsigned DLLs try to load into the LSA by setting the Registry key HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe with

AuditLevel = 8
Command Execution - Monitor executed commands and arguments that may abuse authentication packages to execute DLLs when the system boots.

- Masquerading
  File Metadata - Collect file hashes; file names that do not match their expected hash are suspect. Perform file monitoring; files with known names but in unusual locations are suspect. Look for indications of common characters that may indicate an attempt to trick users into misidentifying the file type, such as a space as the last character of a file name or the right-to-left override characters"\u202E", "[U+202E]", and "%E2%80%AE".
  Image Metadata - Collecting disk and resource filenames for binaries, comparing that the InternalName, OriginalFilename, and/or ProductName match what is expected, could provide useful leads but may not always be indicative of malicious activity.
  Process Metadata - Monitor for file names that are mismatched between the file name on disk and that of the binary's PE metadata, this is a likely indicator that a binary was renamed after it was compiled
- Process Injection
  File Modification - Monitor for changes made to files that may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges.
  Module Load - Monitor DLL/PE file events, specifically creation of these binary files as well as the loading of DLLs into processes. Look for DLLs that are not recognized or not normally loaded into a process
  Process Access - Monitor for processes being viewed that may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges.
  Process Modification - Monitor for changes made to processes that may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges.

## Mitigations

- Exploit Public-Facing Application
  Application Isolation and Sandboxing - Application isolation will limit what other processes and system features the exploited target can access.
  Exploit Protection - Web Application Firewalls may be used to limit exposure of applications to prevent exploit traffic from reaching the application.
  Network Segmentation - Segment externally facing servers and services from the rest of the network with a DMZ or on separate hosting infrastructure.
  Privileged Account Management - Use least privilege for service accounts will limit what permissions the exploited process gets on the rest of the system.
  Update Software - Update software regularly by employing patch management for externally exposed applications.
  Vulnerability Scanning - Regularly scan externally facing systems for vulnerabilities and establish procedures to rapidly patch systems when critical vulnerabilities are discovered through scanning and through public disclosure.
- Valid Accounts
  Password Policies - Applications and appliances that utilize default username and password should be changed immediately after the installation, and before deployment to a production environment. When possible, applications that use SSH keys should be updated periodically and properly secured.
  Privileged Account Management - Audit domain and local accounts as well as their permission levels routinely to look for situations that could allow an adversary to gain wide access by obtaining credentials of a privileged account. These audits should also include if default accounts have been enabled, or if new local accounts are created that have not been authorized. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers.
  User Account Management - Regularly audit user accounts for activity and deactivate or remove any that are no longer needed.
  User Training - Applications may send push notifications to verify a login as a form of multi-factor authentication (MFA). Train users to only accept valid push notifications and to report suspicious push notifications.
- Data Encrypted for Impact
  Behavior Prevention on Endpoint - On Windows 10, enable cloud-delivered protection and Attack Surface Reduction (ASR) rules to block the execution of files that resemble ransomware.
  Data Backup - Consider implementing IT disaster recovery plans that contain procedures for regularly taking and testing data backups that can be used to restore organizational data. Ensure backups are stored off system and are protected from common methods adversaries may use to gain access and destroy the backups to prevent recovery. Consider enabling versioning in cloud environments to maintain backup copies of storage objects.
- Exploitation for Privilege Escalation
  Execution Prevention - Consider blocking the execution of known vulnerable drivers that adversaries may exploit to execute code in kernel mode. Validate driver block rules in audit mode to ensure stability prior to production deployment
  Exploit Protection - Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior. Control flow integrity checking is another way to potentially identify and stop a software exploit from occurring. Many of these protections depend on the architecture and target application binary for compatibility and may not work for software components targeted for privilege escalation.
  Threat Intelligence Program - Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization.
  Update Software - Update software regularly by employing patch management for internal enterprise endpoints and servers.
- Scheduled Task/Job
  Audit - Toolkits like the PowerSploit framework contain PowerUp modules that can be used to explore systems for permission weaknesses in scheduled tasks that could be used to escalate privileges.
  Operating System Configuration - Configure settings for scheduled tasks to force tasks to run under the context of the authenticated account instead of

allowing them to run as SYSTEM. The associated Registry key is located at HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SubmitControl. The setting can be configured through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > Security Options: Domain Controller: Allow server operators to schedule tasks, set to disabled.

- Windows Management Instrumentation
  Privileged Account Management - Prevent credential overlap across systems of administrator and privileged accounts.
  User Account Management - By default, only administrators are allowed to connect remotely using WMI. Restrict other users who are allowed to connect, or disallow all users to connect remotely to WMI.
- Modify System Process: WIndows Service
  Code Signing - Enforce registration and execution of only legitimately signed service drivers where possible.
  Operating System Configuration - Ensure that Driver Signature Enforcement is enabled to restrict unsigned drivers from being installed.
  Audit - Use auditing tools capable of detecting privilege and service abuse opportunities on systems within an enterprise and correct them.
- Boot or Logon AutoStart Execution: Registry Run Keys / Startup Folder
  This type of attack technique cannot be easily mitigated with preventative controls since it is based on the abuse of system features
- Boot of Logon AutoStart Execution: Authentication Package
  Privileged Process Integrity - Windows 8.1, Windows Server 2012 R2, and later versions, may make LSA run as a Protected Process Light (PPL) by setting the Registry key HKLM\SYSTEM\CurrentControlSet\Control\Lsa\RunAsPPL, which requires all DLLs loaded by LSA to be signed by Microsoft.
- Masquerading
  Restrict File and Directory Permissions - Use file system access controls to protect folders such as C:\Windows\System32.
- Process Injection
  Privileged Account Management - Utilize Yama (ex: /proc/sys/kernel/yama/ptrace_scope) to mitigate ptrace based process injection by restricting the use of ptrace to privileged users only. Other mitigation controls involve the deployment of security kernel modules that provide advanced access control and process restrictions such as SELinux, grsecurity, and AppArmor.
  Behavior Prevention on Endpoint - Some endpoint security solutions can be configured to block some types of process injection based on common sequences of behavior that occur during the injection process. For example, on Windows 10, Attack Surface Reduction (ASR) rules may prevent Office applications from code injection

## Indicators of Compromise

| Indicator | Type |
|---|---|
| v-society.official@onionmail[.]org | Email Address |
| ViceSociety@onionmail[.]org | Email Address |
| http://vsociethok6sbprvevl4dlwbqrzyhxcxaqpvcqt5belwvsuxaxsutyad[.]onion | TOR Address |
| 5.255.99.59 | C2 IP Address |
| 5.161.136.176 | C2 IP Address |
| 198.252.98.184 | C2 IP Address |

## Incident Response

If administrators discover signs of attack or system compromise, the MD-ISAC recommends they:

- Immediately isolate affected systems.
- Collect and review relevant logs, data, and artifacts.
- Consider soliciting support from a third-party incident response organization to provide subject matter expertise, ensure the actor is eradicated from the network, and avoid residual issues that could enable follow-on exploitation.
- Report incidents to MD-ISAC via Maryland's 24/7 Operations Center (soc@maryland.gov or (410) 697-9700 - option #5).

## Contact Information

To report suspicious or criminal activity related to information found in this Threat Bulletin, contact the Maryland Security Operations Center at (410) 697-9700 or by email at md-isac@maryland.gov. When available, please include the following information regarding the incident: date, time, and location of the incident; type of activity; number of people affected; type of equipment used for the activity; the name of the submitting company or organization; and a designated point of contact. Please state in the report if you are requesting incident response resources or technical assistance related to the incident.
References