Abstract – This document provides an in-depth analysis of the National Security Agency's (NSA) advisory on combatting cyber threat actors who perpetrate Living Off the Land (LOTL) intrusions. The analysis encompasses a thorough examination of the advisory's multifaceted approach to addressing LOTL tactics, which are increasingly leveraged by adversaries to exploit legitimate tools within a target's environment for malicious purposes.

The analysis offers a high-quality summary of the NSA's advisory, distilling its key points into actionable insights. It serves as a valuable resource for security professionals, IT personnel, policymakers, and stakeholders across various industries, providing them with the knowledge to enhance their defensive capabilities against sophisticated LOTL cyber threats. By implementing the advisory's recommendations, these professionals can improve their situational awareness, refine their security posture, and develop more robust defense mechanisms to protect against the subtle and stealthy nature of LOTL intrusions.

I. Introduction

The document titled "Joint Guidance: Identifying and Mitigating LOTL Techniques" provides guidance on how organizations can better protect themselves against Living Off the Land (LOTL) techniques. These techniques involve cyber threat actors leveraging legitimate tools and software present within the target's environment to conduct malicious activities, making detection more challenging. This approach aims to reduce the availability of legitimate operating system and application tools (LOLBins) that threat actors can exploit.

The guidance is based on insights from a joint advisory, red team assessments by the authoring agencies, authoring agency incident response engagements and collaborative efforts with the industry. It stresses the importance of establishing and maintaining an infrastructure that collects and organizes data to help defenders detect LOTL techniques, tailored to each organization's risk landscape and resource capabilities.

A. Main keypoints

- **Authoring Agencies**: The guide is authored by major cybersecurity and national security agencies from the U.S., Australia, Canada, the United Kingdom, and New Zealand, focusing on common LOTL techniques and gaps in cyber defense capabilities.
- **LOTL Techniques**: Cyber threat actors use LOTL techniques to compromise and maintain access to critical infrastructure, leveraging legitimate system tools and processes to blend in with normal activities and evade detection.
- **Challenges in Detection**: Many organizations struggle to detect malicious LOTL activity due to inadequate security and network management practices, lack of conventional indicators of compromise, and the difficulty of distinguishing malicious activity from legitimate behavior.
- **Detection Best Practices**: Recommendations include implementing detailed logging, establishing activity baselines, utilizing automation for continuous review, reducing alert noise, and leveraging user and entity behavior analytics (UEBA).
- **Hardening Best Practices**: Suggestions involve applying vendor-recommended security hardening guidance, implementing application allowlisting, enhancing network segmentation and monitoring, and enforcing authentication and authorization controls.
- **Software Manufacturer Recommendations**: The guide urges software manufacturers to adopt secure by design principles to reduce exploitable flaws that enable LOTL techniques. This includes disabling unnecessary protocols, limiting network reachability, restricting elevated privileges, enabling phishing-resistant MFA by default, providing secure logging, eliminating default passwords, and limiting dynamic code execution.

B. Secondary keypoints

- The guidance is aimed at helping organizations mitigate Living Off The Land (LOTL) techniques, where threat actors use legitimate tools within the environment for malicious purposes.
- Organizations are advised to exercise due diligence when selecting software, devices, cloud service providers, and managed service providers, choosing those with secure by design principles.
- Vendors should be held accountable for their software's default configurations and adherence to the principle of least privilege.
- Software manufacturers are encouraged to reduce exploitable flaws and take ownership of their customers' security outcomes.
- Network defense strategies include monitoring for unusual system interactions, privilege escalations, and deviations from normal administrative actions.

- Organizations should establish and maintain an infrastructure for collecting and organizing data to detect LOTL techniques, tailored to their specific risk landscape and resource capabilities

II. Benefits and Drawbacks

The analyzed document outlines a comprehensive approach to enhance cybersecurity defenses against LOTL tactics. This approach includes recommendations for detection and logging, centralized logging, behavior analytics, anomaly detection, and proactive hunting.

While the proposed solutions offer significant benefits in enhancing cybersecurity defenses against LOTL tactics, organizations must also consider the potential drawbacks and limitations. Effective implementation requires careful planning, resource allocation, and continuous adjustment to address the evolving threat landscape.

A. Benefits

- **Enhanced Detection Capabilities**: Implementing comprehensive and verbose logging, along with centralized logging, significantly enhances an organization's ability to detect malicious activities. This approach enables behavior analytics, anomaly detection, and proactive hunting, providing a robust defense against LOTL techniques.
- **Improved Security Posture**: The guidance recommends hardening measures such as applying vendor-provided or industry-standard hardening guidance, minimizing running services, and securing network communications. These measures reduce the attack surface and improve the overall security posture of organizations.
- **Increased Visibility**: Centralized logging allows for the maintenance of longer log histories, which is crucial for identifying patterns and anomalies over time. This increased visibility into network and system activities aids in the early detection of potential threats.
- **Efficient Use of Resources**: Automation of log review and hunting activities increases the efficiency of these processes, enabling organizations to better utilize their resources. Automated systems can compare current activities against established behavioral baselines, focusing on privileged accounts and critical assets.
- **Strategic Network Segmentation**: Enhancing network segmentation and monitoring limits lateral movement possibilities for threat actors, reducing the "blast radius" of accessible systems in the event of a compromise. This strategic approach helps contain threats and minimizes potential damage.

B. Drawbacks/Limitations

- **Resource Intensiveness**: Implementing the recommended detection and hardening measures can be resource-intensive, requiring significant investment in technology and personnel training. Smaller organizations may find it challenging to allocate the necessary resources.
- **Complexity of Implementation**: Establishing and maintaining the infrastructure for comprehensive logging and analysis can be complex. Organizations may face challenges in configuring and managing these systems effectively, especially in diverse and dynamic IT environments.
- **Potential for Alert Fatigue**: While reducing alert noise is a goal of the proposed solutions, the sheer volume of logs and alerts generated by comprehensive logging and anomaly detection systems can lead to alert fatigue among security personnel, potentially causing critical alerts to be overlooked.
- **False Positives and Negatives**: Behavior analytics and anomaly detection systems may generate false positives and negatives, leading to unnecessary investigations or missed threats. Fine-tuning these systems to minimize inaccuracies requires ongoing effort and expertise.
- **Dependence on Vendor Support**: The effectiveness of hardening measures and secure configurations often depends on the support and guidance provided by software vendors. Organizations may face limitations if vendors do not prioritize security or provide adequate hardening guidelines.

III. Living Off the Land

Living Off the Land (LOTL) techniques represent a sophisticated cyber threat strategy where attackers exploit native tools and processes already present within a target's environment. This approach allows them to blend seamlessly with normal system activities, significantly reducing the likelihood of detection. The effectiveness of LOTL lies in its ability to utilize tools that are not only already deployed but are also trusted within the environment, thereby circumventing traditional security measures that might block or flag unfamiliar or malicious software.

LOTL techniques are not confined to a single type of environment; they are effectively used across on-premises, cloud, hybrid, Windows, Linux, and macOS environments. This versatility is partly due to the attackers' preference to avoid the costs and efforts associated with developing and deploying custom tools. Instead, they leverage the ubiquity and inherent trust of native tools to carry out their operations.

A. Windows Environments

In Windows environments, which are prevalent in corporate and enterprise settings, LOTL techniques are particularly observed due to the widespread use and trust in the operating system's native tools, services, and features. Attackers exploit these components, knowing they are ubiquitous and generally trusted, making their malicious activities less likely to be detected.

B. macOS and Hybrid Environments

In macOS environments, the concept of LOTL is often referred to as "living off the orchard." Here, attackers exploit native scripting environments, built-in tools, system

configurations, and binaries, known as "LOOBins." The strategy is similar to that in Windows environments but tailored to the unique aspects of macOS. In hybrid environments, which combine physical and cloud-based systems, attackers are increasingly leveraging sophisticated LOTL techniques to exploit both types of systems.

C. *Resources and Known Exploits*

There are several resources provide comprehensive lists and information to understand the specific tools and binaries exploited by attackers:

- The LOLBAS project's GitHub repository offers insights into Living Off The Land Binaries, Scripts, and Libraries.
- Websites like gtfobins.github.io, loobins.io, and loldrivers.io provide lists of Unix, macOS, and Windows binaries, respectively, known to be used in LOTL techniques.

D. *Third-Party Remote Access Software*

Beyond native tools, cyber threat actors also exploit third-party remote access software, such as remote monitoring and management, endpoint configuration management, EDR, patch management, mobile device management systems, and database management tools. These tools, designed to administer and protect domains, possess built-in functionality that can execute commands across all client hosts in a network, including critical hosts like domain controllers. The high privileges these tools require for system administration make them attractive targets for attackers looking to exploit them for LOTL techniques.

IV. Security Baselines and Alert Noise

One of the primary issues identified is the lack of security baselines within organizations, which permits the execution of living off the land binaries (LOLBins) without detection of anomalous activity. Additionally, organizations often fail to fine-tune their detection tools, resulting in an overwhelming number of alerts that are difficult to manage and act upon. This is compounded by automated systems performing highly privileged actions that can flood analysts with log events if not properly categorized.

A. *Challenges in Distinguishing Malicious Activity*

Even organizations with mature cyber postures and best practices in place find it difficult to distinguish between malicious LOTL activity and legitimate behavior:

- LOLBins are commonly used by IT administrators and are therefore trusted, which can mislead network defenders into assuming they are safe for all users.
- There is a misconception that legitimate IT administrative tools are globally safe, leading to blanket "allow" policies that expand the attack surface.
- Overly broad exceptions for tools like PsExec, due to their regular use by administrators, can be exploited by malicious actors to move laterally without detection.

B. *Siloed Operations and Untuned EDR Systems*

The red team and incident response teams have frequently observed that network defenders:

- Operate in silos, separate from IT teams, hindering the creation of user behavior baselines and delaying vulnerability remediation and abnormal behavior investigations.
- Rely on untuned endpoint detection and response (EDR) systems and discrete indicators of compromise (IOCs), which may not trigger alerts for LOTL activity and can be easily altered by attackers to avoid detection.

C. *Logging Configurations and Allowlisting Policies*

Deficiencies in logging configurations and allowlisting policies further complicate the detection of LOTL activities:

- Default logging configurations often fail to capture all relevant activity, and logs from many applications require additional processing to be useful for network defense.
- Broad allowlisting policies for IP address ranges owned by hosting and cloud providers can inadvertently provide cover for malicious actors.

D. *macOS Device Protections*

Network defenders must also ensure adequate protections for macOS devices, which are often mistakenly considered inherently secure:

- macOS lacks standardized system hardening guidance, leading to deployments with default settings that may not be secure.
- The presumption of macOS safety can result in the deprioritization of standard security measures, such as security assessments and application allowlisting.
- In mixed-OS environments, the lower representation of macOS devices can lead to a lack of attention to their security, making them more vulnerable to intrusions.

V. Detection Opportunities

A. *Comprehensive and Detailed Logging*

- **Implementation of Comprehensive Logging**: Establishing extensive and detailed logging mechanisms is crucial. This includes enabling logging for all security-related events across platforms and ensuring that logs are aggregated in a secure, centralized location to prevent tampering by adversaries.
- **Cloud Environment Logging**: For cloud environments, it's essential to enable logging for control plane operations and configure logging policies for all cloud services, even those not actively used, to detect potential unauthorized activities.
- **Verbose Logging for Security Events**: Enabling verbose logging for events such as command lines, PowerShell activities, and WMI event tracing provides

deeper visibility into tool usage within the environment, aiding in the detection of malicious LOTL activities.

B. Establishing Behavioral Baselines

- **Maintaining Baselines**: Continuously maintaining a baseline of installed tools, software, account behavior, and network traffic allows defenders to identify deviations that may indicate malicious activity.
- **Network Monitoring and Threat Hunting**: Enhancing network monitoring, extending log storage, and deepening threat hunting tactics are vital for uncovering prolonged adversary presence leveraging LOTL techniques.

C. Automation and Efficiency

- **Leveraging Automation**: Using automation to review logs continually and compare current activities against established behavioral baselines increases the efficiency of hunting activities, especially focusing on privileged accounts and critical assets.

D. Reducing Alert Noise

- **Refining Monitoring Tools**: It's important to refine monitoring tools and alerting mechanisms to differentiate between typical administrative actions and potential threat behavior, thus focusing on alerts that most likely indicate suspicious activities.

E. Leveraging UEBA

- **User and Entity Behavior Analytics (UEBA)**: Employing UEBA to analyze and correlate activities across multiple data sources helps identify potential security incidents that may be missed by traditional tools and profiles user behavior to detect insider threats or compromised accounts.

F. Cloud-Specific Considerations

- **Cloud Environment Architecting**: Architecting cloud environments to ensure proper separation of enclaves and enabling additional logs within the environment provide more insight into potential LOTL activities.

VI. Hardening Strategies

These strategies are aimed at reducing the attack surface and enhancing the security posture of organizations and their critical infrastructure.

A. Hardening Guidance

Vendor and Industry Hardening Guidance: Organizations should strengthen software and system configurations based on vendor-provided or industry, sector, or government hardening guidance, such as those from NIST, to reduce the attack surface.

1) Platform-Specific Hardening:

- **Windows**: Apply security updates and patches from Microsoft, follow Windows Security Baselines Guide or CIS Benchmarks, harden commonly exploited services like SMB and RDP, and disable unnecessary services and features.
- **Linux**: Check binary permissions and adhere to CIS's Red Hat Enterprise Linux Benchmarks.
- **macOS**: Regularly update and patch the system, use built-in security features like Gatekeeper, XProtect, and FileVault, and follow the macOS Security Compliance Project's guidelines.

2) Cloud Infrastructure Hardening:

- **Microsoft Cloud**: Refer to CISA's Microsoft 365 security configuration baseline guides for secure configuration baselines across various Microsoft cloud services.
- **Google Cloud**: Consult CISA's Google Workspace security configuration baseline guides for secure configuration baselines across Google cloud services.
- **Universal Hardening Measures**: Minimize running services, apply the principle of least privilege, and secure network communications.
- **Critical Asset Security**: Apply vendor hardening measures for critical assets like ADFS and ADCS and limit the applications and services that can be used or accessed by them.
- **Administrative Tools**: Use tools that do not cache credentials on the remote host to prevent threat actors from reusing compromised credentials.

B. Application Allowlisting

Constrain Execution Environment: Implement application allowlisting to channel user and administrative activity through a narrow path, enhancing monitoring and reducing alert volume.

1) Platform-Specific Allowlisting:

- **macOS**: Configure Gatekeeper settings to prevent execution of unsigned or unauthorized applications.
- **Windows**: Use AppLocker and Windows Defender Application Control to regulate executable files, scripts, MSI files, DLLs, and packaged app formats.

C. Network Segmentation and Monitoring

- **Limit Lateral Movement**: Implement network segmentation to limit the access of users to the minimum necessary applications and services, reducing the impact of compromised credentials.
- **Network Traffic Analysis**: Use tools to monitor traffic between segments and place network sensors at critical points for comprehensive traffic analysis.
- **Network Traffic Metadata Parsing**: Utilize parsers like Zeek and integrate NIDS like Snort or Suricata to detect LOTL activities.

D. Authentication Controls

- **Phishing-Resistant MFA**: Enforce MFA across all systems, especially for privileged accounts.

- **Privileged Access Management (PAM)**: Deploy robust PAM solutions with just-in-time access and time-based controls, complemented by role-based access control (RBAC).
- **Cloud Identity and Credential Access Management (ICAM)**: Enforce strict ICAM policies, audit configurations, and rotate access keys.
- **Sudoers File Review**: For macOS and Unix, regularly review the sudoers file for misconfigurations and adhere to the principle of least privilege.

E. Zero Trust Architecture

As a long-term strategy, the guidance recommends implementing zero trust architectures to ensure that binaries and accounts are not automatically trusted and their use is restricted and examined for trustworthy behavior.

F. Additional Recommendations

- **Due Diligence in Vendor Selection**: Choose vendors with secure by design principles and hold them accountable for their software's default configurations.
- **Audit Remote Access Software**: Identify authorized remote access software and apply best practices for securing remote access.
- **Restrict Outbound Internet Connectivity**: Limit internet access for back-end servers and monitor outbound connectivity for essential services.

VII. Detection and Hunting Recommendations

It advocates for regular system inventory audits to catch adversary behavior that might be missed by event logs due to inadequate logging configurations or activities occurring before logging enhancements are deployed. Organizations are encouraged to enable comprehensive logging for all security-related events, including shell activities, system calls, and audit trails across all platforms, to improve the detection of malicious LOTL activity.

A. Network Logs

The detection of LOTL techniques through network logs presents unique challenges due to the transient nature of network artifacts and the complexity of distinguishing malicious activity from legitimate behavior. Network defenders must be vigilant and proactive in configuring and setting up logs to capture the necessary data for identifying LOTL activities. Unlike host artifacts, which can often be found unless deliberately deleted by a threat actor, network artifacts are derived from network traffic and are inherently more difficult to detect and capture. Network artifacts are significantly harder to detect than host artifacts because they are largely transient and require proper configuration of logging systems to be captured. Without the right sensors in place to record network traffic, there is no way to observe LOTL activity from a network perspective.

B. Indicators of LOTL Activity

Detecting LOTL activity involves looking for a collection of possible indicators that, together, paint a picture of the behavior of network traffic.

- **Reviewing Firewall Logs**: Blocked access attempts in firewall logs can signal compromise, especially in a properly segmented network. Network discovery and mapping attempts from within the network can also be indicative of LOTL activity. It is crucial to differentiate between normal network management tool behavior and abnormal traffic patterns.
- **Investigating Unusual Traffic Patterns**: Specific types of traffic should be scrutinized, such as LDAP requests from non-domain joined Linux hosts, SMB requests across different network segments, or database access requests from user workstations that should only be made by frontend servers. Establishing baseline noise levels can help in distinguishing between legitimate applications and malicious requests.
- **Examining Logs from Network Services on Host Machines**: Logs from services like Sysmon and IIS on host machines can provide insights into web server interactions, FTP transactions, and other network activities. These logs can offer valuable context and details that may not be captured by traditional network devices.
- **Combining Network Traffic Logs with Host-based Logs**: This approach allows for the inclusion of additional information such as user account and process details. Discrepancies between the destination and on-network artifacts could indicate malicious traffic.

C. Application, Security, and System Event Logs

Default logging configurations often fail to capture all necessary events, potentially leaving gaps in the visibility of malicious activities. Prioritizing logs and data sources that are more likely to reveal malicious LOTL activities is crucial for effective detection and response.

D. Authentication Logs

Authentication logs play a vital role in identifying unauthorized access attempts and tracking user activities across the network. The guidance recommends ensuring that logging is enabled for all control plane operations, including API calls and end-user logins, through services like Amazon Web Services CloudTrail, Azure Activity Log, and Google Cloud Audit Logs. These logs can provide valuable insights into potential LOTL activities by highlighting unusual access patterns or attempts to exploit authentication mechanisms.

A robust strategy for the separation of privileges is essential for identifying LOTL techniques through authentication logs. Practices such as restricting domain administrator accounts to only log into domain controllers and using Privileged Access Workstations (PAWs) in conjunction with bastion hosts can minimize credential exposure and reinforce network segmentation. Multifactor authentication adds an additional layer of security.

E. Host-based Logs

Sysmon and other host-based logging tools offer granular visibility into system activities that can indicate LOTL exploitation. By capturing detailed information about process

creations, network connections, and file system changes, these tools can help organizations detect and investigate suspicious behavior that might otherwise go unnoticed.

1) Establishing Baselines and Secure Logging

A foundational step in detecting abnormal or potentially malicious behavior is the establishment of baselines for running tools and activities. This involves understanding the normal operational patterns of a system to identify deviations that may indicate a security threat. It's also essential to rely on secure logs that are less susceptible to tampering by adversaries. For instance, while Linux .bash_history files can be modified by nonprivileged users, system-level auditd logs are more secure and provide a reliable record of activities.

2) Leveraging Sysmon in Windows Environments

Sysmon, a Windows system monitoring tool, offers granular insights into activities such as process creations, network connections, and registry modifications. This detailed logging is invaluable for security teams in hunting for and detecting the misuse of legitimate tools and utilities. Key strategies include:

- Using the OriginalFileName property to identify renamed files, which may indicate malicious activity. For most Microsoft utilities, the original filenames are stored in the PE header, providing a method to detect file tampering.
- Implementing detection techniques to identify the malicious use of command-line and scripting utilities, especially those exploiting Alternate Data Streams (ADS). Monitoring specific command-line arguments or syntax used to interact with ADS can reveal attempts to execute or interact with hidden payloads.

3) Targeted Detection Strategies

Enhancing Sysmon configurations to log and scrutinize command-line executions, with a focus on patterns indicative of obfuscation, can help identify attempts by cyber threat actors to bypass security monitoring tools. Examples include the extensive use of escape characters, concatenation of commands, and the employment of Base64 encoding.

4) Monitoring Suspicious Process Chains

Monitoring for suspicious process chains, such as Microsoft Office documents initiating scripting processes, is a key indicator of LOTL activity. It's uncommon for Office applications to launch scripting processes like cmd.exe, PowerShell, wscript.exe, or cscript.exe. Tracking these process creations and the execution of unusual commands from Office applications can signal a red flag and warrants further investigation.

5) Integrating Logs with SIEM Systems

Integrating Sysmon logs with Security Information and Event Management (SIEM) systems and applying correlation rules can significantly enhance the detection of advanced attack scenarios. This integration allows for the automation of the detection process and the application of analytics to identify complex patterns of malicious activity.

6) Linux and macOS Considerations

On Linux machines, enabling Auditd or Sysmon for Linux logging and integrating these logs with an SIEM platform can greatly improve the detection of anomalous activities. For macOS, utilizing tools like Santa, an open-source binary authorization system, can help monitor process executions and detect abnormal behavior by productivity applications

F. Review Configurations

Regularly reviewing and updating system configurations is essential to ensure that security measures remain effective against evolving threats. This includes verifying that logging settings are appropriately configured to capture relevant data and that security controls are aligned with current best practices. Organizations should also assess the use of allowlists and other access control mechanisms to prevent the misuse of legitimate tools by malicious actors.

Regular reviews of host configurations against established baselines are essential for catching indicators of compromise (IOCs) that may not be reverted through regular group policy updates. This includes changes to installed software, firewall configurations, and updates to core files such as the Hosts file, which is used for DNS resolution. Such reviews can reveal discrepancies that signal unauthorized modifications or the presence of malicious software.

- **Bypassing Standard Event Logs**: Cyber threat actors have been known to bypass standard event logs by directly writing to the registry to register services and scheduled tasks. This method does not create standard system events, making it a stealthy way to establish persistence or execute tasks without triggering alerts.
- **System Inventory Audits**: Conducting regular system inventory audits is a proactive measure to catch adversary behavior that may have been missed by event logs, whether due to incorrect event capture or activities that occurred before logging enhancements were deployed. These audits help ensure that any changes to the system are authorized and accounted for.

G. Behavioral Analysis

Comparing activity against normal user behavior is key to detecting anomalies. Unusual behaviors to look out for include odd login hours, access outside of expected work schedules or holiday breaks, rapid succession or high volume of access attempts, unusual access paths, concurrent sign-ins from multiple locations, and instances of impossible time travel.

H. NTDSUtil.exe and PSExec.exe

Specific attention is given to detecting misuse of NTDSUtil.exe and PSExec.exe, tools that, while legitimate, are often leveraged by attackers for malicious purposes, such as attempts to dump credentials or move laterally across the network. By focusing on the behavioral context of these tools' usage, organizations can more effectively distinguish between legitimate and malicious activities.

1) The Exploitation Process

A common tactic involves creating a volume shadow copy of the system drive, typically using vssadmin.exe with commands like Create Shadow /for=C:. This action captures a

snapshot of the system's current state, including the Active Directory database. Following this, ntdsutil.exe is employed to interact with this shadow copy through a specific command sequence (ntdsutil snapshot “activate instance ntds” create quit quit). The attackers then access the shadow copy to extract the ntds.dit file from a specified directory. This sequence aims to retrieve sensitive credentials, such as hashed passwords, from the Active Directory, enabling full domain compromise.

2) Detection and Response

To detect and respond to such exploitation, it's crucial to understand the context of ntdsutil.exe activities and differentiate between legitimate administrative use and potential malicious exploitation. Key log sources and monitoring strategies include:

- **Command-line and Process Creation Logs**: Security logs (Event ID 4688) and Sysmon logs (Event ID 1) provide insights into the execution of ntdsutil.exe commands. Unusual or infrequent use of ntdsutil.exe for snapshot creation might indicate suspicious activity.
- **File Creation and Access Logs**: Monitoring file creation events (Sysmon’s Event ID 11) and attempts to access sensitive files like NTDS.dit (security logs with Event ID 4663) can offer additional context to the snapshot creation and access process.
- **Privilege Use Logs**: Event ID 4673 in security logs, indicating the use of privileged services, can signal potential misuse when correlated with the execution of ntdsutil.exe commands.
- **Network Activity and Authentication Logs**: These logs can provide context about concurrent remote connections or data transfers, potentially indicating data exfiltration attempts. Authentication logs are also crucial for identifying the executor of the ntdsutil.exe command and assessing whether the usage aligns with typical administrative behavior.

3) Comprehensive Analysis of PSExec.exe in LOTL Tactics

PSExec.exe, a component of the Microsoft PsTools suite, is a powerful utility for system administrators, offering the capability to remotely execute commands across networked systems, often with elevated SYSTEM privileges. Its versatility, however, also makes it a favored tool in Living Off the Land (LOTL) tactics employed by cyber threat actors.

4) The Role of PSExec.exe in Cyber Threats

PSExec.exe is commonly utilized for remote administration and the execution of processes across systems, such as execute one-off commands aimed at modifying system configurations, such as removing port proxy configurations on a remote host with commands like:

"C:\pstools\psexec.exe" {REDACTED} -s cmd /c "cmd.exe /c netsh interface portproxy delete v4tov4 listenaddress=0.0.0.0 listenport=9999"

5) Detection and Contextualization Strategies

To effectively counter the malicious use of PSExec.exe, network defenders must leverage a variety of logs that provide insights into the execution of commands and the broader context of the operation:

- **Command-line and Process Creation Logs**: Security logs (Event ID 4688) and Sysmon logs (Event ID 1) are invaluable for tracking the execution of PSExec.exe and associated commands. These logs detail the command line used, shedding light on the process's nature and intent.
- **Privilege Use and Explicit Credential Logs**: Security logs (Event ID 4672) document instances where special privileges are assigned to new logons, crucial when PSExec is executed with the -s switch for SYSTEM privileges. Event ID 4648 captures explicit credential use, indicating when PSExec is run with specific user credentials.
- **Sysmon Logs for Network Connections and Registry Changes**: Sysmon's Event ID 3 logs network connections, central to PSExec’s remote execution functionality. Event IDs 12, 13, and 14 track registry changes, including deletions (Event ID 14) of registry keys associated with the executed Netsh command, providing evidence of modifications to the system's configuration.
- **Windows Registry Audit Logs**: If enabled, these logs record modifications to registry keys, offering detailed information such as the timestamp of changes, the account under which changes were made (often the SYSTEM account due to PSExec's -s switch), and the specific registry values altered or deleted.
- **Network and Firewall Logs**: Analysis of network traffic, especially SMB traffic characteristic of PSExec use, and firewall logs on the target system can reveal connections to administrative shares and changes to the system's network configuration. These logs can correlate with the timing of command execution, providing further context to the activity.

VIII. Remediation Strategies for Compromised Networks

When an organization detects a compromise, especially involving Living Off the Land (LOTL) tactics, it is critical to implement immediate defensive countermeasures. The Joint Guidance on Identifying and Mitigating LOTL Techniques outlines a comprehensive remediation strategy that organizations should follow to mitigate the impact of such incidents.

A. Immediate Response Actions

- Reset credentials for both privileged and non-privileged accounts within the trust boundary of each compromised account.
- Force password resets and revoke and issue new certificates for all accounts and devices.

B. Windows Environment Specific Actions:

- If access to the Domain Controller (DC) or Active Directory (AD) is suspected, reset all local account passwords, including Guest, HelpAssistant, DefaultAccount, System, Administrator, and krbtgt. The

krbtgt account, which handles Kerberos ticket requests, should be reset twice to ensure security due to its two-password history.

- If the ntds.dit file is suspected to have been exfiltrated, reset all domain user passwords.
- Review and adjust access policies, temporarily revoking or reducing privileges to contain affected accounts and devices.
- Reset Non-Elevated Account Credentials: If the threat actor's access is limited to non-elevated permissions, reset the relevant account credentials or access keys and monitor for further signs of unauthorized access, especially for administrative accounts.

C. *Network and Device Configuration Audit*

- **Audit Network Appliances and Edge Devices**: Check for signs of unauthorized or malicious configuration changes. If changes are found:
  - Change all credentials used to manage network devices, including keys and strings securing network device functions.
  - Update all firmware and software to the latest versions.

D. *Remote Access Tool Usage*

Minimize and Control Remote Access: Follow best practices for securing remote access tools and protocols, including guidance on securing remote access software and using PowerShell securely.

IX. Recommendations for Software Manufacturers

These recommendations is crucial in reducing the prevalence of exploitable flaws that enable LOTL tactics.

A. *Minimizing Attack Surfaces*

Software manufacturers are urged to minimize attack surfaces that can be exploited by cyber threat actors using LOTL techniques. This includes disabling unnecessary protocols by default, limiting the number of processes and programs running with escalated privileges, and taking proactive steps to limit the ability for actors to leverage native functionality for intrusions.

B. *Embedding Security in the SDLC*

Security should be embedded into the product architecture throughout the entire software development lifecycle (SDLC). This proactive integration ensures that security considerations are not an afterthought but a fundamental component of the product from inception to deployment.

C. *Mandating Multi-Factor Authentication (MFA)*

Manufacturers should mandate MFA, ideally phishing-resistant MFA, for privileged users and make it a default feature rather than an optional one. This step significantly enhances the security of user accounts, particularly those with elevated access.

D. *Reducing Hardening Guide Size*

The size of hardening guides that accompany products should be tracked and reduced. As new versions of the software are released, the aim should be to shrink the size of these guides over time by integrating their components as the default configuration of the product.

E. *Considering User Experience*

The user experience consequences of security settings must be considered. Ideally, the most secure setting should be integrated into the product by default, and when configuration is necessary, the default option should be secure against common threats. This approach reduces the cognitive burden on end users and ensures broad protection.

F. *Removing Default Passwords*

Default passwords should be eliminated entirely or, where necessary, be generated or set upon first install and then rotated periodically. This practice prevents the use of default passwords as an easy entry point for malicious actors.

G. *Limiting Dynamic Code Execution*

Dynamic code execution, while offering versatility, presents a vulnerable attack surface. Manufacturers should limit or remove the capability for dynamic code execution due to the high risk and the challenge of detecting associated indicators of compromise (IOCs).

H. *Removing Hard-Coded Credentials*

Applications and scripts containing hard-coded plaintext credentials pose a significant security risk. Removing such credentials is essential to prevent malicious actors from using them to access resources and expand their presence within a network.