BeyondTrust Response to Area #3: Security Measures for the Use of Critical Software

Critical software used across the federal government should have rigorous security measures applied to ensure its safe and secure use. Not only must the critical software itself be protected, but the agency IT environment, its customers, partners, and the public must be protected from any potential misuse or rogue activity posed by the software. This paper explores three of the foundational best practices to securing and protecting applications. A robust strategy to secure critical software should include systems hardening and configuration management, privileged access management and application control, and zero trust architecture, with an emphasis on least privilege, executed on an agency's network.

The first strategy, systems hardening and configuration management, provide a baseline for secure software. Systems should be regularly scanned and have automated ways to remove unnecessary, or block-listed, software and applications. Agencies must automate the process of privilege discovery, auditing, and onboarding across their environment to encompass the activities of humans, applications, and machines to ensure these are not leveraged to gain access to the installed critical software. These controls should be applied prior to the device/endpoint receiving access to an agency's network. Such hardening activities should be rigorously enforced across the device lifecycle. The proliferation of BYOD and work-from-home amongst federal and civilian workforces increases the necessity for the continual scan and hardening of such devices, especially if they periodically leave and enter the agency physically and connect to its network.

In rushing to support a large remote workforce during the early days of the COVID-19 pandemic, agencies and enterprises relaxed their hardening policies. In March 2020, internet-facing RDP ports increased by 50%[1] in an ill-advised attempt to support these WFH initiatives. Over the following 12 months, 52% of ransomware attacks leveraged publicly accessible RDP servers to gain initial access[2]. These breaches are a direct result of ignoring basic system and network hardening tenets.

Agencies should strive to provide secure application access independent of VPN access, only enabling contractors and vendors the applications and systems they need. This will increase protection against malicious insiders, external actors, and human errors. Increasingly, attackers are exploiting inadequately protected cloud applications using control planes inadvertently or recklessly exposed to the Internet. Agencies should proxy access to control planes and other critical software to segment and isolate remote access traffic in the cloud. In addition, admin access should be locked down and discoverable to authorized admins only through secure encrypted connections. Providing a locked-down, isolated browser that supports automatic web credential injection is recommended to prevent end users from ever interacting with any credentials.

In addition, non-Windows IT administrators should be assigned commands they can execute and run elevated without needing sudo or root. Agencies should use a policy language that can elevate commands via least privilege and inspect all the options and switches. This allows it to identify malformed or inappropriate commands, which could cause downtime of critical software and expose attack vectors that can be exploited.

Second, privileged access management (PAM) and application control must be applied across all operating systems and Operational Technology (OT). In implementing PAM, agencies should strive to manage privileged credentials. Management of these credentials/accounts also encompasses reducing the number of identities with privileged access, the quantity of privileged access, and the duration of privileged access. This approach not only drastically condenses the attack surface, but also vastly minimizes threat windows. Any hardcoded credentials should be eliminated and replaced with API calls or dynamic secrets, such as with DevOps and CI/CD toolsets. Credential security best practices should be consistently applied, such as enforcing unique, complex passwords, and using 2FA and MFA whenever possible. This helps prevent password re-use, spraying, stuffing, and many other attack vectors, especially with IoT and OT devices.

In 2019, 100% of Critical vulnerabilities in Internet Explorer and Edge would have been mitigated by removing admin rights[3]. A similarly powerful risk-reducing power of least-privilege has also been demonstrated across important third-party applications, with a quick search of the National Vulnerability Database[4], published by NIST, revealing a significant number of registered vulnerabilities mitigated by appropriate Elevated Privilege Management. Eliminating standing admin rights translates

[1] Roccia, T. (2020, May 06). *Cybercriminals Actively Exploiting RDP to Target Remote Organizations.* Retrieved from McAfee: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/cybercriminals-actively-exploiting-rdp-to-target-remote-organizations/

[2] Skulkin, O., Rezvukhin, R., & Rogachev, S. (2021). *Ransomware Uncovered 2020-2021.* Group-IB.

[3] Microsoft. (2020). *Microsoft Vulnerabilities Report.* Redmond, WA: Microsoft.

[4] https://nvd.nist.gov/vuln/search

into a significant risk reduction across many platforms, even absent an appropriate patching process in place. Agencies should remove local admin rights on endpoints and remove all root and admin access rights to servers. Moreover, by defaulting all users to standard privileges, while enabling elevated privileges for applications to perform specific tasks, malware such as samples deployed by Darkside, the group who perpetrated the devastating attack on Colonial Pipeline, could be minimized or fully mitigated.

Access should only be provisioned just-in-time (JIT) when certain contextual parameters are sufficiently met, and subsequently removed or expired immediately upon completion of the activity or change in behavior. A typical always-on privileged account will be active 24 hours a day, 7 days a week, for a total exposure time of 168 hours, versus just a few hours using a JIT approach. Multiplying this effect across all of an agency's privileged accounts will have a truly massive impact on risk-reduction. Context should also dynamically consider real-time vulnerability and threat data about an application, asset, or user so privileged access may be further restricted, denied, or launch workflows requiring additional approvals or monitoring.

The SolarWinds Orion supply chain attack was particularly devastating because the Orion application needed unrestricted access to work. Since the Orion application itself was compromised, threat actors leveraged this unrestricted privileged access throughout victims' environments using the application. Restricting software and system privileges to the minimal range of processes to perform an authorized activity reduces risk of downtime and the attack surface, improving overall operational performance. However, since many legacy applications (i.e., SolarWinds) may not work without these high levels of privilege, agencies should implement more layers of mitigation or discontinue the software's use in the environment. Allow lists, block lists, and grey lists should also be applied in tandem with endpoint privilege management to efficiently control application execution. Moreover, by applying context-rich privileged access security controls, such as content and application control rules and control over launch of child processes, agencies can protect trusted applications from misuse.

The security principle of "Individual Accountability" where privileged accounts are not shared, rather a privileged account is restricted to one user at a time, allows for clear oversight into what actions specific users/identities are performing with the elevated access. Also, agencies should apply, privilege separation, which involves defining and delineating employee, application, and system roles and tasks so that access is only granted to specific, discrete parts of systems or data, as necessary, and each account can only perform a specific, narrow range of actions. Separation of privileges helps contain intruders close to the point of compromise and restricts lateral movement. Thus, if one account is compromised, the range of privileges it affords the attacker is limited.

Multiple strategies for segmenting access can be applied to help enforce least privilege and enable zero trust principles. Segmentation efficaciously helps organizations, in broad strokes, protect the new attack surfaces created by digital transformation and cloud deployments. It is also an essential strategy for protecting sensitive software, applications, and environments, from encroaching on network-based technologies. Verifying every attempted access through proper identity controls with location awareness, identity authentication, MFA, and assuming every attempt at access is a threat actor until verified allow for an identity-centric approach to network segmentation. In this model, the network is not segmented necessarily by network or data link layer devices, it elevates the segmentation and authentication authority to the application layer. Application-level micro-segmentation should be implemented to prevent users from discovering applications they are not authorized to access.

One of the central tenants of zero trust architecture is the continuous and dynamic monitoring of identities that access the network. Agencies should monitor this architecture and manage any session involving privileged activity, whether by human, application, or machine. Monitoring should include performing screen recording command logging, scripts executed, and screen outputs. Anomalous activity should be flagged, with the ability to pinpoint, pause, and terminate suspicious sessions in real-time. Privilege analytics should also be performed to spot potential issues and to optimize a least privilege posture, such as removing unused privileges.

Securing critical software packages in a federal network must include a multi-tiered application of security practices to support continuity and integrity of the agency's mission. Applying appropriate configuration and hardening guidelines ensures the agency's network and software have a solid baseline that can withstand attacks using known vulnerabilities. Privileged access management and application control vastly reduce the risk of unauthorized access, and if a breach occurs, lateral movement is restricted. Zero trust architecture builds on the previous two practices, ensuring identities are verified every time a resource is accessed, thus mitigating the risk of an external actor breaching the network plane.

*Author and proposed speaker: Josh Brodbent, *Sr. Public Sector Security Director*, BeyondTrust
jbrodbent@beyondtrust.com, 870-362-6581