FS-ISAC

Hardening File Transfer Software

August 2023

Executive Summary

File transfer utilities are increasingly targeted by threat actors with sophisticated vulnerabilities and zero-days. These systems are a high-value target, given the volume and sensitive nature of data traversing them. File transfer solutions can be challenging to secure using typical control frameworks due to their opacity. Third-party built and/or hosted solutions may preclude teams from implementing controls.

Often managed by third parties, file transfer software (FTS) may lead to delays in patch application or other mitigations that would otherwise be overseen by in-house security teams. Threat actors could continue to see such solutions as a key access point to steal data. FS-ISAC members should harden these solutions where possible.

This document elucidates Defense-in-Depth strategies for risks associated with third-party FTS.

The Attacker's Advantage

- FTS creates an additional attack surface that may not be managed by internal IT or security teams.
- Since documents sent via FTS are often too sensitive for email, the likelihood of attackers capturing valuable data is high.
- Threat detection can be hampered by third-party management.
- With detection indicators being unknown, new zero-day vulnerabilities present more challenges and a much higher return to attackers that identify them.

Defense-in-Depth Approach

Defense-in-Depth is a cybersecurity best practice, integrating people, technology, and operational capabilities to establish layered barriers against malicious behavior. This approach uses alternative security controls to prevent or detect attacks that evade or bypass a single control. It also applies to third-party governance and the security assurances of vendor-provided solutions.

The extent to which teams can exercise controls depends largely on what vendors can, and are willing, to support. With this strategy, organizations can incorporate security controls in vendor-developed solutions.

A strategy must be comprehensive, adaptive, and proactive, aiming to address known threats and prepare for vulnerabilities. The following set of preventative, detective, and assurance controls provides a Defense-in-Depth strategy for resilience to such vulnerabilities or intrusions.

Response and Resilience

A robust security response involves various stakeholders to ensure the potentially compromised system is removed so that no further compromise or lateral movement is possible. This necessitates transitioning to an alternative business process to ensure continuity of operations.

A transition plan is crucial for business continuity and resilience, and must outline the steps required to deactivate and isolate the compromised system and activate the alternative process, with clear roles and responsibilities defined.

Preventative Controls

Reduce the Attack Surface

- Disable unnecessary functionalities, protocols, and modules in the software.
- Limit who/what can access the software by leveraging network allow lists, private connections, and/or mutual TLS authentication.
- Limit how long files/data is retained, especially on externally facing systems.
- Understand how critical third-party suppliers manage/harden their external attack surface.
- Use encryption services, including independent data-at-rest encryption.

Secure Coding Practices

- Design, write, test, and maintain code to prevent security vulnerabilities.
- Maintain the security of native source code and any third-party libraries leveraging testing modalities such as Threat Modeling, SAST, IAST, DAST, and SCA.

Application Isolation and Sandboxing

- Isolate an application from the system and other applications to prevent a chain infection.

- Sandbox by either:
  - Adding a management layer on the endpoint and limiting the application's ability to communicate with other applications or,
  - Isolating the application in a container, limiting the attack surface of a compromise.

Web Application Firewall (WAF)

- By filtering, monitoring, and blocking malicious web traffic, a WAF can prevent attacks from exploiting an underlying vulnerability.
- A WAF can be deployed as a cloud-based, hardware-based, software-based, or even container-based solution.
- As a WAF does not fully mitigate a vulnerable application, Runtime Application Self Protection (RASP) solutions can complement or offer similar benefits.

Network Architecture

- Network Segmentation and strict Demilitarized Zone (DMZ) segments for perimeter focused applications can help contain a compromised system.
- Strong inbound and outbound firewall restrictions can limit the type of post-exploitation, such as reverse shells, a threat actor can deploy.

Privileged Account Management (PAM)

- A PAM's central goal is to enforce the least privilege necessary of a user, account, application, or system to perform authorized activities.
- PAM maintains strong privilege controls to shrink an organization's attack surface and prevent adverse outcomes from successful exploits.

Patching

- Maintain updated inventories of software and hardware.
- Evaluate and apply security patches in a quick and practiced manner.
- For internally developed software, keep up with open-source libraries/frameworks updates.

Server Hardening Techniques

- Place advanced file system Access Control Lists (ACLs) on webroot path to disallow web server service account from writing or modifying files.

Assurance Controls

Secure by Design Architecture

- Data movement platforms should employ data at rest (DAR) encryption and decryption by default, with governance to validate process adherence to DAR requirements. Subsequent data decryption should occur outside the data movement platform.
- Platforms, especially those that have a measure of opacity regarding internal risk controls, should be isolated using a defined segmentation strategy.
- Access to these platforms should be restricted at the IP address level and/or mutual TLS authentication.
- Adopt a "known good" strategy, including FIM, EDR, DBAM, DLP, and WAF, and ensure vendors support it.

Security Validation & Governance

- Communicate expectations and requirements with vendors regarding their security practices and maintain strong governance practices. Assert expectations regarding:
  - Security testing, including penetration testing and source code analysis.
  - Secure code development and mature secure development life cycle (SDLC) governance.
  - A complete software bill of materials (SBOM) to support vulnerability identification and assessment of incorporated components.
- Repeated or systemic vulnerabilities in a vendor's software suggest latent risks in their entire product portfolio, necessitating a comprehensive review of all their solutions utilized in the environment or by suppliers.
- Exercise the firm's preventive, detective, and containment processes through Red/Purple Team exercises and/or tabletop exercises.

Conclusion

The controls described above can help provide a Defense-in-Depth posture against attacks such as those seen recently. While no single control can eliminate threat, layering sets of preventative, detective, and assurance controls can prepare an organization to withstand malicious activity.

