ACCELLION, INC. FILE TRANSFER APPLIANCE (FTA) EXPLOITS

Technical Report

Mark Ennis
October 6th, 2022

Executive Summary

This report covers two vulnerability chains in the Accellion, Inc.'s File Transfer Appliance (FTA) that resulted in two different attacks. Each vulnerability chain contains two separate vulnerabilities that allowed the attacker to create a malicious script. This script allowed the attacker to exfiltrate sensitive data and files from the compromised server. These exploits were of critical severity. They resulted in the loss of data integrity for users, downtime for servers running the FTA, and a loss of confidence in Accellion. In order to help prevent further attacks of this nature, several security principles should be followed. For developers, increased emphasis should be placed on sanitizing input that comes from an untrusted source. Additionally, increasing Access Control of different components within an application can help prevent attackers from successfully chaining vulnerabilities.

Introduction

Acellion, Inc. is an enterprise company that secures sensitive content. They provide services for secure email, APIs, and file sharing and sending. The Accellion, Inc. File Transfer Appliance (FTA) Security Assessment covers four exploits that were used in two separate but related attacks. The first attack happened in December 2020, and it shall be referred to as the December Exploit. The next attack occurred roughly one month later in January 2021. Similarly, it shall be referred to as the January Exploit. The cybersecurity firm Mandiant was contracted to perform forensic analysis and a security assessment of the FTA. Additional vulnerabilities found by Mandiant that were not exploited as part of the December and January attacks will not be covered by this report.

Since the attack Accellion, Inc. has rebranded to Kiteworks. For consistency, they will be referred to as Accellion for the remainder of this report. Additionally, the attack could have been perpetrated by one or more actors. For the purpose of this report, they shall be referred to as "the attacker".

Body

Discovery

The December Exploit was identified quickly. The web shell that was uploaded to the FTA server was placed in a location that triggered an alert by the built-in anomaly detector. Using that information, the available logs, and the web shell, Accellion was able to identify and patch both the vulnerabilities that were exploited.

The January Exploit was not discovered as quickly. The attacker appears to have learned from the previous attack. Another web shell was created as part of this attack, but it was uploaded to a location that did not trigger the anomaly detector. Without an alert triggered by a monitoring

system, it took Accellion several days and multiple customer service inquiries to determine that there were one or more new exploits affecting their systems.

Vulnerabilities

CVE-ID	Type	Description
CVE-2021-27101	SQL Injection	SQL injection vulnerabilities allow an attacker to run improper SQL queries. Generally, this vulnerability presents itself when input to the database is not properly sanitized, or when an application is not properly using a database library that uses prepared statements.
CVE-2021-27104, CVE-2021-27103	OS Command Execution	Operating System Command Execution (OSCE) vulnerabilities are a form of Arbitrary Code Execution (ACE) vulnerabilities that allow an attacker to execute remote commands. This allows attackers to take advantage of the tools that already exist on the vulnerable machine.
CVE-2021-27102	SSRF	Server Side Request Forgery vulnerabilities allow an attacker to get the server to make requests to unintended endpoints. This vulnerability can present itself when an input that forms URLs or URIs is not properly sanitized. An attacker can use SSRF attacks to extract restricted data from the server or other internal systems or cause the server to write data or files.

Exploits

The attack vector that led to the December Exploit began with CVE-2021-27101, a SQL Injection vulnerability. This vulnerability was a result of improperly sanitized database input pulled from the host header in a request to `document_root.html`. Using this vulnerability, the attacker was able to pull keys from the database. Using these keys, they created valid authentication tokens. With these tokens, they could make additional requests to `sftp_account_edit.php` that ran system commands. While performing this OSCE attack, the attacker was able to create a web shell. A web shell is a malicious script that functions as a backdoor. The attacker can use the web shell to run further commands. The attack path followed the vulnerability chain starting with the SQL injection vulnerability, to the OSCE vulnerability which allowed them to create their own malicious script. Once the web shell was in place they were able to exfiltrate data at will. Additionally, the script took steps to clean up after

itself by sanitizing and deleting logs. This was somewhat successful in that it is not clear what mechanism was used to write the web shell to the disk.

The January Exploit followed a different vulnerability chain to install a variant of the DEWMODE web shell. The January Exploit attack vector began with the SSRF vulnerability. They were able to exploit the vulnerability on the page `wmProgressstat.html` to access an internal web service. The internal SOAP web service contained an OSCE vulnerability that allowed the attacker to create a modified version of the DEWMODE web shell on the compromised machine.

It is worth noting that the attacker was able to exploit the OSCE vulnerability in the SOAP web service not only because of the SSRF vulnerability but also because the web service was not properly authenticating calls. This is likely because it was considered secure because it was not externally available. If the SOAP web service had tighter Access Control, the vulnerability chain may have ended at the SSRF vulnerability.

Severity

Both pairs of exploits were of critical severity. The attacker did not need to be authenticated to exploit the vulnerabilities. If the attacker could access an FTA server that was connected to the internet, they would be able to perform OSCE, as well as access sensitive data in the application database and download sensitive files. This attack successfully compromised the confidentiality of the compromised systems. Until the vulnerabilities were patched, there was no defence against the attack except shutting the server down. The only way to mitigate the attack was to shut off availability completely.

Given the fact that attackers were able to successfully exploit OSCE vulnerabilities, write files to disk, run system commands, and obfuscate some of their actions by altering and deleting local log files, the exploited machines would have to be considered compromised until a forensic analysis could be performed. In this way, the exploits also compromised the integrity of the system.

Additionally, these exploits required no special action from users. If they could locate a public FTA instance, they could perform the attack.

Using the FIRST CVSS calculator, the attack has a score of 9 / 10.

Impact

Presumably, FTA servers would contain sensitive, proprietary files as a secure file-sending application. Both the December and January exploits resulted in the attackers uploading variations of the DEWMODE web shell. The attacker could then export a list of files, their location, the identity of their uploader, and their recipient. The attacker could then download files located on the server.

After the January Exploit, Accellon sent out a message to their customers requiring them to shut down the FTA for three days. Currently, Accellion has large corporations such as Target, Shell, and Hewlett Packard Enterprises. If their current list of customers is similar to the customers they had at the beginning of 2021, even a small disruption that lasted several days could have cost millions of dollars.

Furthermore, Accellion had to procure the services of the cybersecurity firm Mandiant. Taking on the services of outside consulting experts helped ensure that any related vulnerabilities would be discovered and patched. Hiring a cybersecurity firm to assess their application also helps restore some of the reputational damage that Accellion may have suffered.

Reconnaissance

Passive Reconnaissance

Although the security assessments referred to internal APIs and token generation, it is possible that the attacker first familiarized themself with public-facing developer documentation.

It seems likely that the attacker gained access to Accellion FTA source code, or was able to decompile some binaries to reverse engineer more of the source code. From the provided security assessment, it can be determined that the FTA was written in PHP. Since PHP is interpreted rather than compiled, there are no binaries to decompile.

It may be possible for an attacker to identify vulnerable systems through some methods of passive reconnaissance. Similar to many other enterprise-grade companies, many of Accellion's partners, plugins, and customers are featured publicly on their site as a marketing exercise. From there, a dedicated attacker may be able to locate targets running the FTA by following a trail of marketing pages and press releases. Once those targets have been identified, forms of active reconnaissance can be applied.

Active Reconnaissance

There are several ways that the attacker could have identified where public-facing FTA applications were located. Web crawlers are constantly trawling the web. Web crawlers start from a list of known URLs and follow hyperlinks embedded in different pages. Once an attacker has located a URL, they can use a Web Scanner to help find a list of the common endpoints that are available. If certain FTA endpoints are present, the attacker has successfully fingerprinted the application and can perform the attack.

If the application was running on a non-standard port, the attacker may have also had to use a port scanning tool to determine which ports on the server are open and running applications. In

this case, the entry points in the vulnerability chains that led to the exploits were both executed over HTTP or HTTPS so it is likely they were running on a standard port.

Since all of the vulnerabilities present were zero-day vulnerabilities there was no need for the attacker to identify the version of FTA that the servers were running. If they could successfully identify a running server they could exploit the vulnerabilities.

Conclusion

These vulnerability chains could have been prevented if two security principles were followed during development. The first is that input from an external source should not be trusted. In the case of both SQL Injection and SSRF attacks, the back-end was not properly sanitizing input. If the database query that was exploited was executed using prepared statements, and the input was checked for illegal characters, the attacker would not have been able to access the database to create authorization tokens. In the case of the SSRF, the server provided access to an internal web service because the server was not properly handling URLs or endpoints sent from the front-end. In both cases, treating input from the front-end as malicious and checking it more thoroughly would remove the vulnerabilities. Secondly, if the internal web service accessed during the January Exploit had employed stricter Access Control, the attacker would not have been able to chain vulnerabilities successfully. If the web service treated all requests as untrusted, even those from other internal services, it may have forced the attacker to find a way to authenticate themselves. This would have made the attack more difficult, if not prevented it. These are prime examples of the importance of educating developers on secure development principles.