## **CVE-2017-5638**

## **(aka: Apache Struts Remote Code Execution (RCE))**

**Vendor Description**

* The Apache Software Foundation (ASF) is a non-profit organization that provides support for open-source software projects. It was founded in 1999 and is headquartered in the United States.
* The ASF follows a collaborative and community-driven approach to software development, where projects are developed and maintained by a diverse group of volunteers from around the world.
* The mission of the Apache Software Foundation is to provide software for the public good through the development and maintenance of open-source projects.
* The ASF hosts and manages over 350 open-source projects, covering a wide range of domains, including web servers, databases, big data and analytics, machine learning, cloud computing, mobile applications, scientific computing, and more.

**Affected Product Description**

* Apache Struts is an open-source framework for creating enterprise-ready Java Web Applications.
* It was donated to the Apache Foundation in May 2000. Formerly located under the Apache Jakarta Project and Known as Jakarta Struts, it became a top-level Apache project in 2005.
* Apache Struts uses and extends the Java Servlet API to encourage developers to adopt a model-view-controller (MVC) architecture, in which each architectural component is built to handle a specific development aspect of an application.

**Affected Feature Description**

* Jakarta Multipart Parser is a part of the Java Servlet API, which is used for processing HTTP requests and responses in Java web applications.
* The Jakarta Multipart Parser provides a convenient way to handle multipart data in Java web applications and allows developers to easily implement features such as file uploads in their applications.
* A multipart parser is a software component or tool designed to process or parse multipart data (data in a single message that consists of multiple parts or sections which is a way of structuring data in a single message that consists of multiple parts or sections. Each part can contain different types of data, such as text, binary data, or multimedia content. In the context of web applications, a multipart parser is commonly used to handle file uploads

**Technical Description**

The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, and that allows remote attackers to execute arbitrary commands via a crafted Content-Type, Content-Disposition, or Content-Length HTTP header.

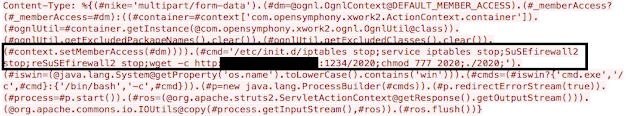
**What does it look like?**

**​​Simple Probing**



In this example the adversary is just running a simple command 'whoami' this could be done to see what user this service is running, ideally root. If a power user was identified the attacker could return with a more sophisticated set of commands.

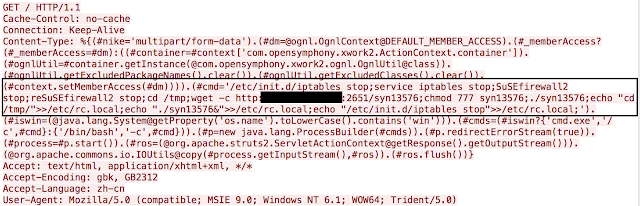
**Increased Sophistication**



This example is a little more aggressive with its attack. The steps include stopping the Linux firewall as well as SUSE Linux firewall. Final steps include downloading a malicious payload from a web server and execution of said payload. The payloads have varied but include an IRC bouncer, a DoS bot, and a sample related to the bill gates botnet.

**Sophistication with Persistence**

Below is another attack example that is similar to the previous example that downloads a malicious payload. The difference with this particular example is the attempted persistence. The adversary attempts to copy the file to a benign directory and then ensure that both the executable runs and that the firewall service will be disabled when the system boots.



**Severity and Impact**

* Common Vulnerability Scoring System Score – 10.0
* Confidentiality Impact – Complete (There is total information disclosure, resulting in all system files being revealed.)
* Integrity Impact – Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
* Availability Impact – Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
* Access Complexity – Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
* Authentication – Not required (Authentication is not required to exploit the vulnerability.)

**How can an attacker exploit the Apache Struts vulnerability?**

* Let unauthenticated attackers to get remote shell on server
* Allow run system commands on a remote server
* Total information disclosure, resulting in all system files being revealed
* Complete loss of system protection, resulting in the entire system being compromised
* Shutdown of the affected resource. The attacker can render the resource completely unavailable

**How to Fix or Mitigate the CVE-2017-5638 Vulnerability**

* Apply the patch: The Apache Software Foundation released a patch for the vulnerability shortly after its discovery. It's essential to apply this patch to vulnerable systems as soon as possible to prevent exploitation of the vulnerability.
* Upgrade to a non-vulnerable version: If the affected product is no longer supported or if it cannot be patched, it's recommended to upgrade to a non-vulnerable version of the software.
* Implement web application firewalls: Web application firewalls (WAFs) can help detect and block malicious requests that exploit the vulnerability. Implementing a WAF can provide an additional layer of defense against this and other types of attacks.
* Implement network segmentation: By implementing network segmentation, organizations can isolate vulnerable systems from other critical systems and limit the potential impact of a successful exploit.
* Implement least privilege access: Restrict access to vulnerable systems to only those users who require it to perform their job functions. This can help limit the attack surface and reduce the likelihood of a successful exploit.
* Monitor for suspicious activity: Monitor network and system logs for any suspicious activity that may indicate an attempted or successful exploit of the vulnerability. This can help detect and respond to attacks in a timely manner.
* Educate users: Educate users about the risks of phishing attacks and social engineering tactics that attackers commonly use to exploit vulnerabilities. This can help prevent users from inadvertently exposing the organization to attacks that exploit CVE-2017-5638 and other vulnerabilities.

**Case Study: 2017 Equifax Data Breach**

Timeline of Events:

* On March 7 2017, the Apache Software Foundation released a patch for the vulnerabilities; on March 9, Equifax administrators were told to apply the patch to any affected systems, but the employee who should have done so didn't.
* Equifax's IT department ran a series of scans that were supposed to identify unpatched systems on March 15; there were in fact multiple vulnerable systems, including the web portal, but the scans seemed to have not worked, and none of the vulnerable systems were flagged or patched.
* From May through July, the attackers were able to gain access to multiple Equifax databases containing information on hundreds of millions of people; a number of poor data governance practices made this possible.
* Equifax's attackers encrypted the data they were moving in order to make it harder for admins to spot; like many large enterprises, Equifax had tools that decrypted, analyzed, and then re-encrypted internal network traffic, specifically to sniff out data exfiltration events like this. But in order to re-encrypt that traffic, these tools need a public-key certificate, which is purchased from third parties and must be renewed annually. Equifax had failed to renew one of their certificates which meant that encrypted traffic wasn't being inspected.
* The expired certificate wasn't discovered and renewed until July 29, at which point Equifax administrators began noticing all that previously obfuscated suspicious activity; this was when Equifax first knew about the breach.
* It took another full month of internal investigation before Equifax publicized the breach, on September 8.
* It potentially affected 143 million people whose names, addresses, dates of birth, Social Security numbers, and drivers' licenses numbers were exposed. About 200,000 records also included credit card numbers.
* As soon as the Equifax breach was announced, infosec experts began keeping tabs on dark web sites, waiting for huge dumps of data that might be connected to it but the data never appeared. This gave rise to what's become a widely accepted theory: that Equifax was breached by Chinese state-sponsored hackers whose purpose was espionage, not theft.
* Why would the Chinese government be interested in Equifax's data records? Evidence of American officials or spies who are in financial trouble could help Chinese intelligence identify potential targets of bribery or blackmail attempts.
* In February of 2020, the United States Department of Justice formally charged four members of the Chinese military with the attack. This was an extremely rare move as the U.S. rarely files criminal charges against foreign intelligence officers in order to avoid retaliation against American operatives. That underscored how seriously the U.S. government took the attack.
* Two years after the breach, Equifax said it had spent $1.4 billion on cleanup costs, including "incremental costs to transform our technology infrastructure and improve application, network, [and] data security."
* In June 2019, Moody's downgraded the company's financial rating in part because of the massive amounts it would need to spend on infosec in the years to come.
* In July 2019, the company reached a record-breaking settlement with the FTC, which wrapped up an ongoing class action lawsuit and will require Equifax to spend at least $1.38 billion to resolve consumer claims.

Lessons:

* Get the basics right. No network is invulnerable. Equifax was breached because it failed to patch a basic vulnerability, despite having procedures in place to make sure such patches were applied promptly. And huge amounts of data was exfiltrated unnoticed because someone neglected to renew a security certificate. Equifax spent millions on security systems, but it was poorly implemented and managed.
* Silos are defensible. Once the attackers were inside the perimeter, they were able to move from machine to machine and database to database. If they had been restricted to a single machine, the damage would've been much less.
* Data governance is key especially if data is your business. Equifax's databases could've been stricter in giving up their contents. For instance, users should only be given access to database content on a "need to know basis"; giving general access to any "trusted" users means that an attacker can seize control of those user accounts and run wild. And systems need to monitor behavior; the attackers executed up to 9,000 database queries very rapidly, which should've been a red flag.