Example 3: Reporting RCE

`Title`: IBM WebSphere Java Object Deserialization RCE

`CWE`: CWE-502: Deserialization of Untrusted Data

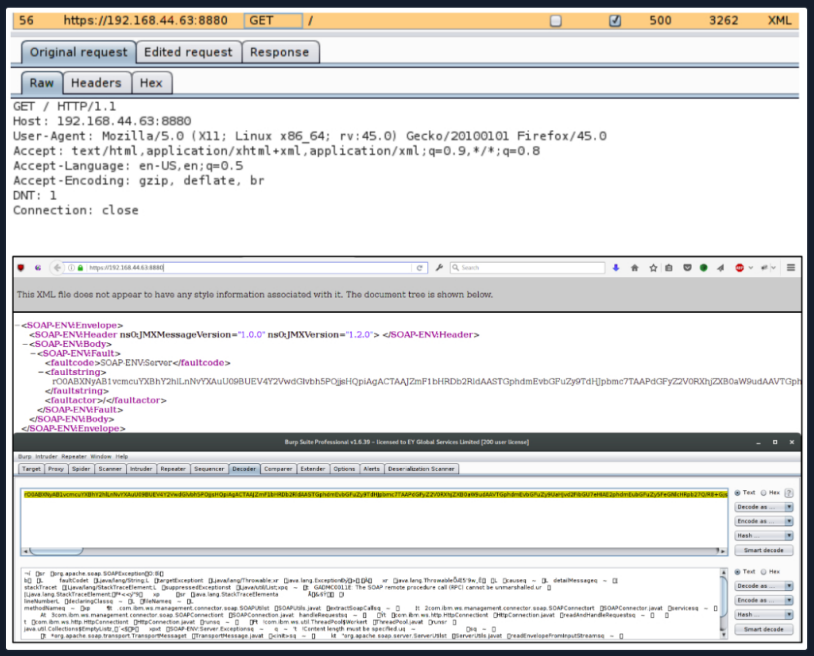
`CVSS 3.1 Score`: 9.8 (Critical)

`Description`: During our testing activities, we identified that the remote WebSphere application server is affected by a vulnerability related to insecure Java object deserialization allowing remote attackers to execute arbitrary commands. By issuing a request to the remote WebSphere application server over HTTPS on port 8880, we identified the existence of raw, serialized Java objects that were base64-encoded. It is possible to identify base64 encoded serialized Java objects by the "rO0" header. We were able to craft a SOAP request containing a serialized Java object that can exploit the aforementioned vulnerability in the Apache Commons Collections (ACC) library used by the WebSphere application server. The crafted Java object contained a `ping` command to be executed by the affected system.

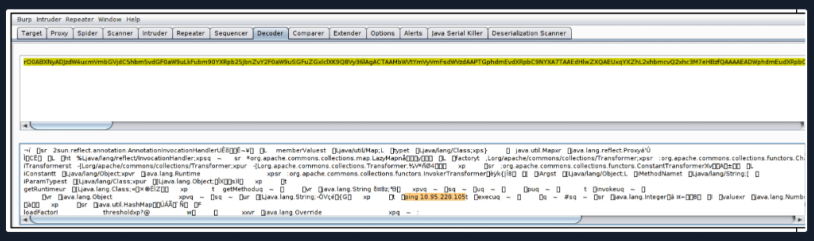
`Impact`: Command injection vulnerabilities typically occur when data enters the application from an untrusted source, such as a terminal or a network socket, without authenticating the source, or the data is part of a string that is executed as a command by the application, again without validating the input against a predefined list of allowed commands, such as a whitelist. The application executes the provided command under the current user's security context. If the application is executed as a privileged user, administrative or driver interface, such as the SYSTEM account, it can potentially allow the complete takeover of the affected system.

`POC`:

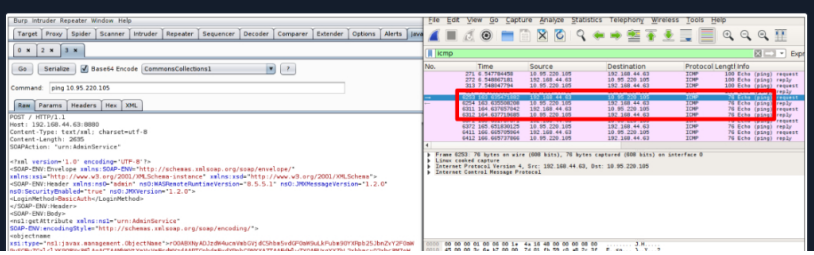
Step 1: We identified that the application uses serialized data objects by capturing and decoding a request to port 8880 of the server. The following images display the original request and the remote server's response, along with its decoded content.



Step 2: We crafted a SOAP request containing a command to be executed by the remote server. The command would send `ping` messages from the affected server to our host. The image below displays the crafted request and its decoded payload.



Step 3: The following image displays the crafted SOAP request allowing to remotely execute a `ping` command from the affected system. Capturing traffic via Wireshark, we observed the `ping` request from the Websphere application server to our machine.

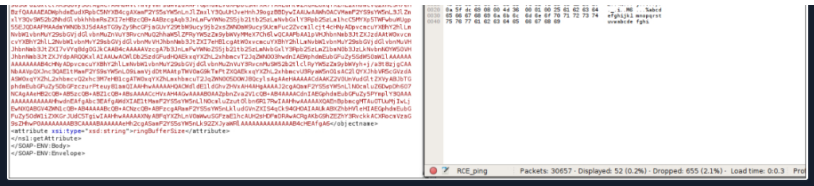


My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left



CVSS Score Breakdown

Attack Vector:	Network - The attack can be mounted over the internet.
Attack Complexity:	Low - All the attacker has to do is send a crafted request against the vulnerable application.
Privileges Required:	None - The attack can be mounted from an unauthenticated perspective.
User Interaction:	None - No user interaction is required to exploit this vulnerability successfully.
Scope:	Unchanged - Since the vulnerable component is the webserver and the impacted component is again the webserver.
Confidentiality:	High - Successful exploitation of the vulnerability results in remote code execution, and attackers have total control over what information is obtained.
Integrity:	High - Successful exploitation of the vulnerability results in remote code execution. Attackers can modify all or critical data on the vulnerable component.
Availability:	High - Successful exploitation of the vulnerability results in remote code execution. Attackers can deny the service to users by powering the webserver off.