**Phase 4. Threat Assessment**

**Hypothetical Threat Assessment Report of Nine Critical Scenarios of Artemis**

Tanvir Ahmed

Springboard

Cybersecurity

June 18, 2025

**Abstract**

Artemis, a natural gas corporation, has commissioned a comprehensive penetration testing and vulnerability assessment to evaluate the security of its digital infrastructure. This report examines nine cybersecurity vulnerabilities present in Artemis web application, network services, RDP, network infrastructure, cloud environment, and enterprise application. Each scenario highlights a distinct security flaw, the associated risks, and potential methods that attackers could use to exploit these weaknesses. The assessment evaluates the technical and operational impact of each vulnerability, and provides a qualitative risk rating. It would also include the mitigation strategies such as monitoring practice, patching, and proper configuration. Special mention to blocking mechanisms using IDS/ IPS will be highlighted in this report along with how to bypass them. Password cracking will be illustrated if encountered. This report is a structured approach to understand different vulnerability scenarios present in Artemis to assess, and mitigate attack vectors and improve the organization's security posture.

**Keywords:** threat assessment, RDP, SQL injection, Apache, password cracking,

**Scenario 1: Unpatched RDP is exposed to the internet**

* **Description:**

Remote Desktop Protocol (RDP) service without proper patching when exposed to the internet significantly increases the attack surface for threat actors. RDP provides a remote access experience that closely mimics physical logging into Windows, making it a valuable target.

* **OS / Version Effected:** Windows Xp, Windows 7, Windows Server 2008
* **Risk:**

Unpatched RDP leads to unauthorized access through exposed ports, most commonly, port 3389. This service often relies on a username and password, which can be bypassed with brute force attack and later useful for credential stuffing attacks. Once RDP login is bypassed, the threat actor can move laterally within the network and even penetrate the Apollo system. Additional risks include man in the middle attacks, where RDP session data can be intercepted and malware can be deployed on compromised hosts.

* **Potential Blocking Mechanism:** Defensive measures include Zero Trust Network Access, attack surface monitoring (ASM) tool, and firewall to block port 3389.

However, to bypass this:

* + **ZTNA**: Bypassed via stolen credential through phishing or installing malware
  + **ASM tools**: Bypassed using DNS tunneling by changing dns record rapidly and evading mapping.
  + **Firewall Blocking Port 3389:** Temporarily unlocking by sending network packets to that port.
* **Cracking:** Brute force technique with automated tools such as John the ripper, Hashcat, Ncrack, or THC Hydra to guess login credentials and gain unauthorized access.
* **Risk of attempting to exploit:**

With Repeated brute force attack can trigger account lockout and may crash the host

* **Remediation:** 
  + Enforcing strong and complex passwords.
  + Regularly patching the vulnerability by updating RDP software.
  + Disable RDP when not in use, enabling only when needed to access it.
  + Only allowing trusted devices to connect to the network and using VPN to create a secure tunnel for RDP connections.
  + Employ antivirus and anti malware tools to detect and block malicious operations related to RDP.
* **CVSS Score: 9.8 Critical**

**Scenario 2: Web application is vulnerable to SQL Injection**

* **Description:**

A Structure Query Language (SQL) injection vulnerability occurs when an attacker is able to manipulate backend SQL queries by inserting malicious input through components of a web application. Without proper sanitization or input validation, SQL queries get executed, enabling unauthorized access or manipulation of the database.

* **OS / Version Effected:** SQL injection vulnerabilities are platform independent, but commonly effect PHP, Javascript, [ASP.](http://asp.net)NET MySQL
* **Risk:**

Exploiting SQL injections can allow attackers to access sensitive information such as encryption keys from the database, user credentials, and root-level documents. Attackers may gain full read/write access to the database, impersonate administrators, and escalate privileges. Attackers can execute arbitrary commands, pivot into internal networks, or exfiltrate confidential data, if the database permits system level operations. Additionally, it enables attackers to delete records, corrupt databases, or disable critical functionality. If Apollo has a web interface or using API database of PostgreSql or Mysql then an attacker can exfiltrate trade secrets and manipulate records.

* **Potential Blocking Mechanism:** ORM designed to reduce SQL injections, Web Application Firewalls to filter SQLi payloads, Logging Monitoring to detect known patterns, Input Validation Whitelist that rely on tight regex
  + **ORM:** Bypassed using SQLi via interpolated input
  + **WAF:** Obfuscation and payload mutation with tools sqlmap and Burp Suite intruder
  + **Logging & Monitoring:** Rare payloads
  + **Whitelisting:** Input like admin and use of regex
* **Cracking:** Login authentication can be bypassed with SQL injection, then password hashes are extracted from the database and offline password cracking is performed with brute force technique, dictionary or rainbow tables. Tools such as John the Ripper, Hashcat or Hydra can be employed.
* **Risk of attempting to exploit:**

Can corrupt data and get detected by WAF or logging systems, attackers payload may be visible in logs.

* **Remediation:**
  + Educate developers, QA staff, DevOps teams, and SysAdmins on secure coding practice
  + Never Trust user input and always validate using whitelist and regex constraint
  + Use latest version and technology for PHP, PDO
  + Employ verified mechanism with parameterized queries or stored procedures
  + Scan web application regularly with tools such as Acunetix, and OWASP ZAP
  + Enable continuous monitoring and alerting for suspicious SQL behavior
* **CVSS Score:** **9.8 Critical**

**Scenario 3: Default password on Cisco admin portal**

* **Description:**

Cisco device and admin portals are shipped with default username and password such as admin / admin or cisco / cisco. If these default credentials are not changed during initial setup, they pose a significant risk. Attackers can easily find default credentials through public repositories allowing them to log in and gaining full administrative access.

* **OS / Version Effected:** Cisco Version 7.2
* **Risk:**

Potential threat is an attacker who successfully login with default credentials can authenticate with administrative privilege of the device or portal. This enables them to reconfigure network settings, add rouge devices to infiltrate the network, steal sensitive information and install malware. Even redirect traffic for man in the middle attacks, disable security features and establish persistent access for future intrusion. Also, possible for remote exploitation without the internal network access.

* **Potential Blocking Mechanism:** MFA, Password Policy Endorsement to change default passwords, Firewall to restrict access to admin portal, IDS/IPS
  + **MFA:** Bypassed via phishing attacks to steal tokens or session cookies
  + **Password policy Endorsement:** Brute force attack and credential stuffing attack if weak password
  + **Firewall:** IP spoofing to appear as allowed IP address
  + **IDS/IPS:** Payload obfuscation and encrypted traffic or rate limiting evasion can bypass detection
* **Cracking:**

Using THC Hydra or Ncrack to directly attack with brute force technique. Using John the Ripper for offline hash cracking. Credential stuffing for previously leaked usernames and password from breached databases.

* **Risk of attempting to exploit:**

Getting detected through logging logs and admin alert due to repeated failed login

* **Remediation:**
  + Enable multi factor authentication for a more secure layer.
  + Immediately change default username and password as soon as possible.
  + Restrict network access to the admin using IP whitelisting, VPNs, or SSH tunnels
  + Use alternative authentication mechanisms like kerebrose or PKI
  + Using SIEM and logging tools for continuous monitor
* **CVSS Score : 9.8 Critical**

**Scenario 4: Apache web server vulnerable to CVE-2019-0211**

* **Description:**

Privilege escalation vulnerability in Apache HTTP server, affecting Unix-based systems. This flaw allows anyone who doesn’t have privilege or less privilege to execute arbitrary code with parent process privilege. It is dangerous for multi-user environments where attackers can upload and run malicious scripts.

* **OS / Version Effected:** Apache HTTPD 2.4.17 to 2.4.38
* **Risk:**

The flaw allowed untrusted users to upload and run arbitrary scripts. Since there is no restricted privilege the uploaded script would run as root on the Apache web server. Once the script is on the web with root privilege, the system can easily be exploited. If the attackers gain access to the site, they would have full takeover of the server, and gain lateral movements within the hosting infrastructure. Even allow for persistent malware and modification of the website. If Apollo is hosted on an Apache server then an attacker can gain root privilege compromising the system. With full control the CIA triad is impacted making this vulnerability extremely significant.

* **Potential Blocking Mechanism: Mandatory Access Control (MAC), IDS**
  + **MAC:** Using misconfigured writing logs to bypass MAC rules
  + **IDS:** Bypass with payload obfuscation or encrypted tunnels
* **Cracking:** Not a password based vulnerability
* **Risk of attempting to exploit:**

May crash the web server and can be flagged by AV or EDR

* **Remediation:**
  + Upgrade Apache HTTP servion to later versions where the flaw is patched
  + Avoid running Apache as root
  + Prevent privilege sharing across users
  + Apply SELinux or app armor profiles to restrict script behavior
* **CVSS Score: 8.8 High**

**Scenario 5: Web server is exposing sensitive data**

* **Description:**

Sensitive data such as exposed .git folders, backup files, .bak .zip, configuration files, and even credit card numbers, health records, may be publicly accessible if exposed from web servers. This can lead to severe data leak and compromise organization security.

* **OS / Version Effected:** OS hosting a Web server
* **Risk:**

Sourceexposure can lead to stealing intellectual property, and unauthorized source code modifications. With this exposure other vulnerabilities can also be leaked. Credentials exposure can happen, leading to account compromise, and misuse of services. Attackers may use sensitive data for fraudulent activity or to access other services. Passwords, credit card numbers, health records, personal information, and business secrets, all of these information can leak, causing a big data leak. With so much at risk the effect can be catastrophic.

* **Potential Blocking Mechanism:** Web Server configuration, Access Control Lists (ACL)
  + **Web Server Configuration:** Bypassed with misconfiguration or missing deny rules
  + **ACL:** Bypassed with default permission
* **Cracking:** Using GitDumper to clone leaked .git directories, Gobuster to brute force hidden paths and files. John the Ripper or Hashcat for password hashes discovered in leaked data
* **Risk of attempting to exploit:**

Possibility of getting flagged by data loss prevention detection

* **Remediation:**
  + Never store sensitive data in web root, especially config and backup files
  + Encrypt sensitive data at rest and in transit using HTTPS and TLS 1.2
  + Disable caching for response to contain sensitive data
  + Store passwords using strong adaptive hashing algorithm like PBKDF2, and Argon2
  + Restrict access hidden system files with server configuration
  + Using cryptographic randomness for more secure encryption
  + Use authenticated encryption such as AES
* **CVSS Score: 7.5 High**

**Scenario 6: Web application has broken access control**

* **Description:**

Any user who isn’t authorized accessing admin functions or viewing private data. This includes viewing and modifying others' private data.

* **OS / Version Effected:** Application with role based access control
* **Risk:**

Principle of least privilege gets violated and can be accessed by any user who can bypass authorization by modifying the request url. Attackers can use attacking tools, modifying API requests. This would permit viewing or editing someone else’s account by providing direct object references. Accessing the API gives the ability to post, delete and put, for missing access controls. Json metadata may be abused by altering claim values or omitting signature validation. Privilege escalation may occur, granting regular users with administrative access.

* **Potential Blocking Mechanism: URL path restriction, WAF**
  + **URL path restriction**: Bypassed with modified user in query parameters
  + **WAF:** Payload obfuscation, API enumeration using Burp Suite
* **Cracking:**

IDOR exploitation by changing user in requests and tampering JWT or by sending crafter requests in Burp Suite to elevate roles or access other users data.

* **Risk of attempting to exploit:**

API tampering can be detectable and logs may reveal unauthorized access.

* **Remediation:**
  + Deny by default except for public resources.
  + Implement access control mechanisms and reuse them through the application.
  + Enforce record ownership rather than accepting for user to create, read, update, or delete
  + If log access fails alert admin
  + Stateful sessions should be invalidated after logout.
* **CVSS Score: 8.6 High**

**Scenario 7: Oracle WebLogic Server vulnerable to CVE-2020-14882**

* **Description:**

This critical vulnerability allows unauthenticated remote code execution (RCE) via crafter HTTP request to the admin console. Exploitation requires no authentication and no input validation. This flaw can allow attackers to execute arbitrary code.

**OS / Version Effected:** Oracle Weblogic 10.3.6.0 0, 12.1.3.0.0, 12.1.3.0.0, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.0

* **Risk:**

This type of vulnerability is easily exploitable with publicly available proof of concept (PoC). Used in active mass scanning campaigns in IPv4 space. Attackers can deploy, facilitate ransomware access and install webshells to exfiltrate sensitive data. If they gain access to the server, the threat actor can move laterally into the enterprise network.

* **Potential Blocking Mechanism: WAF, IPS/IDS, Network Segmentation**
  + **WAF:** Obfuscate request path
  + **IPS/IDS:** Bypass using encryptedHTTPS traffic or time delayed payload injection
  + **Network Segmentation:** Bypass using public exposure of admin port
* **Cracking:**

After gaining access, dump memory may contain WebLogic admin passwords in plain or encoded, if encoded then john the ripper or hashcat can be used to crack it.

* **Risk of attempting to exploit:**

Risk of crash from malformed HTTP and malware deployment can be flagged

* **Remediation:**
  + Immediately patching the vulnerability.
  + If unable to patch then ensure the admin portal is not exposed to the public internet.
  + Restrict access to /console and admin ports from public or external networks
  + Use network level ACL to control access to WebLogic admin
  + Review application logs for HTTP requests.
  + Monitor traffic for suspicious HTTP requests.
* **CVSS Score: 9.8 Critical**

**Scenario 8: Misconfigured cloud storage (AWS security group misconfiguration, lack of**

**access restrictions)**

* **Description:**

Misconfigurations in cloud environments, such as insecure AWS security groups, unrestricted permission or lack of IAM policies can expose sensitive services and data to the internet unintentionally.

* **OS / Version Effected: AWS infrastructure**
* **Risk:**

Misconfigured clouds significantly elevate the risk of unauthorized access, data exfiltration, and compliance violations. It opens a path for attackers to access cloud based data, steal it, ransom it and even install digital skimming code. Digital skimming code can capture sensitive information from web apps such as credit card number, account number, social security number and send it to the hacker’s controlled server. Those data then can be sold for money or used for illegal activity. If Apollo stores data in the cloud then weak policies of IAM can cause attackers to extract confidential manufacturing data. Attackers can also deploy malware, crypto mines, or command and control backdoors on misconfigured instances. Misconfiguration cloud environments can attract hackers to move laterally within the cloud environment leading to violations of GDPR, HIPAA or PCI DSS.

* **Potential Blocking Mechanism:** AWS IAM Policy, Firewall, Cloud Security Posture Management (CSPM)
  + **AWS IAM Policy:** Bypass with stolen access tokens
  + **Firewall:** Misconfigured ports or ranges
  + **Cloud Security Posture Management (CSPM):** Bypass with hidden IAM users or disabling logging temporarily with stop CloudTrail logging with stolen admin credentials
* **Cracking:**

Brute forcing with enumeration tool such as AWSBucketDump and s3canner using token stolen from exposed logs or leaked credentials

* **Risk of attempting to exploit:**

Data exfiltration can trigger DLP or SIEM alerts.

* **Remediation:**
  + Implement logging practice to manage the number of users making changes in the cloud environment.
  + Use bucket policies and access control list (ACL) to prevent public access
  + Enforce Principle of Least Privilege (PolP) for all IAM roles
  + Enabling encryption to protect from unauthorized data viewing.
  + Limit permission to only individuals who need it for their job.
  + Regular audit needs to be implemented to monitor misconfiguration and other cloud based threats.
  + Continuous monitor with tools like AWS config and AWS GuardDuty
  + Integrate strong policy to build or enhance cloud infrastructure.
  + Automate remediation with Kaspersky, Bitdefender to alert misconfigurations.
* **CVSS Score: 8.5 High**

**Scenario 9: Microsoft Exchange Server vulnerable to CVE-2021-26855**

* **Description:**

Is a server side request forgery (SSRF) vulnerability in which it allows unauthenticated attackers to bypass authentication and impersonate administrators by sending crafter HTTP requests to Exchange servers.

* **OS / Version Effected:** Microsoft Exchange 2013, 2016, 2019
* **Risk:**

Unauthenticated remote attackers can exploit this flaw by crafting a HTTP request to a vulnerable Exchange server. With access to the server, an attacker can steal the full contents of users' mailboxes and the IP address they wish to target from the Exchange Server. Exploitation of this flaw can allow an unauthenticated attacker to execute arbitrary code to gain persistent system access and pivot to another system within the network.

* **Potential Blocking Mechanism:** WAF, IDS/IPS, VPN
  + **WAF:** Bypass using payload obfuscation
  + **IDS/IPS:** Bypass using encrypted traffic
  + **VPN:** Bypass with compromised credential or internal endpoint
* **Cracking:**

If credential are extracted from mailboxes then tools like John the Ripper Hashcat can be used to brute force account hashes

* **Risk of attempting to exploit:**

Microsoft Defender may trigger ATP alerts, service instability and detection from log anomaly

* **Remediation:**
  + Patch the vulnerability immediately.
  + Scan for web shells
  + Implement MFA for all Exchange user accounts
  + Examine systems for TTPs and use IOCs to detect any malicious activity.
  + Set up a VPN to separate the Exchange Server from external access.
  + Ensure regular auditing and logging for unusual activity in admin server
* **CVSS Score: 9.8 Critical**

**Conclusion**

After a careful threat assessment of Artemis network, many threats were identified and examined with systemic through a systematic procedure. A variety of critical vulnerabilities rangin from high to critical severity were found. These include, misconfigured cloud environments, unpatched RDP services, broken access control, unauthenticated RCE in Oracle WebLogic , sensitive data exposure, default admin credentials, Apache HTTP server of privilege escalation and Microsoft Exchange Server SSRF vulnerability.. Each scenario was systematically analyzed, outlining specific risks, exploitation strategies, mitigation techniques. Password cracking methods were also examined, along with blocking mechanisms that can be bypassed. The assessment highlest techniques to mitigate risk and enhance the security posture of Artemis. Implementing these strategies will help maintain the confidentiality, integrity and availability of the Artemis network.

**Reference**

Acunetix. (n.d.). *What is SQL injection (SQLi) and how to prevent it*.<https://www.acunetix.com/websitesecurity/sql-injection/>

Apache Software Foundation. (2024). *Apache HTTP Server 2.4 vulnerabilities*.<https://httpd.apache.org/security/vulnerabilities_24.html#:~:text=(CVE-2024-38475),execution%20or%20source%20code%20disclosure>

Cisco. (2011, June 1). *Cisco Network Registrar Denial of Service Vulnerability*. Cisco Security Advisories.<https://www.cisco.com/c/en/us/support/docs/csa/cisco-sa-20110601-cnr.html>

Cloudflare. (n.d.). *RDP security risks*.<https://www.cloudflare.com/learning/access-management/rdp-security-risks/>

Cybersecurity & Infrastructure Security Agency. (2021, March 3). *AA21-062A: Mitigate Microsoft Exchange Server vulnerabilities*. CISA.<https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-062a>

Fortect. (n.d.). *Remote Desktop Protocol (RDP): Risks and how to protect your Windows PC*.<https://www.fortect.com/how-to-guides/remote-desktop-protocol-risks-and-how-to-protect-your-windows-pc/#:~:text=Risks%20of%20Remote%20Desktop%20Protocol,unauthorized%20access%20to%20your%20system>

Imperva. (2020, November 4). *Bug hunting for a quick buck using WebLogic vulnerability CVE-2020-14882*.<https://www.imperva.com/blog/bug-hunting-for-a-quick-buck-using-weblogic-vulnerability-cve-2020-14882/>

Red Hat. (2019). *CVE-2019-0211 - Apache HTTP Server privilege escalation vulnerability*. Red Hat Customer Portal.<https://access.redhat.com/security/cve/cve-2019-0211>

Tenable. (2020, November 2). *CVE-2020-14882: Oracle WebLogic remote code execution vulnerability exploited in the wild*.<https://www.tenable.com/blog/cve-2020-14882-oracle-weblogic-remote-code-execution-vulnerability-exploited-in-the-wild>

Tenable. (n.d.). *Apache HTTP Server < 2.0.55 mod\_imap Arbitrary File Disclosure*.<https://www.tenable.com/plugins/nessus/23938>

Trend Micro. (2019, April 4). *CVE-2019-0211 patched: Apache HTTP Server root privilege escalation flaw a priority for web hosting providers*.<https://www.trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/cve-2019-0211-patched-apache-http-server-root-privilege-escalation-flaw-a-priority-for-web-hosting-providers>