Sri Lanka Institute of Information Technology

Web Security - IE2062

Bug Bounty Report 6

PERERA A.P.J

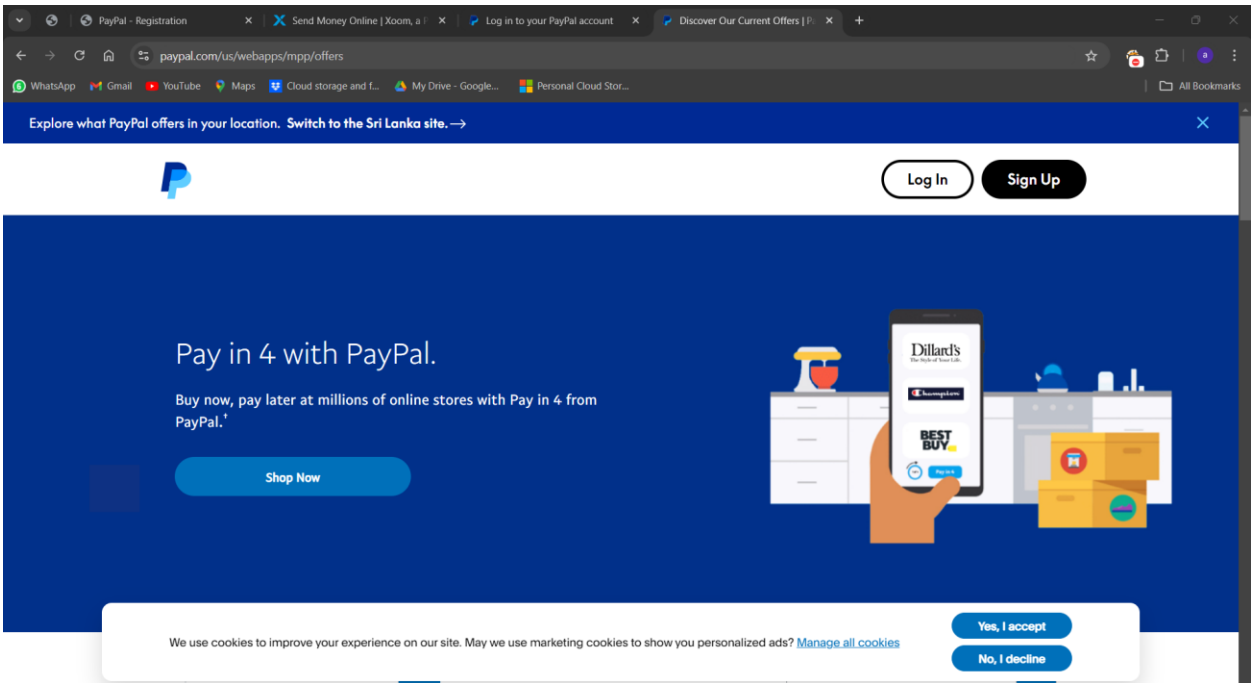
IT22280992

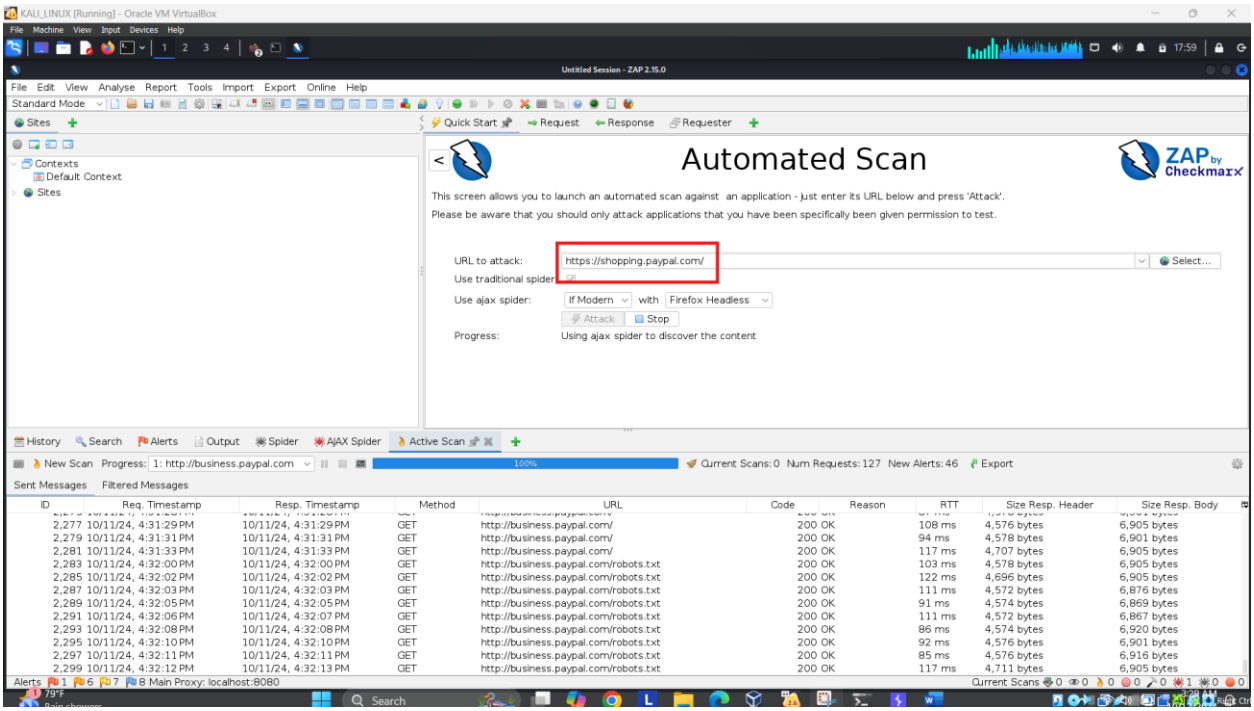
Group Y2S2.CS

SLIIT
Discover Your Future

Table of Contents

1. **TARGET: https://shopping.paypal.com/**

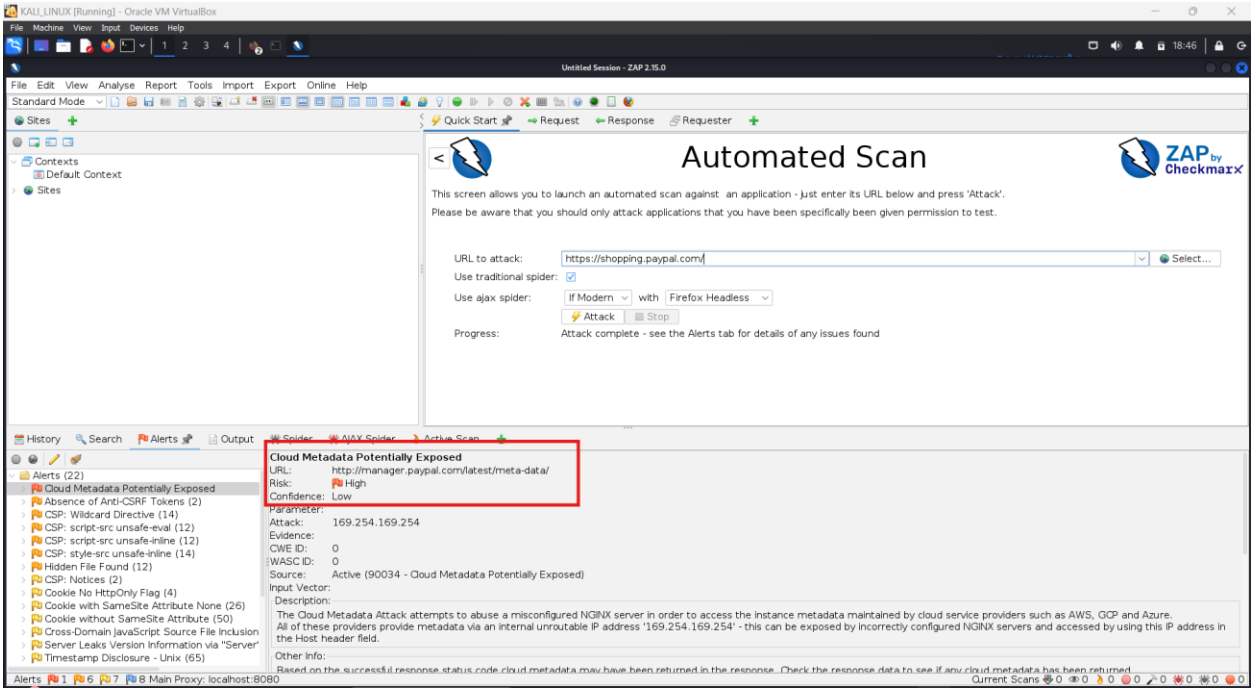



KALI_LINUX [Running] - Oracle VM VirtualBox
Untitled Session - ZAP 2.15.0
Automated Scan
ZAP by Checkmarx
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack: https://shopping.paypal.com/
Use traditional spider:
Use ajax spider: If Modern with Firefox Headless
Attack Stop
Progress: Using ajax spider to discover the content
History Search Alerts Output Spider AJAX Spider Active Scan
New Scan Progress: 1: http://business.paypal.com 100% Current Scans: 0 Num Requests: 127 New Alerts: 46 Export
Sent Messages Filtered Messages
ID Req. Timestamp Resp. Timestamp Method URL Code Reason RTT Size Resp. Header Size Resp. Body
2,277 10/11/24, 4:31:29 PM 10/11/24, 4:31:29 PM GET http://business.paypal.com/ 200 OK 108 ms 4,576 bytes 6,905 bytes
2,279 10/11/24, 4:31:31 PM 10/11/24, 4:31:31 PM GET http://business.paypal.com/ 200 OK 94 ms 4,578 bytes 6,901 bytes
2,281 10/11/24, 4:31:33 PM 10/11/24, 4:31:33 PM GET http://business.paypal.com/ 200 OK 117 ms 4,707 bytes 6,905 bytes
2,283 10/11/24, 4:32:00 PM 10/11/24, 4:32:00 PM GET http://business.paypal.com/robots.txt 200 OK 103 ms 4,578 bytes 6,905 bytes
2,285 10/11/24, 4:32:02 PM 10/11/24, 4:32:02 PM GET http://business.paypal.com/robots.txt 200 OK 122 ms 4,696 bytes 6,905 bytes
2,287 10/11/24, 4:32:03 PM 10/11/24, 4:32:03 PM GET http://business.paypal.com/robots.txt 200 OK 111 ms 4,572 bytes 6,876 bytes
2,289 10/11/24, 4:32:05 PM 10/11/24, 4:32:05 PM GET http://business.paypal.com/robots.txt 200 OK 91 ms 4,574 bytes 6,869 bytes
2,291 10/11/24, 4:32:06 PM 10/11/24, 4:32:07 PM GET http://business.paypal.com/robots.txt 200 OK 111 ms 4,572 bytes 6,867 bytes
2,293 10/11/24, 4:32:08 PM 10/11/24, 4:32:08 PM GET http://business.paypal.com/robots.txt 200 OK 86 ms 4,574 bytes 6,920 bytes
2,295 10/11/24, 4:32:10 PM 10/11/24, 4:32:10 PM GET http://business.paypal.com/robots.txt 200 OK 92 ms 4,576 bytes 6,901 bytes
2,297 10/11/24, 4:32:11 PM 10/11/24, 4:32:11 PM GET http://business.paypal.com/robots.txt 200 OK 85 ms 4,576 bytes 6,916 bytes
2,299 10/11/24, 4:32:12 PM 10/11/24, 4:32:13 PM GET http://business.paypal.com/robots.txt 200 OK 117 ms 4,711 bytes 6,905 bytes
Alerts 1 6 7 8 Main Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 1 0 0

2. Vulnerability

2.1 Vulnerability title

- Cloud meta data potentially Exposed.



2.2 Vulnerability description

- The Vulnerability in Amazon Cloud Metadata exposure is a security issue that sits around the usage of sensitive information outside of their designated purposes, particularly cloud metadata to unauthorized users or services. Cloud platforms such as AWS, Microsoft Azure and Google Cloud store critical information like resources, configurations as well as user credentials within the metadata. Exposed metadata can be abused by perpetrators using Cross Site Scripting (XSS) or Server Side Request Forgery (SSRF) techniques to secrete information such as IAM roles, API keys and other crucial data. It can also lead to unlocking of cloud configurations and resources without the leased tenants consent which increases the chances of data infringement and security related issues. In addressing this threat, organizations are encouraged to implement strong access safeguards, apply strong IAM policies, and ensure that management nett nearly civil pinons as this limit facilitates metadata protection.

2.3 Affected components

The Cloud Metadata Potentially Exposed Vulnerability can affect several key components within a cloud infrastructure:

1. Metadata Services: The core component where metadata is stored and accessed. For instance, in AWS, the instance metadata service (IMDS) can be vulnerable if not properly secured.

2. Compute Instances: Virtual machines or containers that may inadvertently expose metadata through poorly configured applications or services running on them.

3. Network Configuration: Security groups, firewalls, or network access control lists (ACLs) that do not adequately restrict access to metadata services, allowing attackers to exploit vulnerabilities.

4. API Endpoints: Any application or service that interacts with cloud APIs can be affected if it inadvertently exposes metadata through insecure coding practices or insufficient input validation.

5. Identity and Access Management (IAM): Misconfigured IAM roles or policies can lead to excessive permissions, allowing unauthorized access to sensitive metadata.

6. Serverless Functions: Functions or services that run in response to events may have access to metadata, making them potential vectors for exploitation if not properly secured.

7. Third-party Integrations: External applications or services that integrate with cloud environments may inadvertently access or expose metadata, increasing the risk of exploitation.

2.4 Impact assessment

Impact assessment for the Cloud Metadata Potentially Exposed Vulnerability involves evaluating the potential consequences of an exploitation. The key impacts can include:

1. Unauthorized Access: Attackers may gain access to sensitive information, including credentials, API keys, and access tokens, which can be used to compromise other cloud resources.

2. Data Breach: Exposure of sensitive data stored within cloud services can lead to significant breaches, resulting in financial loss, reputational damage, and legal ramifications.

3. Resource Manipulation: With access to metadata, attackers may manipulate cloud resources, leading to service disruptions, unauthorized changes to configurations, or even the deployment of malicious resources.

4. Account Compromise: If attackers obtain IAM roles or service account credentials, they can impersonate legitimate users, escalate privileges, and further exploit the cloud environment.

5. Compliance Violations: Failure to protect sensitive metadata can lead to violations of regulations such as GDPR, HIPAA, or PCI-DSS, resulting in fines and increased scrutiny from regulatory bodies.

6. Operational Disruption: The incident response to a metadata exposure can cause significant operational disruption, requiring resources for investigation, remediation, and restoring services.

7. Reputational Damage: Organizations may suffer long-term reputational harm as customers lose trust in their ability to safeguard sensitive data, potentially affecting customer retention and acquisition.

2.5 Steps to reproduce

steps to Reproduce

1. Set Up an AWS Environment:
    - Launch an EC2 instance with default settings.
    - Ensure the instance has an IAM role with permissions to access specific AWS services (e.g., S3).

2. Access the Instance:
    - SSH into the EC2 instance using appropriate credentials.

3. Access Instance Metadata:
    - Use a command-line tool like curl or wget to access the instance metadata service by running:
    - This command should return various metadata items, such as instance ID, AMI ID, and IAM role details.

4. Extract Sensitive Information:
    - Access specific metadata endpoints to extract sensitive data. For example:
    - This will provide the IAM role associated with the instance.

5. Retrieve IAM Role Credentials:
    - If an IAM role is returned, use the following command to access the role's temporary security credentials:
    - This will reveal access key, secret key, and session token, which can be used to authenticate and make API calls to AWS services.

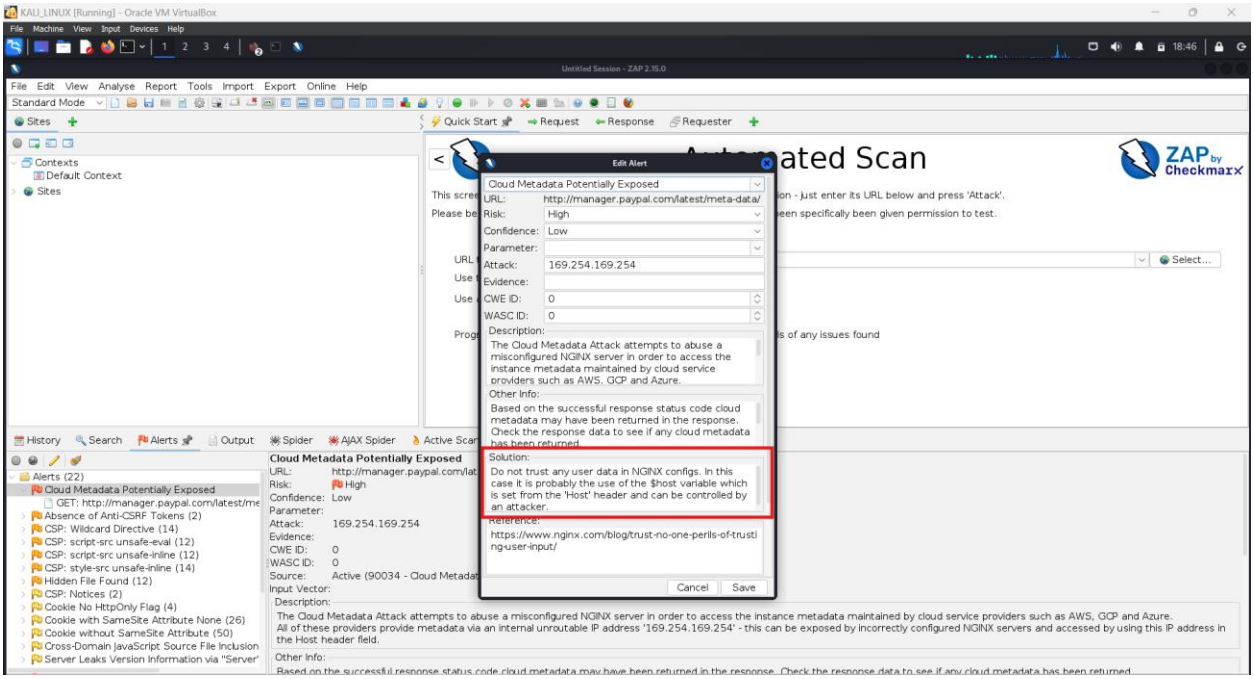
6. Test for Vulnerabilities:

    - Attempt to access other sensitive resources using the retrieved credentials, such as S3 buckets or RDS instances, to demonstrate the extent of the exposure.

7. Document Findings:

    - Record all commands and outputs, highlighting any sensitive information exposed during the process and discussing the potential impact on security.

2.6 Proof of concept



2.7 Proposed mitigation or fix

- Do not trust any user data in NGINX configs. In this case it is probably the use of the $host variable which is set from the 'Host' header and can be controlled by an attacker.
- Restrict Meta data Access.
- Implement Instance Metadata Service Version 2.
- Apply Principle of Least Privilege.
- Use Security Groups and Networks ACLs.
- Conduct Regular Security Audits.
- Implement Logging and Monitoring.
- Educate Developers and Administration.
- Use Encryption.