Damn Vulnerable AWS API

Garret Arp | Ashler Benda | Karthik Kasarabada | Andrew Bowen | Ahmed Nasereddin | Ayo Ogunsola | Ethan Douglass

Group #: sdmay24-11 Advisor: Dr. Julie Rursch Client: Jon Schnell on behalf of RSM

Introduction

- Our client provides cloud security testing as a consulting service to their customers
- We were tasked with creating a learning tool for AWS Pentesting that could also be used for skill assessment

Design Requirements

- Utilize AWS specific vulnerabilities and exploits
- Result in full account compromise
- Use Cloudformation Templates
- Minimum cost by using free tier resources

Included AWS Services

- API Gateway
- Relational Database Service (RDS)
- Simple Storage Service (S3)
- Elastic Compute Cloud (EC2)
- Identity and Access Management (IAM)
- Lambda Function
- Virtual Private Cloud (VPC)
- Systems Manager (SSM)
- AWS CLI

Intended Users

- New AWS users
- Risk Consultants
- IT Administrators
- Software Architects
- Cybersecurity Students

Use Cases

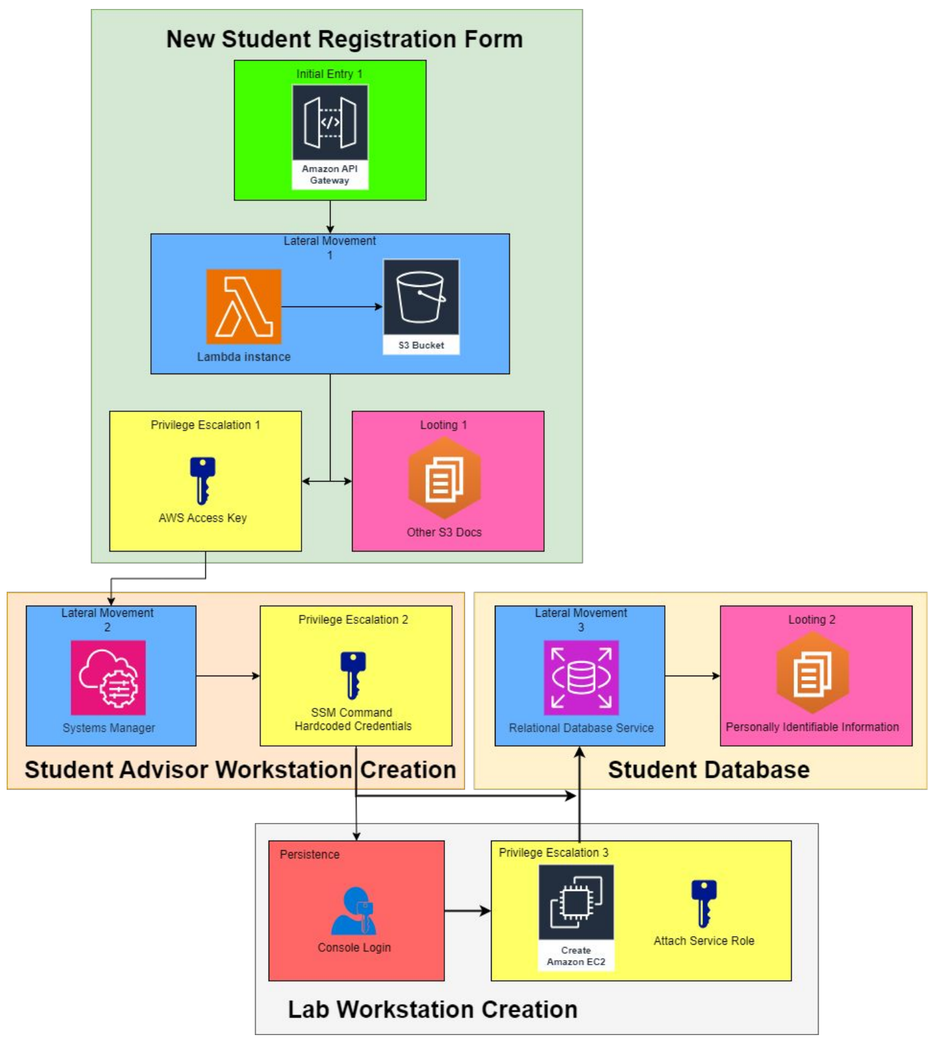
- Testing and Development
- Security
- Education
- Skill Analysis

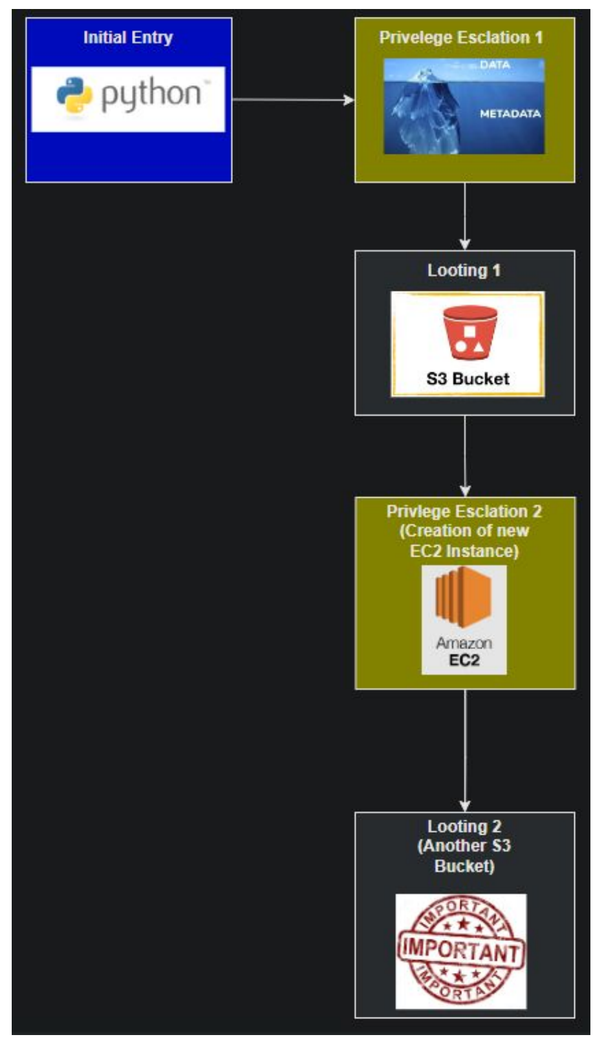
IaC Visualization

```
"Resources": {
    "PostDataRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "RoleName": "PostDataRole",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    },
                    {
                        "Sid": "Statement1",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "iam.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "PermissionsBoundary": "arn:aws:iam::308130987840:policy/Boundaries",
            "Path": "/",
            "Policies": [
                {
```

Design Approach

- Organization is based on a standard cyber attack flow
- Chosen medium was Infrastructure as Code to maximize impact through the resulting simplified setup and breakdown of the environment
- Two seperate attack paths to maximize educational content and minimize resource consumption
- Model real world systems, misconfigurations, and attacks





Technical Details

Attack Path 1:

- Initial access to a website that allows improperly scoped access to an S3 bucket that reveals an AWS Access Key
- Access Key allows enumeration of resources to identify backend database and improperly decommissioned System's Manager command document with credentials
- Credentials allow interaction with EC2 instance that can be passed a higher privilege role for full account compromise and looting of the database

Attack Path 2:

- Initial access to a Django web server which is leaking credential files through the exposed AWS metadata service
- Using the exposed credentials, resource enumeration reveals an older IAM policy version with poorly scoped permissions which users can revert to in order to escalate their privileges
- Additional permissions allow creation of an unsecure EC2 instance

Testing

All tests are performed in AWS environment provided by the client

- **Unit Testing** Each individual component as seen in final design iterations is tested to confirm the resources interact as intended
- **Integration Testing** The developer of each component begins integration testing to confirm that privilege escalation or resource communication works as intended with attached components
- **Full System Testing** All components in a given Attack Path are consolidated into a full stack for testing by following through the documented user walkthrough to validate all functional requirements

Resources (45)

Search resources

Logical ID	Physical ID	Type	Status
FillRDS	AP1-FullStack-FillRDS-XNUFFzmueGxk	AWS::Lambda::Function	CREATE_COMPLETE
InstanceSecurityGroup	sg-0bc8cfaaf65e3360b	AWS::EC2::SecurityGroup	CREATE_COMPLETE
InternetGateway	igw-01bd77061fe4fd728	AWS::EC2::InternetGateway	CREATE_COMPLETE
InternetGatewayAttachment	IGW|vpc-04cb35935b2d4055d	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE
LambdaExecutionRole	AP1-FullStack-LambdaExecutionRole-38zmZCMdjhzT	AWS::IAM::Role	CREATE_COMPLETE
NatGateway1	nat-059d2e9fe82927647	AWS::EC2::NatGateway	CREATE_COMPLETE
NatGateway1EIP	18.219.233.141	AWS::EC2::EIP	CREATE_COMPLETE
persistenceUser	bnoel	AWS::IAM::User	CREATE_COMPLETE
PopLambda	PopulateS3Lambda	AWS::Lambda::Function	CREATE_COMPLETE
PopulateLambdaRole	PopulateLambdaRole	AWS::IAM::Role	CREATE_COMPLETE
PostDataRole	PostDataRole	AWS::IAM::Role	CREATE_COMPLETE