Securing Terraform Deployments

KICS Vulnerabilities and Leveraging Auto-Remediation

Let's understand

1. Kics docker image Installation on my sand box (https://killercoda.com/)

2. vulnerable Terraform files saved for lab testing simple.tf

3.Results summary.

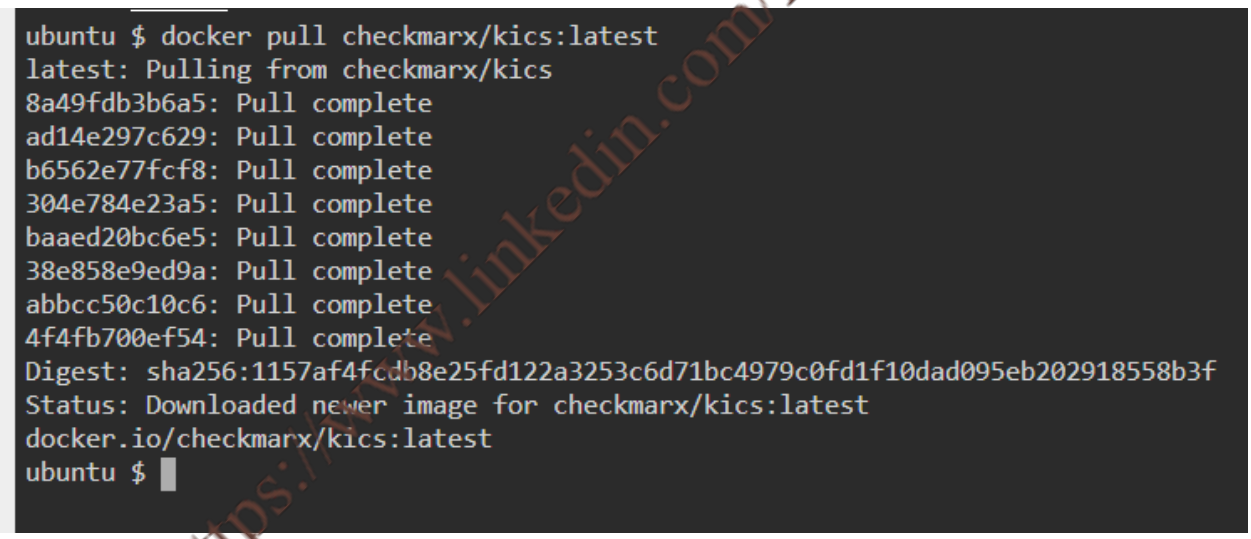
4.how to check the results on json format.

1. Kics docker images Installation on my sand box/local system

docker pull checkmarx/kics:latest

https://docs.kics.io/latest/documentation/ steps available on as official documentation.

Hardcoded Credentials



2. vulnerable Terraform files for lab testing.

vulnerable Terraform files for lab testing simple.tf with Hardcoded Credentials

```
provider "aws" {
 access_key = "YOUR_ACCESS_KEY"
 secret_key = "YOUR_SECRET_KEY"
 region     = "us-west-2"
}
```

https://www.linkedin.com/in/manitechy/

```
resource "aws_instance" "example" {
 ami           = "ami-0c94855ba95c71c99"
 instance_type = "t2.micro"

 tags = {
   Name = "vulnerable-instance"
 }
}
```

```
docker run -v /root:/path checkmarx/kics:latest scan -p "/path/simple.tf" -o "/root/results/"
```

```
ubuntu $ pwd
/root
ubuntu $ docker run -v /root:/path checkmarx/kics:latest scan -p "/path/simple.tf" -o "/root/results/"

            w
                .0MO.
                OMMMx
                ;NMX;
                 ...          ...             ....
WMMMd     cWMMM0.  KMMMO      ;xKWMMMMNOc.     ,xXMMMMMWXkc.
WMMMd   .0MMMN:    KMMMO    :XMMMMMMMMMMMWl   xMMMMMWMMMMMMl
WMMMd  lWMMMO.     KMMMO   xMMMMKc...'lXMk   ,MMMMx   .;dXx
WMMMd.0MMMX;       KMMMO  cMMMMd        '    'MMMMNl'
WMMMNWMMMMl        KMMMO  0MMMN               oMMMMMMMXkl.
WMMMMMMMMMo        KMMMO  0MMMX                .ckKWMMMMMM0.
WMMMMWokMMMMk      KMMMO  oMMMMc              .     .:OMMMM0
WMMMK.  dMMMM0.    KMMMO   KMMMMx'    ,kNc   :WOc.    .NMMMX
WMMMd    cWMMMX.   KMMMO    kMMMMMWXNMMMMMd .WMMMMWKO0NWMMMl
WMMMd     ,NMMMN,  KMMMO     'xNMMMMMMMNx,   .l0WMMMMMMMMwk,
xkkk:      ,kkkkx  okkkl        ;xKXKx;          ;dOKkkc


Scanning with Keeping Infrastructure as Code Secure v1.7.2


Preparing Scan Assets: DoneExecuting queries: [---->_____________________________________________________________] 5.80%Executing q
ueries: [---->_____________________________________________________] 6.11%Executing queries: [---->_____________________
_____________________________________] 6.50%Executing queries: [---->_____________________________________________________
_____] 6.73%Executing queries: [----->_____________________________________________________] 7.04%Executing queries: [-----
>_____________________________________________________] 7.35%Executing queries: [----->_____________________________________
_____________________] 7.74%Executing queries: [----->_____________________________________________________] 8.05%E
xecuting queries: [------>_____________________________________________________] 8.28%Executing queries: [------>___________
_____________________________________] 8.67%Executing queries: [------>_____________________________________________
_____________] 8.98%Executing queries: [------>_____________________________________________________] 9.21%Executing queri
es: [------>_____________________________________________________] 9.52%Executing queries: [------->_____________________
_____________________________] 9.67%Executing queries: [------->_____________________________________________________
__] 9.98%Executing queries: [------->_____________________________________________________] 10.29%Executing queries: [------->_
_____________________________________________] 10.60%Executing queries: [------->_____________________________________
_____________________] 10.91%Executing queries: [-------->_____________________________________________________] 11.22%Execu
```

https://www.linkedin.com/in/manitechy/

```
Passwords And Secrets - Generic Secret, Severity: HIGH, Results: 1
Description: Query to find passwords and secrets in infrastructure code.
Platform: Common
Learn more about this vulnerability: https://docs.kics.io/latest/queries/common-queries/common/3e2d3b2f-c22a-4df1-9cc6-a7a0aebb0c99

        [1]: ../../path/simple.tf:3

                002:   access_key = <SECRET-MASKED-ON-PURPOSE>
                003:   secret_key = <SECRET-MASKED-ON-PURPOSE>
                004:   region     = "us-west-2"


Passwords And Secrets - Generic Access Key, Severity: HIGH, Results: 1
Description: Query to find passwords and secrets in infrastructure code.
Platform: Common
Learn more about this vulnerability: https://docs.kics.io/latest/queries/common-queries/common/7f370dd5-eea3-4e5f-8354-3cb2506f9f13

        [1]: ../../path/simple.tf:2

                001: provider "aws" {
                002:   access_key = <SECRET-MASKED-ON-PURPOSE>
                003:   secret_key = <SECRET-MASKED-ON-PURPOSE>


EC2 Instance Has Public IP, Severity: HIGH, Results: 1
Description: EC2 Instance should not have a public IP address.
Platform: Terraform
Learn more about this vulnerability: https://docs.kics.io/latest/queries/terraform-queries/aws/5a2486aa-facf-477d-a5c1-b010789459ce

        [1]: ../../path/simple.tf:7

                006:
                007: resource "aws_instance" "example" {
                008:   ami           = "ami-0c94855ba95c71c99"
```

```
Passwords And Secrets - Generic Access Key, Severity: HIGH, Results: 1
Description: Query to find passwords and secrets in infrastructure code.
Platform: Common
Learn more about this vulnerability: https://docs.kics.io/latest/queries/common-queries/common/7f370dd5-eea3-4e5f-8354-3cb2506f9f1

        [1]: ../../path/simple.tf:2

                001: provider "aws" {
                002:   access_key = <SECRET-MASKED-ON-PURPOSE>
                003:   secret_key = <SECRET-MASKED-ON-PURPOSE>


EC2 Instance Has Public IP, Severity: HIGH, Results: 1
Description: EC2 Instance should not have a public IP address.
Platform: Terraform
Learn more about this vulnerability: https://docs.kics.io/latest/queries/terraform-queries/aws/5a2486aa-facf-477d-a5c1-b010789459c

        [1]: ../../path/simple.tf:7

                006:
                007: resource "aws_instance" "example" {
                008:   ami           = "ami-0c94855ba95c71c99"



Results Summary:
HIGH: 3
MEDIUM: 1
LOW: 1
INFO: 3
TOTAL: 8

Results saved to file /root/results/results.json
Generating Reports: DoneScan duration: 1m3.480325841s
ubuntu $
```

https://www.linkedin.com/in/manitechy/

3.Results summary.

Results Summary:
HIGH: 3
MEDIUM: 1
LOW: 1
INFO: 3
TOTAL: 8

We need check above terraform file misconfiguration and vulnerabilities details cloud engineer/ client and fix the issue and re-run the scan and we can see the results on json format

How check the results on json format:-

ubuntu $ find / -name "results.json"

/var/lib/docker/overlay2/131af55f316c3e8ce2f8b3fae76a152f3867425a1e763a955c6b7740b121918d/diff/root/results/results.json

ubuntu $ cd /var/lib/docker/overlay2/131af55f316c3e8ce2f8b3fae76a152f3867425a1e763a955c6b7740b121918d/diff/root/results/

```
ubuntu $ find -name "results.json"
ubuntu $ find / -name "results.json"
/var/lib/docker/overlay2/131af55f316c3e8ce2f8b3fae76a152f3867425a1e763a955c6b7740b121918d/diff/root/results/results.json
ubuntu $ cd /var/lib/docker/overlay2/131af55f316c3e8ce2f8b3fae76a152f3867425a1e763a955c6b7740b121918d/diff/root/results
ubuntu $ ls
results.json
ubuntu $ nano results.json
ubuntu $ nano results.json
ubuntu $
```

```
Editor   Tab 1   +
  GNU nano 4.8
{
        "kics_version": "v1.7.3",
        "files_scanned": 1,
        "lines_scanned": 18,
        "files_parsed": 1,
        "lines_parsed": 18,
        "lines_ignored": 0,
        "files_failed_to_scan": 0,
        "queries_total": 1046,
        "queries_failed_to_execute": 0,
        "queries_failed_to_compute_similarity_id": 0,
        "scan_id": "console",
        "severity_counters": {
                "HIGH": 2,
                "INFO": 7,
                "LOW": 1,
                "MEDIUM": 2,
                "TRACE": 0
        },
        "total_counter": 12,
        "total_bom_resources": 0,
        "start": "2023-07-13T12:44:54.6209377532",
        "end": "2023-07-13T12:46:14.9047741092",
        "paths": [
                "/path/ec2.tf"
        ],
        "queries": [
```

https://www.linkedin.com/in/manitechy/