Incident Repsonse Playbook Template

Incident Type

GuardDuty Finding: Execution:EC2/MaliciousFile

Introduction

This playbook is provided as a template to customers using AWS products and who are building their incident response capability. You should customize this template to suit your particular needs, risks, available tools and work processes.

Security and Compliance is a shared responsibility between you and AWS. AWS is responsible for “Security of the Cloud”, while you are responsible for “Security in the Cloud”. For more information on the shared responsibility model, please review our documentation (https://aws.amazon.com/compliance/shared-responsibility-model/).

You are responsible for making your own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) references current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. This document is provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Summary

This Playbook

This playbook outlines response steps for incidents involving deployment of a privileged container. These steps are based on the NIST Computer Security Incident Handling Guide (https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf) (Special Publication 800-61 Revision 2) that can be used to:

- Gather evidence
- Contain and then eradicate the incident
- Recover from the incident
- Conduct post-incident activities, including post-mortem and feedback processes

Interested readers may also refer to the AWS Security Incident Response Guide (https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/welcome.html) which contains additional resources.

Once you have customized this playbook to meet your needs, it is important that you test the playbook (e.g., Game Days) and any automation (functional tests), update as necessary to achieve the desired results, and then publish to your knowledge management system and train all responders.

Note that some of the incident response steps noted in each scenario may incur costs in your AWS account(s) for services used in either preparing for, or responding to incidents. Customizing these scenarios and testing them will help you to determine if additional costs will be incurred. You can use AWS Cost Explorer (https://aws.amazon.com/aws-cost-management/aws-cost-explorer/) and look at costs incurred over a particular time frame (such as when running Game Days) to establish what the possible impact might be.

Throughout this playbook you will see references to AWS API calls that can assist for gathering information or making modifications to resources. These commands can be executed in multiple ways including from CloudShell, within the AWS Management Console, or from your favorite IDE or CLI utility.

In reviewing this playbook, you will find steps that involve processes that you may not have in place today. Proactively preparing for incidents means you need the right resource configurations, tools and services in place that allow you to respond to an incident. The next section will provide a summary of this incident type, and then cover the five steps (parts 1 - 5) for handling privileged containers.

This Incident Type

An Amazon GuardDuty finding represents a potential security issue detected within your network. GuardDuty generates a finding whenever it detects unexpected and potentially malicious activity in your AWS environment. All GuardDuty finding references in this playbook will be related to the GuardDuty finding JSON that can be seen in the GuardDuty console, downloaded from the GuardDuty or Security Hub console, or exported to S3.

Execution:EC2/MaliciousFile (https://docs.aws.amazon.com/guardduty/latest/ug/findings-malware-protection.html#execution-malware-ec2-maliciousfile)

A malicious file has been detected on an EC2 instance

This finding indicates that the GuardDuty Malware Protection scan has detected one or more malicious files on the listed EC2 instance within your AWS environment. This listed instance might be compromised.

Details on what resources are involved with this activity can be found in the finding details (https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings-summary.html#findings-resource-affected).

Incident Response Process

Part 1: Acquire, Preserve, Document Evidence

For any Incident

1. You become aware of potential indicators of compromise (IoCs). These could come in various forms, but the original source is a GuardDuty finding:

    - An internal ticketing system (the sources of the ticket are varied and could include any of the means below)
    - From an alert in one of your monitoring systems either inside or external to AWS (that are ingesting GuardDuty Findings, in AWS, this could include AWS Security Hub)
    - Alarms or observations that resources have been created or deleted in your account that cannot be accounted for in your CMDB, exist in regions that you do not operate infrastructure in, or themselves have generated alerts Amazon Detective (https://aws.amazon.com/detective/getting-started/) is a useful tool for understanding these relationships)

2. Confirm a ticket/case has been raised for the incident using your designated ticketing system. If not, manually raise one.

3. Determine and begin to document any end-user impact/experience of the issue. Findings should be documented in the ticket/case related to the incident

4. Open an incident war room using your standard communication tools, and page in the relevant stakeholders

5. In the case of automatically created tickets/cases, verify the finding in GuardDuty (what caused the ticket to be created?)

For This Incident Type

1. Identify the specific EC2 instance impacted documenting the Instance ID, Account Id, and Region associated with the Instance. In GuardDuty this will be in the Overview section of the finding details.
2. Identify the volume Amazon Resource Name (ARN) where the malware is located. In GuardDuty this will be in the Threats Detected section and labeled Volume ARN.
3. Identify and document the location and name of the malware identified by GuardDuty. In GuardDuty this will be in the Threats Detected section and labeled File path and File name.
4. Identify what GuardDuty finding caused the malware scan to occur. In GuardDuty this will be in the Malware scan details labeled Trigger finding ID. This information will help determine what this malware was doing, for example command and control activity. To see a list of findings that initiate a malware protection scan please visit the GuardDuty Documentation (https://docs.aws.amazon.com/guardduty/latest/ug/gd-findings-initiate-malware-protection-scan.html)
5. List security group information that will be used to quarantine the instance in the containment section using the AWS CLI command below.
    ```
    aws ec2 describe-instances
    --instance-ids $Instance_ID
    --query "Reservations[0].Instances[0].NetworkInterfaces[].Groups"
    ```
6. Acquire images of the instance volumes using the AWS CLI command below.
    ```
    aws ec2 create-snapshot
    --volume $Data_Volume_ID
    --description "Data Volume Snapshot of Bad Web Server from Account $Account_ID in Region $Region"
    --tag-specifications 'ResourceType=snapshot,Tags=['"\({Region}'_BadWebServer_DataVolume_Snapshot_'\)(date
    -u +"%Y-%m-%d")"$(date -u +"%H:%M:%SZ")'"}]'
    ```

7. Enable termination protection for the instance using the AWS CLI command below. This step is needed if you want to preserve the instance for forensics.
   ```
   aws ec2 modify-instance-attribute
   --instance-id $Instance_ID
   --attribute disableApiTermination
   --value true
   ```
8. Disable the "DeleteOnTermination" setting for all volumes using the command below. This will ensure that if the instance is deleted the volumes are not. This step is needed if you want to preserve the instance for forensics.
   ```
   aws ec2 modify-instance-attribute
   --instance-id $Instance_ID
   --block-device-mappings "[{"DeviceName": "/dev/xvda","Ebs":{"DeleteOnTermination":false}}]"
   ```
9. Ensure instance shutdown behavior is set to "Stop" versus "Terminate" with the command below to ensure stopping or shutting down does not terminate the instance.
   ```
   aws ec2 modify-instance-attribute
   --instance-id $Instance_ID
   --attribute instanceInitiatedShutdownBehavior
   --value stop
   ```
10. Tag the instance for future identification during containment with the AWS CLI command below.
    ```
    aws ec2 create-tags
    --resources $Instance_ID
    --tags Key=Status,Value=Quarantine
    ```
11. Follow your defined playbook for performing host based forensics on EC2. If you do not have a defined playbook AWS has published different resources to help you get started. Forensic investigation environment strategies in the AWS Cloud (https://aws.amazon.com/blogs/security/forensic-investigation-environment-strategies-in-the-aws-cloud/), How to automate forensic disk collection in AWS (https://aws.amazon.com/blogs/security/how-to-automate-forensic-disk-collection-in-aws/), Automated Forensics Orchestrator for Amazon EC2 (https://aws.amazon.com/solutions/implementations/automated-forensics-orchestrator-for-amazon-ec2/), and a hands on workshop EC2 Incident Response and Forensic Analysis Workshop (https://catalog.us-east-1.prod.workshops.aws/workshops/e524ee28-a1ac-4cc6-b599-d5ced0fc5de1/en-US)

Part 2: Contain the Incident

If you determine that the activity is unauthorized, or decide it is prudent to assume so, the first priority is to prevent further compromise without impact to production workloads.

1. Determine if disabling the EC2 instance in question will not result in a service outage.
2. If the EC2 instance is a part of an Auto Scaling group it will need to be removed from the group. You can remove the instance from the auto-scaling group with the AWS CLI command below.
   ```
   aws autoscaling detach-instances
   --instance-ids $Instance_ID
   --auto-scaling-group-name
   ```
3. Next the instance will need to be deregistered if attached to an Elastic Load Balancer to prevent requests from being sent.
   ```
   aws elb deregister-instances-from-load-balancer
   --instances $Instance_ID
   --load-balancer-name
   ```
4. Remove the instance profile from the instance with the AWS CLI command below to remove IAM permissions.
   ```
   aws ec2 disassociate-iam-instance-profile --association-id $Association_ID
   ```

5. Prevent further connections to Isolate and quarantine the instance via security groups with the AWS CLI commands below in step 5. (Note that this isolation security group needs to be created beforehand. This can be a security group that doesn't allow any traffic, or traffic allowed specific to your use case) **Note** Be aware that there is currently no way to completely isolate/contain/quarantine an Instance using Security Groups, due to Connection Tracking. While Security Groups can be used to prevent future connections, they CANNOT be used to terminate existing Tracked Connections, as noted in the documentation Connection Tracking (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/security-group-connection-tracking.html). The following are the only methods to completely isolate/contain/quarantine an Instance:
    1. Shutting the instance down
        - You must ensure that you've collected all necessary volatile data needed for investigations like memory, disk, and/or system logs BEFORE shutting down!
    2. Using the endpoint's local/native utilities
        - Implementing OS/host firewall rules to terminate/block connections at the system level
    3. Using an Endpoint Agent
        - Leveraging an EDR or similar endpoint agent with containment/isolation capabilities
    4. NACL's
        - Note that using NACL's will affect an entire subnet and cannot be used to isolate a singular Instance
    5. Disconnecting the ENI
        - This is only possible for non-Primary ENI's as you cannot disconnect a primary ENI aws ec2 modify-instance-attribute
--instance-id $Instance_ID
--groups $Isolation_SG_ID

You can also automate the incident response and forensics capabilities utilizing AWS services. Examples on how to automate this activity can be found at this AWS documentation (https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/automate-incident-response-and-forensics.html)

Part 3: Eradicate the Incident

This is the stage for taking remedial action to minimize the impact of the unauthorized activities.

1. To remediate the compromised EC2 instance, follow the steps below
    1. Investigate the potentially compromised instance for malware and remove any discovered malware. You may check the AWS Marketplace (https://aws.amazon.com/marketplace) to see if there are helpful partner products to identify and remove malware.
    2. If you are unable to stop unauthorized activity on your EC2 instance, we recommend that you terminate the compromised EC2 instance and replace it with a new instance as needed or only leave it running for additional data acquisition. Use the AWS CLI Command below to shut the system down. aws ec2 stop-instances --instance-ids $Instance_ID
    3. If it is determined in your forensic analysis (part 5) that your AMI is potentially compromised you will need to follow your playbook for creating or recreating AMIs to ensure full eradication of any potential compromise.

Part 4: Recover from the Incident

1. For resources that were modified during the compromise:
    1. If the resource can be removed and replaced, do so. For example, EC2 instances in an EC2 Auto Scaling group, with a Launch Configuration referencing a non-compromised AMI, or Launch Templates used to create EC2 instances, terminate the instance, allow EC2 Auto Scaling to adjust capacity as required
    2. If the resource cannot be replaced, either:
        1. Restore the resource from a known good backup, or;
        2. Prepare a new resource and configure it into the application's infrastructure, while isolating the compromised resource and removing it from the application's infrastructure. Update the CMDB accordingly
        3. Either remove the compromised resource, or continue to leave it isolated for post-incident forensics
2. For resources that were deleted during the compromise:
    1. Determine what (if any) application the resource belonged to, by checking in the CMDB, or confirming the resource's tag(s) (Check AWS Config if the tags aren't listed in the CloudTrail entry and the resource is supported by AWS Config (https://docs.aws.amazon.com/config/latest/developerguide/resource-config-reference.html))
    2. If the deleted resource can be restored from backup, commence the restore procedure
    3. If you are using immutable architectures or similar deployment methods, redeploy the relevant resources
    4. If neither of the above options are possible, consult the CMDB to obtain the resource's configuration, and recreate the resource and configure it into the application's infrastructure.
    5. If the CMDB information is inadequate or non-existent, assign resource creation to the relevant personnel in your organization
3. For resources that were created during the compromise. For more details on what to do if you notice unauthorized activity refer to this knowledge center article (https://aws.amazon.com/premiumsupport/knowledge-center/potential-account-compromise/)
    1. Confirm these have been deleted or isolated for further analysis, per the steps in **Part 3**.
4. Validate that the affected resources were remediated and the expected state of any impacted systems were restored

Part 5: Post-Incident Activity

This activity contains two parts. Firstly, some compromised resources may require forensic analysis, either to fulfil regulatory obligations or improve incident handling, both taking input from the root cause analysis that will result from forensic investigation. The second part is a “sharpen the saw” activity which helps teams to assess their response to the actual incident, determine what worked and what didn't, update the process based on that information and record these findings.

Firstly, perform any required forensic investigation to determine (for compromised resources) what methods the actors may have used and to determine if additional risks and risk mitigations are required for the resources and/or applications in question.

1. For any compromised resources that have been isolated for further analysis, perform the forensic activity on those resources and incorporate the findings into the post-incident report. For an example, please refer to the Automated Forensics Orchestrator for Amazon EC2 (https://aws.amazon.com/solutions/implementations/automated-forensics-orchestrator-for-amazon-ec2/)
2. Ensure that the CMDB is correctly updated to reflect the current status of all resources and applications impacted

Secondly, review the incident itself and the response to it, to determine if anything needs to be changed for handling any similar incidents in the future.

1. Review the incident handling and the incident handling process with key stakeholders identified in Part 1, Step 8.
2. Document lessons learned, including attack vector(s) mitigation(s), misconfiguration, etc.
3. Store the artifacts from this process with the application information in the CMDB entry for the application and also in the CMDB entry for the credential compromise response process.

Additional Resources

Incident Response

https://aws.amazon.com/blogs/security/how-get-started-security-response-automation-aws/ (https://aws.amazon.com/blogs/security/how-get-started-security-response-automation-aws/)
https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/welcome.html (https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/welcome.html)
https://aws.amazon.com/blogs/security/how-to-automate-incident-response-in-aws-cloud-for-ec2-instances/ (https://aws.amazon.com/blogs/security/how-to-automate-incident-response-in-aws-cloud-for-ec2-instances/)
https://aws.amazon.com/blogs/security/forensic-investigation-environment-strategies-in-the-aws-cloud/ (https://aws.amazon.com/blogs/security/forensic-investigation-environment-strategies-in-the-aws-cloud/)
https://aws.amazon.com/blogs/security/how-to-automate-forensic-disk-collection-in-aws/ (https://aws.amazon.com/blogs/security/how-to-automate-forensic-disk-collection-in-aws/)

GuardDuty

https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings-summary.html (https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings-summary.html)

Preperation

Security and Networking sections in Best practices for Amazon EC2 (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-best-practices.html). Amazon EC2 security groups for Linux instances (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html) and Amazon EC2 security groups for Windows instances (https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-security-groups.html). Tips for securing your EC2 instances (Linux) (https://aws.amazon.com/articles/tips-for-securing-your-ec2-

instance/). AWS security best practices (https://aws.amazon.com/architecture/security-identity-compliance/) Infrastructure Domain Incidents on AWS (https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/infrastructure-domain-incidents.html) Browse for further assistance on the AWS developer forums: *https://forums.aws.amazon.com/index.jspa (https://forums.aws.amazon.com/index.jspa)* If you are a Premium Support package subscriber, you can submit a technical support (https://support.console.aws.amazon.com/support/home?#/case/create?issueType=technical) request.