Continuous Security: Monitoring & Active Defense in the Cloud

Get application and software security training at SANS Institute!

DEV540: Secure DevOps and Cloud Application Security

NORTHERN VIRGINIA 2018
McLean, VA | Mar 19-23
sans.org/u/ytK

SANS 2018
Orlando, FL | Apr 3-7
sans.org/u/zld

RSA 2018
San Francisco, CA | Apr 12-16
sans.org/u/Ac7
Eric Johnson

- Application Security Curriculum Manager, SANS Institute
  - SANS Certified Instructor
  - Contributing author of DEV540, DEV531, and DEV544

- Senior Security Consultant, Cypress Data Defense
  - Security assessments – source code reviews, web app pen tests, mobile app pen tests
  - Coder – security tools, demos, not enough to be called a developer anymore

- Education & Training
  - Iowa State M.S. Information Assurance, B.S. Computer Engineering
  - CISSP, GSSP-Java, GSSP-.NET, GWAPT

- Contact information
  - Email: ejohnson@sans.org
  - Twitter: @emjohn20
Agenda

• Continuous Security: Monitoring & Active Defense in the Cloud

Monitoring & Active Defense in the Cloud

1. Insufficient Logging & Monitoring
2. Monitoring & Active Defense
3. Purple Team Target Environment
4. Postmortem Review
#1 Insufficient Logging & Monitoring
Insufficient auditing and logging

- Auditable events such as logins, access control violations, high value transactions, etc. are not logged
- Application logs are not monitored for suspicious activity
- Thresholds and alert notifications are not in place
- System is unable to detect and in some cases respond to active attacks in real time
CASE STUDY: S3 SECURITY MISCONFIGURATIONS

Several known S3 misconfiguration vulnerabilities left files open to the public:

- Booz Allen Hamilton / National Geospatial-Intelligence Agency (NGA) mistakenly reveals geospatial intelligence data, battlefield intelligence, drone surveillance imagery
- Deep Root Analytics stores 198 million voter names, home addresses, date of birth, phone numbers and voter registration
- Israeli software company Nice Systems stores 14 million Verizon subscriber records in an unprotected Amazon S3 bucket
This would never happen in real life, would it?

- CVE 2017-5638 Struts 2 remote code execution vulnerability disclosed
- Attackers discover a vulnerable server on the Internet several months later
- Compromise the server and issue millions of HTTP requests to a backend server for customer data
- Extract 140MM customer records from the server to an attacker controlled machine
WHAT DO THESE BREACHES HAVE IN COMMON?

• Effective monitoring & logging facilities could have limited the blast radius
• Security audit log events do not exist or security teams do not know where the audit logs are stored
• No one is monitoring audit logs
• No automated vulnerability detection
#2 Monitoring & Active Defense
WHAT IS ACTIVE DEFENSE?

The process of analysts monitoring for, responding to, learning from, and applying their knowledge to threats internal to the network.

Robert M. Lee
The Sliding Scale of Cyber Security
SANS Reading Room
Aug 2015
Log data is stored in a number of different locations that are not always easy to find:

- Virtual Private Cloud (VPC) Flow Logs
- CloudFront Distribution Logs
- Application Load Balancer (ALB) Access Logs
- Application Server Logs
### VPC FLOW LOGS

- Generated by VPC/NAACL/ENI
- Stored in Cloud Watch Logs
- Connection 5-tuple, packet count, data size, ACCEPT/REJECT

<table>
<thead>
<tr>
<th>#</th>
<th>Initiator</th>
<th>Initiator MAC</th>
<th>Initiator IP</th>
<th>Target</th>
<th>Target MAC</th>
<th>Target IP</th>
<th>Source MAC</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Time In</th>
<th>Time Out</th>
<th>Decision</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>eni-1418a4ac</td>
<td>117.80.152.174</td>
<td>10.36.20.165</td>
<td>29459</td>
<td>23</td>
<td>6</td>
<td>1</td>
<td>40</td>
<td>1512097597</td>
<td>1512097656</td>
<td>REJECT</td>
<td>OK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>eni-1418a4ac</td>
<td>181.214.87.6</td>
<td>10.36.20.165</td>
<td>57226</td>
<td>3392</td>
<td>6</td>
<td>1</td>
<td>40</td>
<td>1512097668</td>
<td>1512097716</td>
<td>REJECT</td>
<td>OK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>eni-1418a4ac</td>
<td>10.36.20.165</td>
<td>10.36.20.88</td>
<td>51532</td>
<td>6</td>
<td>5</td>
<td>4295</td>
<td>1512321447</td>
<td>1512321503</td>
<td>ACCEPT</td>
<td>OK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>eni-1418a4ac</td>
<td>10.36.20.165</td>
<td>10.36.115.77</td>
<td>43704</td>
<td>443</td>
<td>6</td>
<td>9</td>
<td>1077</td>
<td>1512321447</td>
<td>1512321503</td>
<td>ACCEPT</td>
<td>OK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CLOUDFRONT DISTRIBUTION LOGS

- Generated by CloudFront distributions
- Stored in S3, as compressed text
- Endpoint, verb, response code, user agent, TLS protocol

1 2018-02-15 04:51:37 ORD54 1868 67.3.94.83 POST d3u41jzgezzd9m.cloudfront.net /rest/user/login 500 https://d3u41jzgezzd9m.cloudfront.net/ Mozilla/5.0%2520(Macintosh;%2520Intel%2520Mac%2520OS%2520X%252010_13_3)%2520AppleWebKit/537.36%2520(KHTML,%2520like%2520Gecko)%2520Chrome/64.0.3282.167%2520Safari/537.36 -- Error VEiwPfQ0mRZkJqeqqQ50pNEjshtx02h3y4IvdZzNzzF4gkGBDRWJPgQ== d3u41jzgezzd9m.cloudfront.net https 607 0.044 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Error HTTP/1.1 --
ALB ACCESS LOGS

- Generated by Application Load Balancer (ALB)
- Stored in S3, as compressed text
- URL, User Agent, 4-tuple, sizes & times, TLS cipher info, target group, TLS certificate

```
2017-12-03T17:34:07.737496Z app/DELTA-rALBW-1IGP33T9UXF17/8a93492e9b69fc1a
18.217.211.55:42362 10.36.10.251:80 0.001 0.013 0.000 200 200 49 1562 "GET
https://delta-ralbw-1igp33t17-639.us-east-1.elb.amazonaws.com:443/wordpress/wp-
includes/js/wp-embed.min.js?ver=4.9.1 HTTP/2.0" "Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.12; rv:57.0) Gecko/20100101 Firefox/57.0" ECDHE-RSA-AES128-GCM-
SHA256 TLSv1.2 arn:aws:elasticloadbalancing:us-east-
1:537026129675:targetgroup/DELTA-rALBW-1CBYUDS2LIYVV/92bbb020fa8a5b84 "Root=1-
5a24358f-546f637f2f0788226db15a5d" "delta-ralbw-1igp33t9uxf17-639737194.us-east-
1.elb.amazonaws.com" "arn:aws:iam::537026129675:server-certificate/non-
production-testing-server-cert"
```
**APPLICATION SERVER LOGS**

- **EPHEMERAL** log files generated by apache
- Stored in `/var/log/httpd` on the instance
- ELB IP, timestamp, request, response code & size, user agent

| 1 | 10.36.115.77 - - [03/Dec/2017:17:34:08 +0000] "GET /wordpress/?p=1 HTTP/1.1" 200 60759 "https://delta-ralbw-ligp33t9uxf17-639737194.us-east-1.elb.amazonaws.com/wordpress/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:57.0) Gecko/20100101 Firefox/57.0"
| 2 | 10.36.115.77 - - [03/Dec/2017:17:34:09 +0000] "GET /wordpress/wp-content/themes/twentyseventeen/style.css?ver=4.9.1 HTTP/1.1" 200 83036 "https://delta-ralbw-ligp33t9uxf17-639737194.us-east-1.elb.amazonaws.com/wordpress/?p=1" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:57.0) Gecko/20100101 Firefox/57.0"
The Monitoring Sucks team provides a central collection of git repositories covering:

- Tool repositories
- Metrics catalog
- Relevant blog posts

Monitoring Sucks git repo:
- https://github.com/monitoringsucks
The Monitoring Sucks Metrics Catalog created by Jason Dixon (@obfuscurity) guides your metrics for:

- Network infrastructure
- Web servers
- Application servers
- Databases
- Message queues
- And more...

**http (80/tcp):**
- duration_connect_msec (u_int32)
- duration_firstbyte_msec (u_int32)
- duration_total_msec (u_int32)
- request_bytes (u_int32)
- response_bytes (u_int32)
- response_code (string)
The Monitoring Sucks Metrics Catalog guides our interpretation of the data:

• Spikes of 404 errors may indicate a scanner fingerprinting the site, looking for backdoors, testing for vulnerable software

• Spikes of 500 errors may indicate injection attacks (e.g., SQL injection) attacks can trigger these

• Tracking user agents from known attack tools – sqlmap, nikto, w3af, nmap, etc.
• Data overload
  – Requests generate log entries in up to 5 locations!
  – Volume of logs to search for those behaviors, multiple locations (e.g., S3, Cloud Watch Logs, etc)

• Humans are not great at processing text logs
  – No time for human analysis
  – No time to acquire, analyze, decide, and implement a change

• How do we turn meaningful metrics into visual dashboards?
• What can we actively automate to defend ourselves?
ACTIVE SECURITY MONITORING

Monitoring attack driven production data in CloudWatch:
• Actions we can take, *internal* to our network, to defend our app
  – Block the IP address(es) generating large numbers of the items above
  – Redirect traffic from suspect IPs to alternate location

• **AWS Web Application Firewall (WAF)**
  – Integrates with ALB and CloudFront
  – Protections for SQL Injection, Cross Site Scripting
  – Ability to add customized rules
  – Managed rules available via marketplace partners (re:Invent 2017)
  – API for automation
  – IP lists – block, allow
AWS Labs CloudFormation template:
  • Blocks SQL injection
  • Blocks Cross site scripting
  • Manually managed IP lists (block, allow)
  • IP reputation list integration
    - Spam Haus, Emerging Threats, Tor Exit Nodes
    - Hourly updates
  • HTTP flood protection
  • Honeypot URL
  • Log parsing that updates a dynamic block list
  • https://github.com/awslabs/aws-waf-security-automations
• Incoming requests pass through an ALB or CloudFront distribution
• XSS / SQLi signatures are blocked
• Lambda function parses access logs and blacklists suspicious IPs
• Known bad IP lists are automatically parsed and update WAF blacklists
ACTIVE DEFENSE: HONEYPOT ENDPOINTS

• Attacker / scanner spiders the web site
• Web site hides the honeypot endpoint (URL) in a HTML comment, hidden link, or custom header
• Visiting the link invokes an API Gateway endpoint
• API Gateway invokes a Lambda function that automatically blacklists the IP address
• Hidden links or custom headers invoke honey pot endpoint
• API gateway endpoint invokes the lambda bad bot function
• Bad bot IP address is added to the blacklist
SERVERLESS TO THE RESCUE!

- **Warning**: code kung fu required!
- DevSecOps opportunity to build relationships with your development teams!
- Languages Supported
  - Python (2.7 or 3.6), Node.js (4.3, 6.10), C#, Java 8, and Go
- AWS Serverless Application Repository
  - In “preview” - announced at re:Invent 2017
def lambda_handler(event, context):
    response = { 'statusCode': 200, 'headers': {'Content-Type': 'application/json'}, 'body': '' }
    print '"[lambda_handler] Start"'
    try:
        global waf
        if environ['LOG_TYPE'] == 'alb':
            session = boto3.session.Session(region_name=environ['REGION'])
            waf = session.client('waf-regional')
        else:
            waf = boto3.client('waf')

        source_ip = event['headers']['X-Forwarded-For'].encode('utf8').split(',')[0].strip()
        waf_update_ip_set(environ['IP_SET_ID_BAD_BOT'], source_ip)
        message = {}
        message['message'] = "[%s] Thanks for the visit."%source_ip
        response['body'] = json.dumps(message)
    except Exception as e:
        print e
    print '"[lambda_handler] End"'
    return response
#3
Purple Team
Target Environment
ACTIVE DEFENSE PURPLE TEAM EXERCISE

- Red team attacks
- Blue team monitors the logs, dashboards, and actively defends
- The environment:
  - PCI DSS Compliance infrastructure stack
  - WordPress stack
  - JuiceShop instance
- https://github.com/ejohn20/sans-cloud-monitoring
• Management VPC contains a bastion jump box and private admin resources

• Production VPC with CloudFront / ALB to backend web resources, database

• Logging to S3, CloudTrail, and CloudWatch alarms
• CloudFront distribution protected by the AWS WAF forwards traffic
• DMZ ALB forwards traffic to the WordPress instances
• WordPress instances connect to backend RDS Aurora MySQL databases
• OWASP Juice Shop (https://www.owasp.org/index.php/OWASP_Juice_Shop_Project)
• Cloudfront distribution protected by the AWS WAF forwards traffic
• EC2 instance origin running NodeJS app in a docker container
LET THE GAMES BEGIN!

The @SecDSM folks are pretty good at this:

- 1st: 2017 DEFCON Capture the Packet
- 1st: 2017 THOTCON CTF
- 1st: 2017 CircleCityCon CTF
- 1st: 2017 WWHackinFest CTF
- 1st: 2017 GRRCON CTF
#4 Postmortem Review
WordPress web site identified the resources in scope. The rules are as follows:

- 1 hour attack window
- Rules of engagement
  - No DoS attacks
  - No protocol flooding
  - No resource flooding
• Infrastructure dashboard shows a spike in reject packets from 27 / minute to 77,300 per minute
• Request count average climbs from 422 / minute to 6,850 / minute
• ELB 500 errors Blocked rise sharply from 2 / minute to 107 / minute
• Request count average climbs from 50 / minute to 3,500 / minute
• Error rate climbs sharply for 400 and 500 response codes
• Blocked WAF requests rises sharply from 5 / minute to 921 / minute
BLUE TEAM WIN: WAF AUTOMATION

- WAF bad bot lambda functions blacklists the IP address within the first 2 minutes
- Triggers included nmap scans, fuzzer invoking honey pot endpoint, command execution payloads

```
ERROR

The request could not be satisfied.

Request blocked.

Generated by cloudfront (CloudFront)
Request ID: ZXW1vNLHsabIjV89J2GdIoCHcEsD9EeaQt9tqEITj009ttUBysEDQ==
```
• Configuring CloudWatch to monitor thresholds and send SNS notifications:
  – Email
  – SMS
  – Slack
  – HipChat

• The figure to the right shows my inbox from various events during the purple team setup and event
• WordPress scans (WPScan) were not blocked by the WAF

• Vulnerable WordPress plugin allowed command injection instance

```
[root@pohlim1 ~]# nc -l 2222
bash: no job control in this shell
bash-4.2$ pwd
/var/www/wordpress/wordpress
bash-4.2$
```
WordPress secrets management fail in the wp-config.php file:

```php
/** ABSPOH */
/* */
* @link https://codex.wordpress.org/Editing_wp-config.php *
* */
* Package WordPress */

// ** MySQL settings /**
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wpadmin' );

/** MySQL database password */
define( 'DB_PASSWORD', 'F4K0MD0E3JUijhJzdhljBhBhDheDe' );

/** MySQL hostname */
define( 'DB_HOST', 'wordpress-rds-1b9h4r5C5yGd-databasescluster-ju2v32nndox6.cluster-c6yfrjdiubzw.us-east-1.rds.amazonaws.com' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/** Authentication Unique Keys and Salts. */
```
RED TEAM WINS: DATABASE ELEVATED PRIVILEGES

- WordPress database connection with elevated privileges is able to drop the entire database:

```
Database changed
mysql> show tables:
+--------------------+
| Tables_in_wordpress |
+--------------------+
| wp_commentmeta     |
| wp_comments        |
| wp_links           |
| wp_options         |
| wp_postmeta        |
| wp_posts           |
| wp_term_relationships |
| wp_term_taxonomy   |
| wp_terms           |
| wp_usermeta        |
| wp_users           |
+--------------------+
12 rows in set (0.00 sec)
mysql> drop database wordpress
drop database wordpress
+ -> ;
Query OK, 12 rows affected (0.15 sec)
mysql> system()
```
WordPress home page re-written to include:

• Command shell
• Meet @tompohl
Wordpress is a special beast. Harden plugins / secrets management

Database dashboards showing connection and query activity would alert in the event of a compromise

AWS PCI quick start template:
- By default, no VPC flow logs in the management stack
- SSH logging on the bastion server is not enabled by default

JuiceShop application running in the docker container are difficult to extract
- Docker run with the "awslogs" switch did not deliver to CloudWatch
Audit logging and security monitoring provides defense in depth protection to limit the blast radius of attacks in progress

Controls such as least privilege, code reviews are critical pieces to the entire puzzle

In a real environment, active defense techniques such as honeypot endpoints can be very useful for the blue team

Simulated events help establish a culture of learning and continuous improvement
Thank you for attending!
ejohnson@sans.org / @emjohn20

Get application and software security training at SANS Institute!

DEV540: Secure DevOps and Cloud Application Security

NORTHERN VIRGINIA 2018
McLean, VA | Mar 19-23
sans.org/u/ytK

SANS 2018
Orlando, FL | Apr 3-7
sans.org/u/zld

RSA 2018
San Francisco, CA | Apr 12-16
sans.org/u/Ac7