I GOT 99 PROBLEMS, BUT A WAF AIN'T ONE

BY YOUR BOY, RUN D.M.Z.

Micah K Brown

- Twitter: @MicahKBrown
- Munich Re: IT Security Engineer II
- GitHub: https://github.com/micahkbrown
  - DLP Demystified (2018 talk)
  - Star Wars: How an ineffective Data Governance Program Destroyed the Galactic Empire (2019 talk)
  - How to cook a Five Star Meal from the Convenience of Your Hotel Room (Derby Con 2019)
  - Doing simple at scale (2020 talk)
- Vice President of Greater Cincinnati ISSA Chapter
- CISSP

- Served 45 pounds of free Pulled Pork to @DerbyCon 2019!
- Real Corp 2018 goal: "Learn to Cook Brisket Like a Texan."
- Real Corp 2019 goal: "Continue to Cook Brisket Like a Texan."
- On most Fridays, find me smoking both an old fashioned and pizza!

*Thoughts and view are my own and do not reflect that of my employer.

Greater Cincinnati IT Security groups

ISSA

http://www.cincy-issa.org/

SMBA

https://sites.google.com/view/cincysmba

CiNPA
Cincinnati Networking Professionals Association
Security Special Interest Group
SECURITY SIG

https://www.meetup.com/TechLife-Cincinnati/events/

(ISC)²®

https://www.infoseccincy.org/

Micah K Brown Hydration Challenge™

Fill a pint glass with **water** and take a drink when:

- I say ‘A.I.’ or ‘M.L.’
- I stress ‘the importance of testing’
- See ‘this is the worst project I ever worked on’ (case insensitive)
- See a diagram on screen
- Micah alludes to **Queen City Con2021** (including this shameless plug)

“This, is the WORST project I ever worked on”
— Micah K Brown

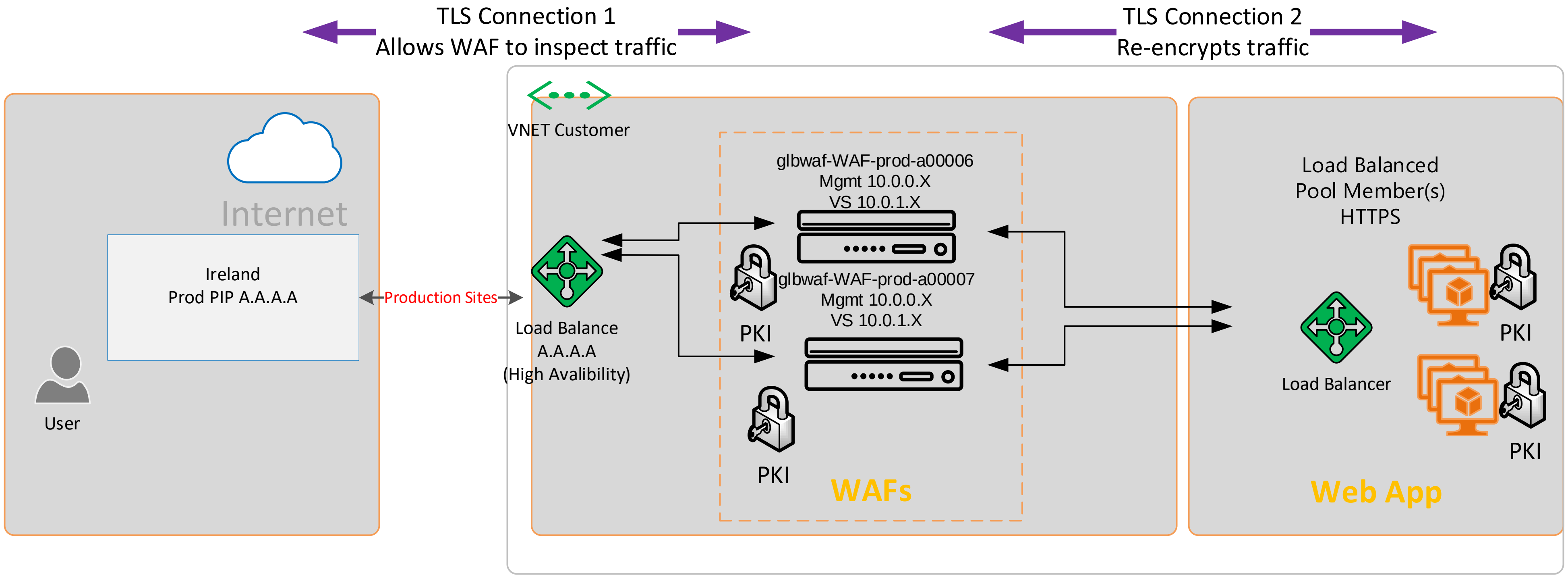
WHAT IS A WAF:

WHAT IS A WAF

A web application firewall (WAF) is a specific form of application firewall that filters, monitors, and blocks HTTP traffic to and from a web service. By inspecting HTTP traffic, it can prevent attacks exploiting a web application's known vulnerabilities, such as SQL injection, cross-site scripting (XSS), file inclusion, and improper system configuration.

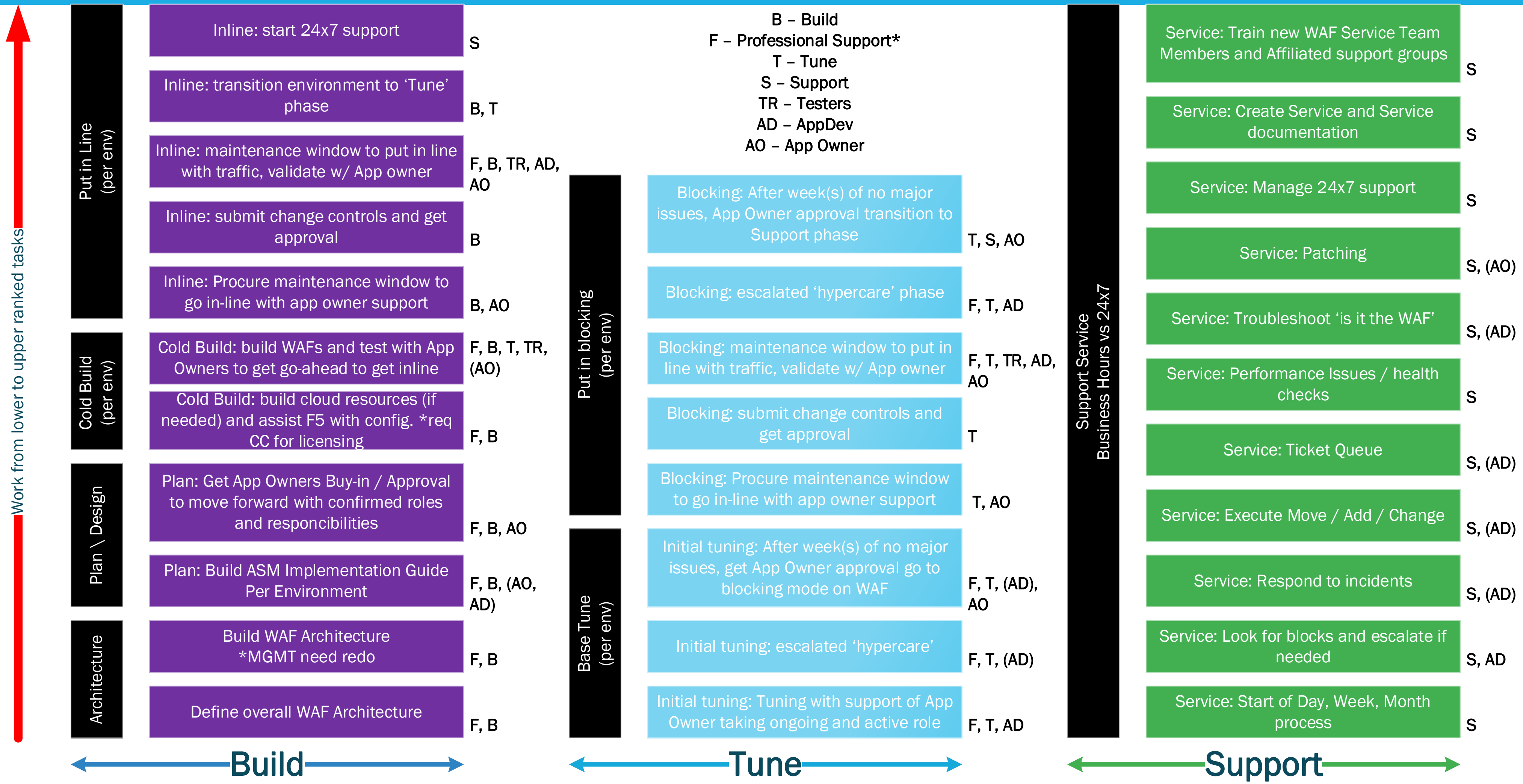
https://en.wikipedia.org/wiki/Web_application_firewall

WHAT IS A WAF



- Permit
- Permit with alert
- Modify traffic
- Block no alert
- Block with alert

BUILD – TUNE – SUPPORT MODEL



Work from lower to upper ranked tasks

B – Build
F – Professional Support*
T – Tune
S – Support
TR – Testers
AD – AppDev
AO – App Owner

Build

Put in Line (per env)	Inline: start 24x7 support	S
	Inline: transition environment to 'Tune' phase	B, T
	Inline: maintenance window to put in line with traffic, validate w/ App owner	F, B, TR, AD, AO
	Inline: submit change controls and get approval	B
	Inline: Procure maintenance window to go in-line with app owner support	B, AO
Cold Build (per env)	Cold Build: build WAFs and test with App Owners to get go-ahead to get inline	F, B, T, TR, (AO)
	Cold Build: build cloud resources (if needed) and assist F5 with config. *req CC for licensing	F, B
Plan \ Design	Plan: Get App Owners Buy-in / Approval to move forward with confirmed roles and responcibilities	F, B, AO
	Plan: Build ASM Implementation Guide Per Environment	F, B, (AO, AD)
Architecture	Build WAF Architecture *MGMT need redo	F, B
	Define overall WAF Architecture	F, B

Tune

Put in blocking (per env)	Blocking: After week(s) of no major issues, App Owner approval transition to Support phase	T, S, AO
	Blocking: escalated 'hypercare' phase	F, T, AD
	Blocking: maintenance window to put in line with traffic, validate w/ App owner	F, T, TR, AD, AO
	Blocking: submit change controls and get approval	T
	Blocking: Procure maintenance window to go in-line with app owner support	T, AO
Base Tune (per env)	Initial tuning: After week(s) of no major issues, get App Owner approval go to blocking mode on WAF	F, T, (AD), AO
	Initial tuning: escalated 'hypercare'	F, T, (AD)
	Initial tuning: Tuning with support of App Owner taking ongoing and active role	F, T, AD

Support

Support Service Business Hours vs 24x7	Service: Train new WAF Service Team Members and Affiliated support groups	S
	Service: Create Service and Service documentation	S
	Service: Manage 24x7 support	S
	Service: Patching	S, (AO)
	Service: Troubleshoot 'is it the WAF'	S, (AD)
	Service: Performance Issues / health checks	S
	Service: Ticket Queue	S, (AD)
	Service: Execute Move / Add / Change	S, (AD)
	Service: Respond to incidents	S, (AD)
	Service: Look for blocks and escalate if needed	S, AD
	Service: Start of Day, Week, Month process	S

BIGGEST CHALLENGES

Like most projects, success and failure depends on the team.

- High Total Cost of Ownership and Management Support
- Overcoming Fear, Uncertainty, Doubt!
- Clearly defining post SDLC life under a WAF.
- Conflicts with work already in ‘App Dev Pipeline’.
- Winning trust and support from App Dev.
- Not all environments are equal (feature by feature).
- How can you intercept Web Traffic (**DNS** vs passthrough)
  - If use DNS, your TTL will control how fast you can move WAF out of line with traffic.

“THIS, is the worst project I ever worked on!”
— Micah K Brown

WAF TEAM, ASSEMBLE

YOU ARE GOING TO NEED A BIGGER BOAT

TEAM WAF

Build

- Strong Skills: networking, IT Architecture, Cloud Architecture, strongly understand organizations policies

Tune

- Strong Skills: Application / Web Development Security, common AppDev vulnerabilities.

Support

- Strong Skills: Networking, Customer specific WAF support, Application / Web Development Security, company policies such as hardening guides / change control / and ticketing systems, Building WAF Support Service, Training WAF support team

TEAM APPLICATION

Application Owner

- Generally this will be a manager or architect of an application. This person that is responsible for uptime and availability of an application. Technical knowledge is not required, but they need to pull the correct technical people in.

Application Developer

- A member of the Application Development team should be assigned to the project team. (Critical)

Testers

- A member of the Application Development team should be assigned to the project team. (Critical)

“This, IS the worst project I ever worked on!”
— Micah K Brown

BUILD PHASE:

BUILD PHASE: ARCHITECTURE

- Define overall WAF Architecture
- Build out and WAF support systems
  - Centralized Management
  - Logging
  - Licensing

BUILD PHASE: PLAN / DESIGN

- Work with App Owner \ App Dev to understand Application and how to integrate a WAF.
- Formalize environment learning into a documented plan that defines:
  - Entire implementation and support strategy.
  - Rolls and Responsibilities.
  - Processes, Procedures, SLAs.
  - Diagrams are awesome for showing this
- Gain agreement with App Owner and App Dev before building.

BUILD PHASE: COLD BUILD

- Work with IT to stand up initial build of WAF.
- Apply initial Security Policy and any customizations needed for the App.
- At the end of this phase Build resource can turn the WAFs over to Tune, AppDev, and Testers to test application via host file redirection ahead of planning inline. Any learning suggestions / alerts should be reviewed for tune / block decision.
  - Volume and Diversity of traffic is critical for testing / validation.

BUILD PHASE: PUT IN LINE

- Get App Owner approval that ready to go inline in learning / transparent mode based based on host file testing.
- Get date, needed resources, and submit needed change control.
- I found implementation via Bridge call was best.
- Execute maintenance window to go in-line and validate functionality:
  - Transition WAF over to Tune Phase.
  - WAF now starts 24x7 / after ours support.

“This, is the worst project, I eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeever worked on!”
— Micah K Brown

TUNE PHASE:

TUNE PHASE: INITIAL TUNING

- When we first go in-line the WAF will be in transparent / learning mode.
- Tune will meet regularly with App Dev to review every learning suggestion / alert or tune / block decisions.
    - Both volume and diversity of traffic are critical.
    - Make sure that niche functionality is also tested. (Cross function testers)
    - Have Testers execute tests often.
- At some point you will see significant reduction to new learning suggestions / alerts. At this time start working with AppDev and App Owner on declaring success and start planning to put WAF into Blocking Mode.

TUNE PHASE: INITIAL TUNING

- Get App Owner approval that ready to convert to Blocking Mode.
- Get date, needed resources, and submit needed change control.
- I found implementation via Bridge call was best.
- Execute maintenance window to go in-line and validate functionality:
  - At this point the WAF should be kept in the Tune phase for a period of time to ensure no issues significant arise before handing over to the WAF Support Service.

“This, is the worst project, I ever WORKED on!”
— Micah K Brown

SUPPORT PHASE:

SUPPORT PHASE: BUSINESS AS NORMAL

- Day in / day out support.
- Start of day, week, month procedures.
- Regular review for learning suggestions and alerts. Escalate to AppDev if needed.
- Move / Add / Change tickets.
- Troubleshooting / performance testing procedures
- Patching
- Documentation
- Training

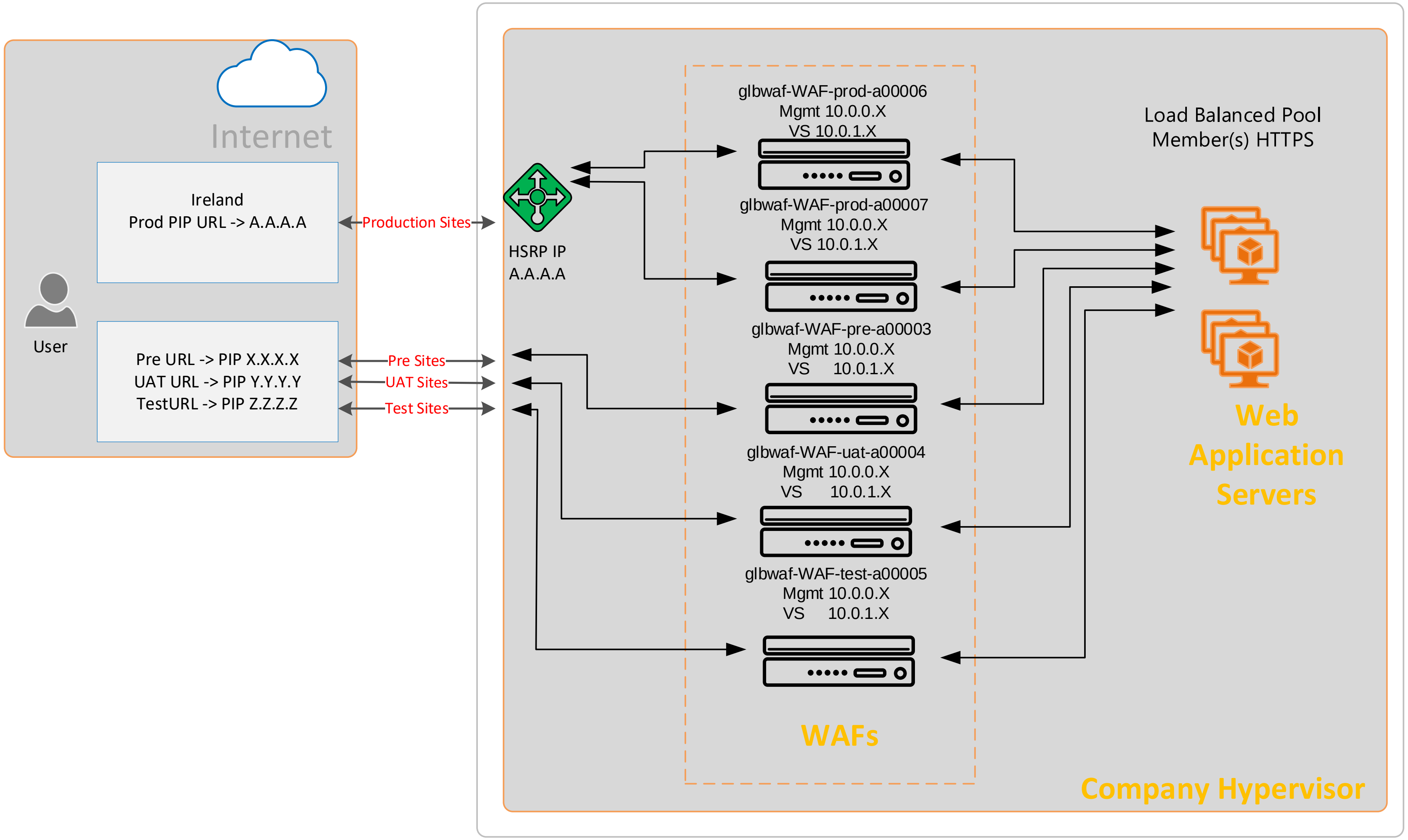
“This, is the worst project, I ever worked ON!”
— Micah K Brown

ARCHITECTURE:

ON PREM: CONSIDERATIONS

- Physical vs virtual machines.
- How to intercept traffic (DNS vs transparent capture)
- MGMT port should be on internal ‘protected’ VLAN.
- Front end web-application servers sould not be internet accessible once protected by a WAF.

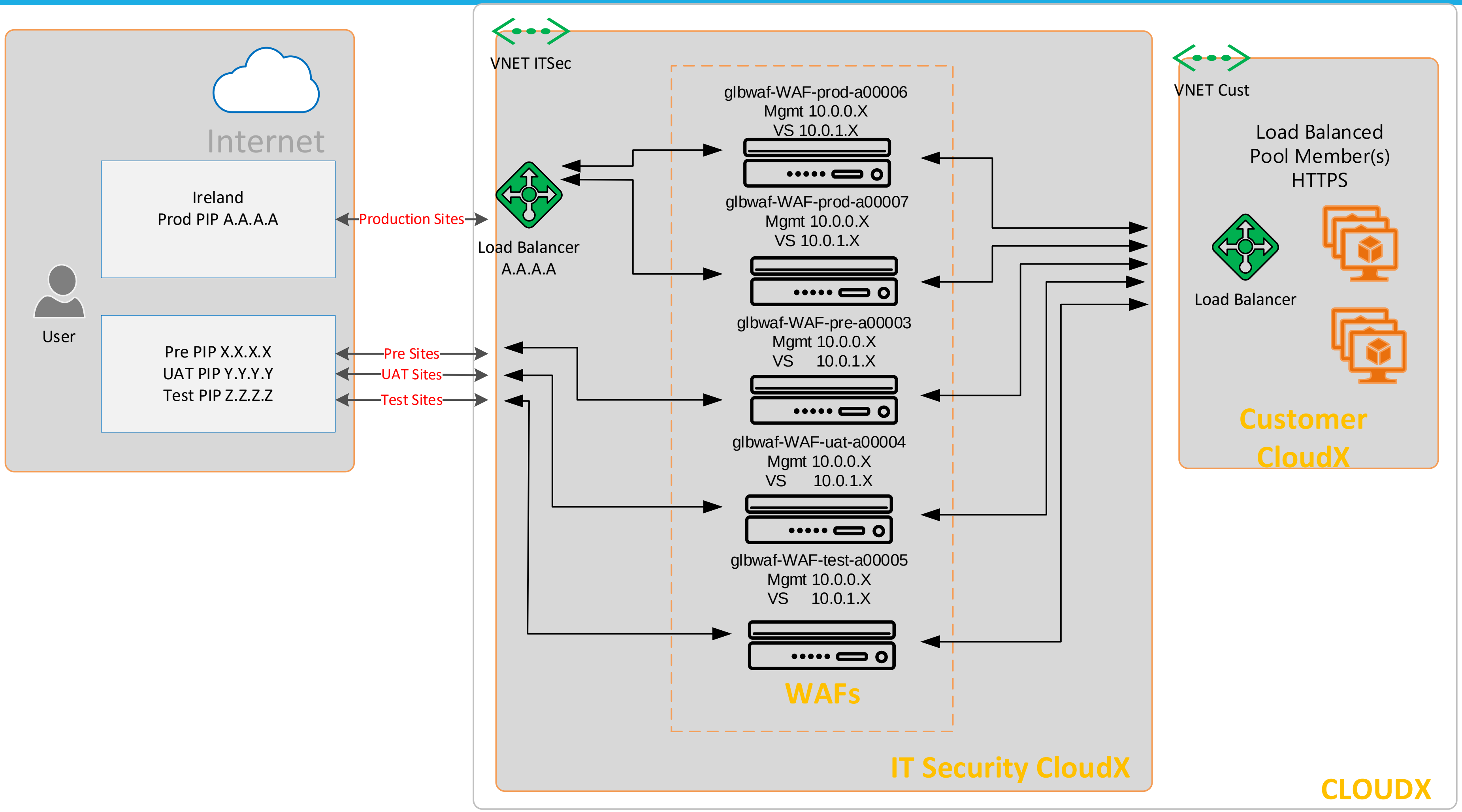
ON PREM


Internet
Ireland
Prod PIP URL -> A.A.A.A
User
Pre URL -> PIP X.X.X.X
UAT URL -> PIP Y.Y.Y.Y
TestURL -> PIP Z.Z.Z.Z
Production Sites
Pre Sites
UAT Sites
Test Sites
HSRP IP
A.A.A.A
glbwaf-WAF-prod-a00006
Mgmt 10.0.0.X
VS 10.0.1.X
glbwaf-WAF-prod-a00007
Mgmt 10.0.0.X
VS 10.0.1.X
glbwaf-WAF-pre-a00003
Mgmt 10.0.0.X
VS 10.0.1.X
glbwaf-WAF-uat-a00004
Mgmt 10.0.0.X
VS 10.0.1.X
glbwaf-WAF-test-a00005
Mgmt 10.0.0.X
VS 10.0.1.X
WAFs
Load Balanced Pool
Member(s) HTTPS
Web
Application
Servers
Company Hypervisor

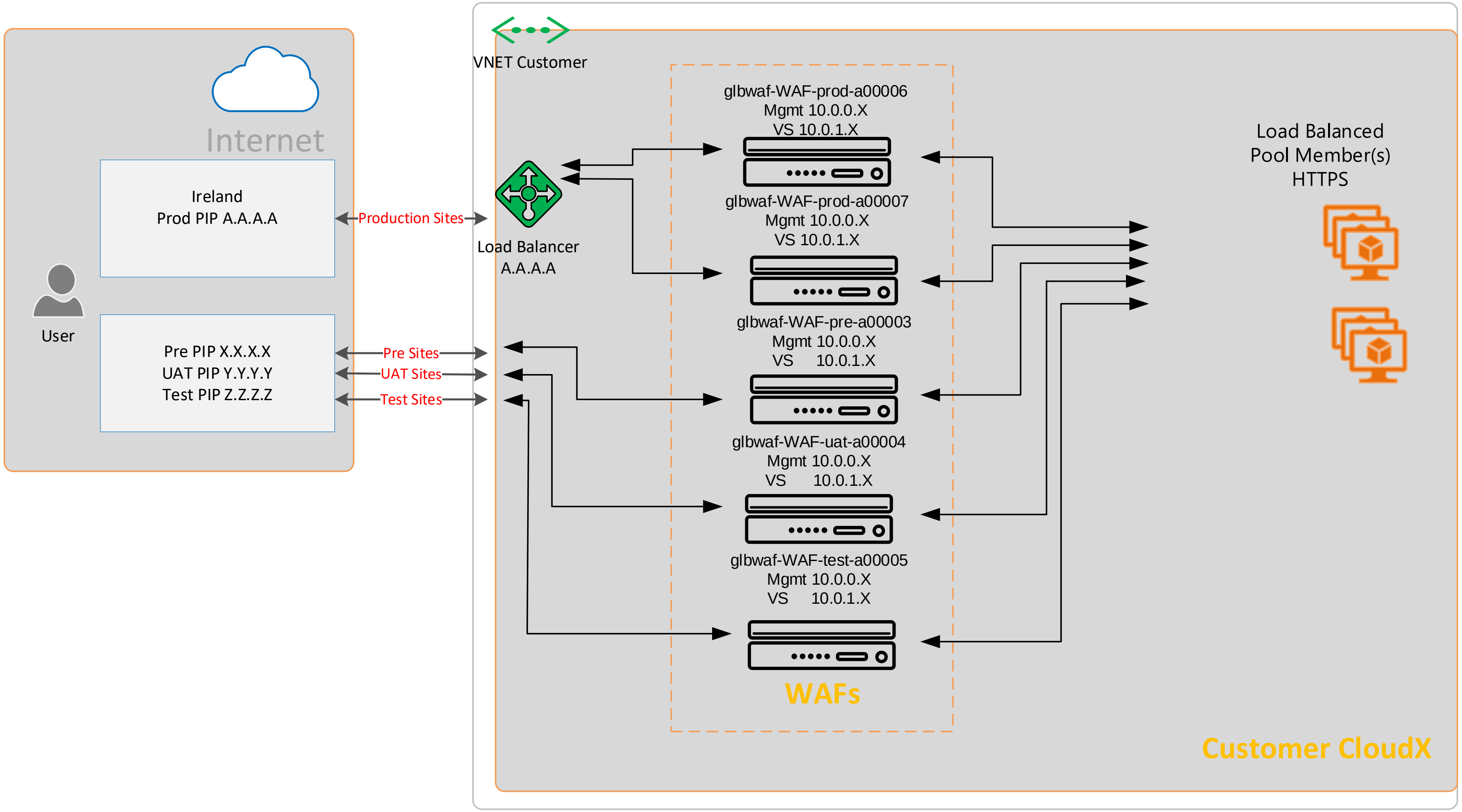
CLOUD: CONSIDERATIONS

- Implement is same cloud container or different?
  - Implement in same environment as servers is quickest but then AppDev might have more control over the WAF than you desire.
  - Implement in different cloud environments how do you pass traffic (over internet, vnet peering, VPN)
- How do you manage WAFs
  - Hardened Jump Server vs direct internet login

CLOUD: MULTIPLE CONTAINERS



CLOUD SINGLE CONTAINER



Internet

Ireland
Prod PIP A.A.A.A

User

Pre PIP X.X.X.X
UAT PIP Y.Y.Y.Y
Test PIP Z.Z.Z.Z

Production Sites

Pre Sites

UAT Sites

Test Sites

VNET Customer

Load Balancer
A.A.A.A

glbwaf-WAF-prod-a00006
Mgmt 10.0.0.X
VS 10.0.1.X

glbwaf-WAF-prod-a00007
Mgmt 10.0.0.X
VS 10.0.1.X

glbwaf-WAF-pre-a00003
Mgmt 10.0.0.X
VS 10.0.1.X

glbwaf-WAF-uat-a00004
Mgmt 10.0.0.X
VS 10.0.1.X

glbwaf-WAF-test-a00005
Mgmt 10.0.0.X
VS 10.0.1.X

WAFs

Load Balanced
Pool Member(s)
HTTPS

Customer CloudX

ONE POLICY TO RULE THEM ALL

Do you want to have a single policy to rule them all?

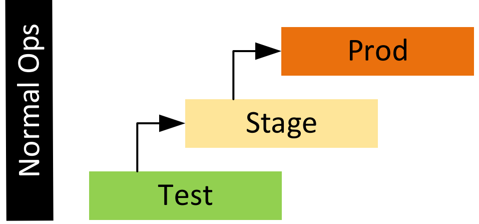
- Default answer from App Dev is that they want lower level environments to mirror Production as much as possible.
- Often, lower level environments are not identical. Thus, blindly copying learning suggestions / config / policy could have adverse effects.

Pay attention to cryptography (lower environments)

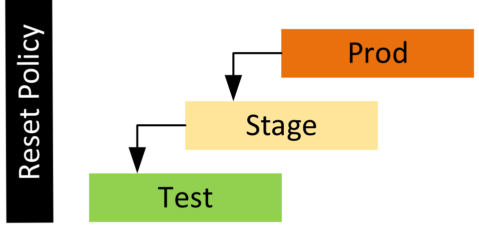
- Do not use vendor supplied / generic wildcard
- Ensure strength and cyphers
- Use trusted CAs.

LEARNING SUGGESTION COULD OCCUR AT ANY LEVEL

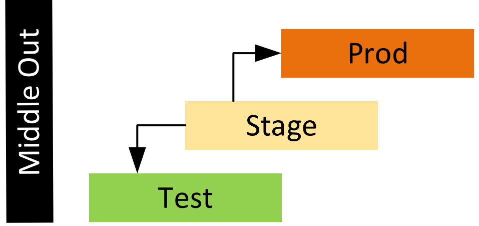
Normal Policy Flow



‘Reset’ Policy Flow



‘Middle Out’ Policy Flow



During the course of normal work it is expected that Test, Stage, and Production could have different revisions of ‘the policy’ based off the traffic observed and learning suggestions created by the AI / ML:

- Production would represent the ‘now’ state of production.
- Stage would represent the ‘near future’ state of production.
- Test would represent the ‘future future’ state of production.

We need to create process and procedures to allow (and encourage) learning suggestions flowing ‘bottom up’, ‘top down’, and ‘middle out’ as shown directly to the left of this sentence that you are reading currently. This brings up other questions including, but not limited to:

POLICY PUSH QUESTIONS

- What is the trigger to invoke policy push? (time vs event)
  Critical push should happen as soon as change control / maintenance window allows?
  Batch process for medium to low block?
  All blocks matter and should be treated with post haste?
- When demoting policy from higher environment to lower environment how to we ensure that learning suggestions unique to that environment are retained? (Prior to demote do we push from the lower environment to the higher environment, then push the newly updated higher environment to the lower environment?)
- Can you promote / demote single Learning Suggestion vs entire policy?
- I have heard talk about forking policies to be specific per web application. (So each WAF in we would have one parent policy and 5 / 6 child policies. How would this effect promote / demote activities? If we promote a policy from test to stage are we at risk for resetting the learning database on other policies?)
- How do we track this for our own sake and audit?

“This, is THE worst project, I ever worked on!”
— Micah K Brown

BONUS: MY SAMPLE GO INLINE CHANGE DOCUMENTATION

CHANGE TO GO INLINE 1:2

Plan to go inline SEARiousMeats
Date: TBD
Time: TBD
Meeting Invite: To be generated

Players:

- Build: Micah K Brown - Coach
- WAF Testers: Sally
- SEARiousMeats DNS: Tom
- SEARiousMeats Tester: Dick
- SEARiousMeats App Owner / Acceptance: Harry

Prep – To be completed ahead of time
0.1. SEARiousMeats: set DNS TTL to (must be done 48 hours in advance)

Reduce TTL down to 90 seconds:

FQDN (A or CName)	Original IP	TTL (sec)
https://SEARiousMeats.net	1.1.1.1	90

0.2. IT Sec SSL Labs

CHANGE TO GO INLINE 2:2

Implementation Window

1. Tom / SEARiousMeats: change DNS Records

FQDN (A or CName)	Original IP	New IP
https://SEARiousMeats.net	1.1.1.1	2.2.2.2

2. IT Sec confirm configuration and do basic validation / testing (10 min)

a. Certificate matching

i. https://SEARiousMeats.net a7713fbc abb7188f940d77f33c9fac819 ded6824

b. Test sites above from external wifi, VPN, cell phone

c. Test with SSL Labs

3. Modify FW / Cloud ACL to prevent WAF bypass

4. The Web-apps from step one are relased to App Owners / App testers. (2h)

5. With one hours left in maintenance window a go / no go decision is made.

   a. SEARiousMeats changes DNS back to original settings (10 min)

   b. IT Sec do basic validation / testing (10 min)

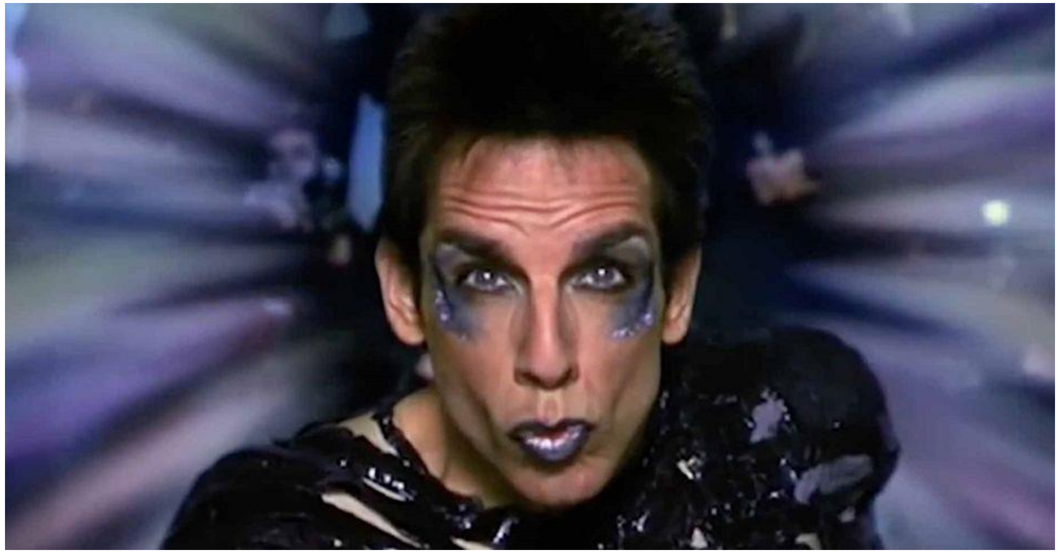
      i. local ARP may need to be flushed

   c. The Web-apps from step one are released to App Owners / App testers (1h)

Post: Tom / SEARiousMeats: Reset DNS TTL normal - TBD

FQDN (A or CName)	Original IP	TTL (sec)
https://SEARiousMeats.net	2.2.2.2	1800

The BRO-jito



- ¼ cup of mint leaves (more for garnish)
- 2 tablespoons of sugar
- 3 limes
- Blackberrys
- 3 cups of Apple Cider
- Ice (as desired)
- 2 cups of Sparkling Blackberry Juice

1. In medium pitcher, add mint leaves and sugar. Muddle porously.
2. Squeeze juices of limes into pitcher. Combine with mint and sugar. Add cider to pitcher and combine.
3. Stir in the ice and float the sparkling blackberry juice.
4. Pour over fresh ice and garnish with mint, blackberry, and lime wedge.