Preview of Vulture’s upcoming web filtering engine

Pass the SALT 2018.
Security And Libre Talks.
2-4 juillet 2018 - Lille, France.
Vulture?

• A brief history
  • 2003: Linux software (httpd / mod_perl + PHP Web UI)
  • 2016: FreeBSD Cluster (pf, haproxy, httpd + Django Web UI)

• Web SSO: mod_vulture + django portal

• Web application firewall
  • Clustered mod_security, using hiredis
  • mod_defender, aka "Naxsi for Apache2"  [whitelisting]
  • mod_svm
  • mod_svm
Vulture’s current filtering engine

Client

FreeBSD pf

Apache httpd

IP Reputation
GeoIP
mod_vulture
mod_defender
mod_security
mod_svm

Immediate block
Immediate block
Authentication & SSO
Request scoring ++
Request scoring ++
Request scoring ++

Security level
Permissive
Paranoid
Current limits

• Works well
• No performance issue, but we can do much better
• 3 engines => quite complex
  • Code overlapping
  • Complex UI, httpd knowledge recommended
• Rule-based approach
• Human-based approach
  • Time consuming
  • Need tuning
  • Not mistake proof configuration
The need for a better, unified engine

• Focused on performance

• High availability required

• Precision and intelligence

• No bullshit

• Simplicity -> For users AND for filter devs

Internal name: « D.A.R.W.I.N. »
Overview

Architecture

Client

FreeBSD PF

HAProxy

PF Blacklist (with timeout)

D.A.R.W.I.N.

GeoIP

Reputation

Session

Injection

Decision

Vulture’s Portal

Backend

Advence

Security for the Digital Age
Overview: XSS Filter

Filter

HTTP POST BODY

UNIX Socket

XSS Core Filter

Thread 1
Thread 2
Thread 3
Thread 4
...
Thread N

Monitoring

XSS Score: 87%
Overview: Filter Workflow

D.A.R.W.I.N.
Overview: Filter Workflow

D.A.R.W.I.N.

Filter 1 (ex: reputation)

Filter 2 (ex: SQLi)

Filter N

Manager

Management Socket
Performance?

• HAProxy asynchronous events
• C/C++14
• UNIX socket
• Shared in-memory cache (REDIS)
  • Context-sharing between filters among the Vulture cluster
  • Used by Darwin’s Neural Networks to track events in time
• Supports GPU acceleration
  • TensorFlow as AI library
High Availability

{"type": "update_filters", "filters": ["Decision"]}
High Availability

{ "type": "update_filters", "filters": ["Decision"] }
High Availability

Manager

Filter N

Decision

Decision

{"type":"update_filters","filters":["Decision"]}

Management Socket
High Availability

Manager

Filter N

Decision

Management Socket

{"type":"update_filters","filters":["Decision"]}

{"status":"OK"}
Precision and Intelligence

• **Precision**
  - Multiple small filters
    - Very efficient for one unit task
    - Ability to chain filters (workflow)

• **Intelligence**
  - Decision filter based on Artificial Intelligence
    - Prediction based on filters’ results
    - Active learning capabilities: Interact with human to correct itself
  - Human focuses on high-level “decisions”
    - The AI manages the technical security rules
No Bullshit

• Heuristic / basic correlation in a black box is not AI

• Those methods are promising but...
  • We use some of them in v3 (SVM, regression...)
  • Few false-positives
  • Unfortunately, few false-negatives: rules still needed

• We work hard to take it to the next level!
  • “AI first”: by design, not an add-on component
  • Excellent results so far, beta-version coming this year
Simplicity

• Easy for users
  • Minimalist configuration
  • Autonomous system
  • Simple feedback (Normal or Malicious request)

• Easy for developers
  • Filters mostly independent
  • Simple SDK
  • On Github soon ;)
Portable

• Replace HAProxy with anything you want
  • Simply develop a connector

• Not only HTTP !

• Real world example (aDvens): DARWIN + Rsyslog
  • mmdarwin plugin
  • Real time log analysis
  • Real time log enrichment
  • On Github soon... ;)
Questions ?
Thank You!

hugo.soszynski@advens.fr
jeremie.jourdin@advens.fr

https://www.vultureproject.org
https://github.com/VultureProject/mod_defender
https://github.com/VultureProject/darwin (coming soon)