Practical Defense with Mod Security Web Application Firewall (WAF)

Marian Ventuneac
marian.ventuneac@gmail.com

October 25th, 2013

whoami

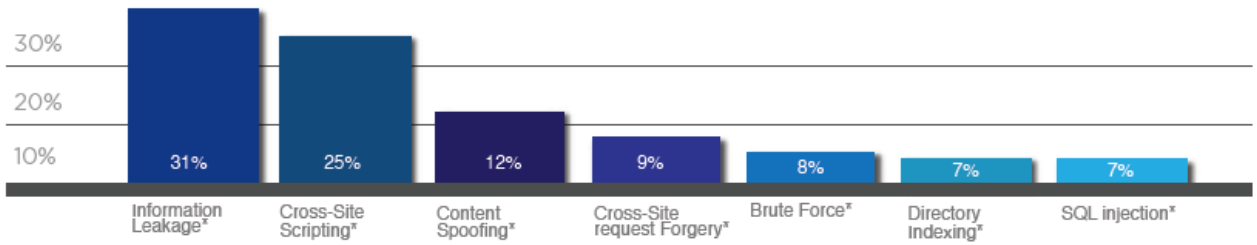
- Security Architect CISM, CISA with Genworth Financial
- Member of OWASP and ISACA global organizations
- OWASP Ireland Limerick Chapter Leader https://www.owasp.org/index.php/Ireland-Limerick
- OWASP Romania Board Member
- Security Researcher PhD, MEng, B.Sc
  - http://www.ventuneac.net
  - http://secureappdev.blogspot.com
  - http://dcsl.ul.ie

Presentation Outline

- ❑ Web Applications & Security Risks
- ❑ Brief Introduction to Web Application Firewalls (WAFs)
- ❑ Introducing mod_security WAF
- ❑ Demo

Why is Application Security Important?

- Most common vulnerabilities observed in 2013
  (Source: WhiteHat Website Security Statistics Report 2013)



- Web applications are one, if not the leading target of cyber-attacks
- For large organizations, more than 54% of breaches were linked to exploitation of application vulnerabilities

  (Source: Verizon 2012 Data Breach Investigation Report)

Web Application & Security Vulnerabilities

- Factors leading to vulnerable Web applications
  - Improper design
  - Insecure configuration/deployment
  - Lack of knowledge on secure coding
  - No secure code review
  - Lack of or improper security testing
  - Vulnerable 3rd party software/APIs/development frameworks
  - …

Classification of Application Security Risks

- OWASP Top 10 Application Security Risks 2013
  https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

- MITRE Common Weaknesses Enumeration (CWE)
  http://cwe.mitre.org/

- CWE/SANS Top 25 Software Errors
  http://www.sans.org/top25-software-errors/

- WASC Threat Classification
  http://projects.webappsec.org/w/page/13246978/Threat Classification

- …

OWASP Top 10 Application Security Risks 2013

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

- A1: Injection
- A2: Cross-Site Scripting (XSS)
- A3: Broken Authentication and Session Management
- A4: Insecure Direct Object References
- A5: Cross Site Request Forgery (CSRF)
- A6: Sensitive Data Exposure
- A7: Missing Function Level Access Control
- A8: Insecure Cryptographic Storage
- A9: Using Known Vulnerable Components
- A10: Unvalidated Redirects and Forwards

Web Application Firewalls (WAFs)

- Deployed to establish an external security layer that increases security, detects, and prevents attacks before they reach web applications

- What is it and what is it good for?
  - An intermediary device (appliance/server plugin/filter) that applies custom rules to incoming/outgoing traffic at application layer (OSI layer 7)
  - Inspect content of HTTP/SOAP/XML-RPC requests and responses

Web Application Firewalls (WAFs)

- What is it and what it is good for?
  - Could detect unusual traffic
  - Could use attack signatures to detect and stop dangerous traffic
- Architectural considerations
  - Typically installed in front of Web Servers
  - Detect & stop dangerous traffic before reaching the application
- Various commercial and open source solutions available

When Do We Need a WAF?

- ❑ Log traffic details (including POST requests)
- ❑ Add an extra layer of security to protect Web applications
- ❑ Rapid mitigation of known security risks affecting your Web applications

WhiteHat Website Security Statistics Report 2013 – Financial Services Industry Scorecard

- ❑ Regulatory compliance requirement
  - ❑ Payment Card Industry (PCI) Data Security Standard (DSS) - Install a web-application firewall in front of public-facing web applications

ModSecurity WAF Overview

- Open source WAF solution
- Currently developed by Trustwave SpiderLabs
- Available for Linux, Windows, Solaris, FreeBSD, OpenBSD, NetBSD, AIX, Mac OS X, and HP-UX
- Works with
  - Apache HTTP server
  - IIS Server
  - Nginx Server
  - for Java (now in beta testing, Google Summer of Code 2013)
    - uses JNI to hook into Java application servers

ModSecurity – Architectural Considerations

- Embeddable web application firewall
- Can be deployed as part of your existing web server infrastructure (Apache, IIS7 and Nginx).
  - No changes to existing network
  - No single point of failure
  - Implicit load balancing and scaling
  - Minimal overhead
  - No problem with encrypted or compressed content.

mod_security for Apache

- mod_security is an Apache module
  - runs inside Apache HTTP server
- Architectural considerations
  - Embed ModSecurity with individually deployed Apache HTTP servers
  - Protect Web Applications by using an Apache-based reverse proxy server with ModSecurity installed
- Attack detection and prevention rules
  - Trustwave's SpiderLabs provides free certified rule set for ModSecurity 2.x.
  - OWASP ModSecurity Core Rule Set (CRS)

mod_security - Attack Detection and Prevention

- Provides generic protection from unknown vulnerabilities
  - Negative security model
    - monitors requests for anomalies, unusual behaviour, and common web application attacks
    - log/reject invalid requests (e.g. with malformed HTTP headers, etc)
  - Known weaknesses and vulnerabilities
    - mitigate application vulnerabilities without modifying the code (code fixes need time)
  - Positive security model
    - only valid requests are accepted

mod_security - Attack Detection and Prevention (cont)

- Rules
  - Formed using regular expressions
  - Analyzes headers, cookies, environment variables, server variables, POST payload, script output, …
  - Custom rules supported
- Actions
  - Reject request with status code or with redirection
  - Execute internal binary
  - Log request
  - Rule chaining
  - …

mod_security - Attack Detection and Prevention (cont)

- OWASP CRS provides generic web applications protection
  - Common Web Attacks Protection (XSS, SQLi, etc)
  - Identification of Application Defects
  - HTTP Protection
  - Web-based Malware Detection (uses Google Safe Browsing API)
  - HTTP Denial of Service Protections
  - Integration with AV Scanning for File Uploads
  - Tracking Sensitive Data
  - …
- ModSecurity Virtual Patching
  - develop custom rules to prevent exploitation of known application vulnerabilities

mod_security - Attack Detection and Prevention (cont)

- Test, test and test again before deployment in production
  - Deploy in detection mode
  - Where valid traffic is blocked, tweak the rules
  - Once fine tuned, switch to protection mode
  - Logs monitoring is recommended
- Potential performance degradation
  - Switching on all protection rules will affect application performance
  - Identify what attacks you want to protect your application from, and enable only the required rules

mod_security - Attack Detection and Prevention (cont)

- When deployed with each instance of Apache HTTP server you administer, consider the effort required to maintain it
  - Potential solution: have ModSecurity installed, and switch it on only when required (to mitigate known risks)
- When deployed on a reverse proxy Apache HTTP server to protect multiple applications, make sure you don't introduce a bottleneck (single point of failure)
- When looking to protect business critical applications:
  - either become an expert yourself
  - or look for commercial support

Mod-security Demo

- ❑ ModSecurity install and configuration
  - ❑ Windows OS
  - ❑ Apache HTTP server configured in reverse proxy mode
  - ❑ ModSecurity with OWASP CRS rules
  - ❑ OWASP WebGoat Vulnerable Application

Mod-security Demo

- Detection of common Web application attacks
  - Log all suspicious traffic for analysis, don’t block it yet
- Generic protection against common Web application attacks (XSS, SQLi, etc)
- Create custom rules
  - Overriding default core rules to handle false positives
  - Virtual patching for a known application vulnerabilities

Additional Resources

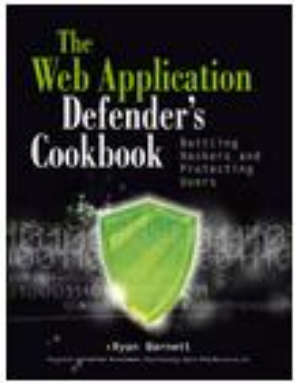
- ❑ ModSecurity home page

  http://www.modsecurity.org

- ❑ OWASP ModSecurity Core Rule Set (CRS)

  https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

- ❑ Books



Q&A

Thank You