$ whois WendelGH

- PT Consultant at Trustwave's SpiderLabs
- Over 7 years in the security industry
- Vulnerability discovery Webmails, AP, Citrix, etc
- Spoke in YSTS 2.0, Defcon 16, H2HC and others
- Affiliated to Hackaholic team
$ whois SandroGauci

- Founder and CSO EnableSecurity
- From .mt
- Security software
  - VOIPPACK (CANVAS addon)
  - Surfjack - insecure cookies
  - SIPVicious
- Security research papers
- Been around for > 9 years
Introduction

- WAF - Web Application Firewall
- next generation protection
- what can we do?
  - can be identified, detected
  - bypassing the rules
  - exploit WAFs
What is WAF?

- Attack signatures or abnormal behavior based
- WAFs products: software or hardware appliance.
- Flavors:
  - a reverse proxy
  - embedded
  - connected in a switch (SPAN or RAP)
- WAF products detect both inbound
- Some also detect outbound attacks
Who uses WAFs?

- Many banks around the world
- Companies which need high protection
- Many companies in compliance with PCI DSS (Payment Card Industry - Data Security Standard)
Operation Modes

- Negative model (blacklist based)
- Positive model (whitelist based)
- Mixed / Hybrid
The negative model

- Relies on a database of known attacks
- Eg. XSS strings like `<script>`, `</script>`, `String.fromCharCode`, etc.
- Often regular expressions
Whitelist model

- Whitelist based
- Learning mode to create a security policy of known “good” HTTP traffic
  - Known as dynamic profiling technology by some
- Example:
  Page news.jsp, the field "id" only accept numbers [0-9] and starting at 0 until 65535
  - news.jsp?id=-1 would not be allowed
Common Weaknesses

- Design issues
  - WAFs have to be similar to the web apps and http servers that they need to protect
  - Blacklists are by design “flawed”

- Bad implementation
  - Parsing issues

- Again - a WAF needs to do a lot of things that the web app and http server does
  - ergo they can have similar security flaws!
Detection

- A number of products can be detected
  - sometimes by design
- Detection is not a big deal but
  - ... sometimes we’re told that WAFs are ‘invisible’
  - the better you know your enemy (or client), the better
  - helps in a penetration test or targeted attack
  - shows that stealth attacks are possible
Detection

- Cookies
  - Reason: some WAFs are also load balancers

- Headers
  - Header rewriting
  - Most obvious would be "Server"
  - Sometimes is a feature called “server cloaking”
  - “Connection” header might be changed to Cneonction or nnCoection

- Response codes
  - 404 error codes for existent scripts
  - and 403 for non existent ones
Detection via response codes

- 404 error codes for existent scripts
- Different error codes (404, 400, 401, 403, 501, etc) for hostile parameters (even non existent ones) in valid pages.
starting up httpfox to monitor the responses
Automating WAF detection

- WAFW00F
  - Detect 20 different WAF products
    - the number keeps changing thanks to contributions :-)
  - Options to detect multiple WAFs in place
  - Generic detection methods included!

- Get your copy
  - waffit.googlecode.com
  - Please contribute
a short demo of wafw00f in action
Bypassing a WAF

- Fingerprint the rules
- Detect allowed / denied strings
- Combinations of allowed or denied strings
- Modify your attack to not match the blacklist
More on bypassing WAFs

- Encoding and language support, character sets
- Spaces, comments, case sensitive mutation, Unicode (%uc0af and %c0%af), etc
- The web server may parse, decode and interpret and HTTP request differently from the WAF
- HTML and JS is very flexible
- Various methods to split and encode your strings
Bypassing rules

■ “Our Favorite XSS Filters and how to Attack Them” by Eduardo Vela & David Lindsay
  ▪ Bypass the rules by splitting the attack (eval('al%'2b'lert(0)'))

■ “Shocking News in PHP Exploitation” by Stefan Esser
  ▪ Using “malformed” multipart/form-data to bypass most Modsecurity rules
  ▪ F5 BIG-IP ASM could be bypassed by sending it multipart/form-data that was interpreted differently by PHP than ASM
Vulnerable to XSS
I Am XSSable
The positive model

- It’s well known that the negative model is broken
- What about positive model?
- They are really secure?
- If we find a positive model should we give up?
Vulnerable to XSS

hi again
Testing WAFs for bypasses is a tedious job

- Which is why we automate it :-)

- WAFFUN - works in progress
  - Checks if the script echos back (esp in the case of xss)
  - Can check if error suppression is supported
  - Finds out how the WAF responds when a it reacts to an attack
  - Goes through a list of well known blacklisted strings
  - If any were blocked, it tries different encoding methods, null characters, unicode
Hello there, this is a vulnerable ASP file

test

this script is vuln to xss
WAFFUN: XSS constructor

- Tries a number of tags to find out which are allowed through
- Tries a number of DHTML event handlers
- Tries a number of Javascript methods
WAFs may be vulnerable too!

- Security software is not necessarily secure
- Web Application specific issues: XSS, SQLi
- Overflows
- DoS
Known issues

■ ModSecurity 2.5.9
  ▸ addresses 2 vulnerabilities
    ▪ "Fixed PDF XSS issue where a non-GET request for a PDF file would crash the Apache httpd process."
    ▪ "Fixed parsing multipart content with a missing part header name which would crash Apache."

■ Profense 2.6.3
  ▸ Profense Web Application Firewall Cross-Site Scripting and Cross-Site Request Forgery

■ DotDefender 3.8-5 (this week)
  ▸ Command Execution in dotDefender Site Management
    ▪ (requires authentication)
    ▪ seems like it is vulnerable to XSRF
Under Construction

The site you were trying to reach does not currently have a default page. It may be in the process of being upgraded and configured.

Please try this site again later. If you still experience the problem, try contacting the Web site administrator.
Thank you

- Do you have ideas / resources to improve our tools?
- wsguglielmetti [em] gmail [ponto] com
- sandro [em] enablesecurity [ponto] com
- Questions?