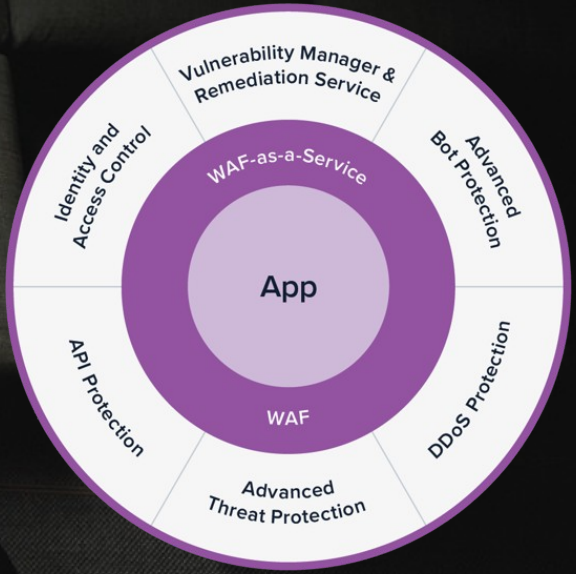
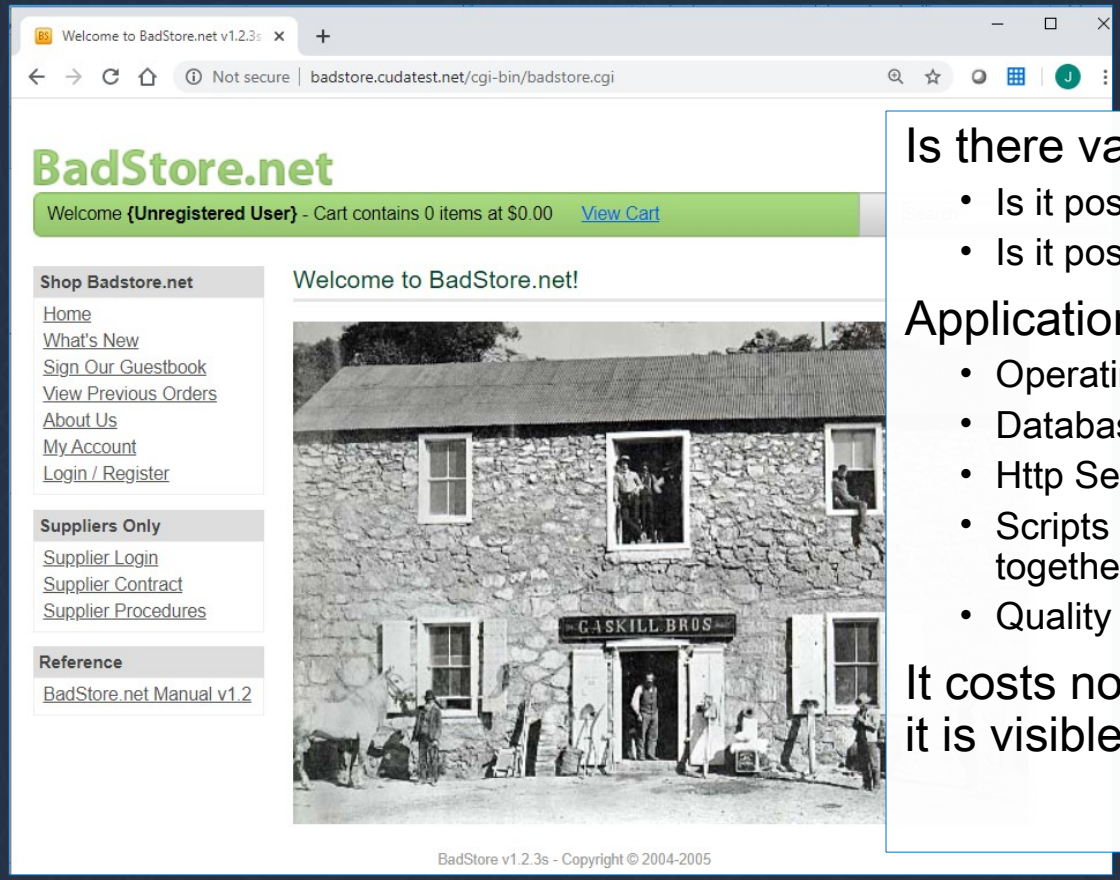
Secure Web Applications



Is every application worth protecting?



Is there valuable data?

- Is it possible to login?
- Is it possible to upload files?

Applications are multi-layered:

- Operating System (Linux?)
- Database (My SQL?)
- Http Server (Apache?)
- Scripts and software to make everything work together
- Quality control application (20%?)

It costs nothing to attack an application when it is visible in the Internet!

BadStore server for tests

BadStore.net

Welcome **{Unregistered User}** - Cart contains 0 items at $0.00 View Cart

Search Go

Shop Badstore.net

- Home
- What's New
- Sign Our Guestbook
- View Previous Orders
- About Us
- My Account
- Login / Register

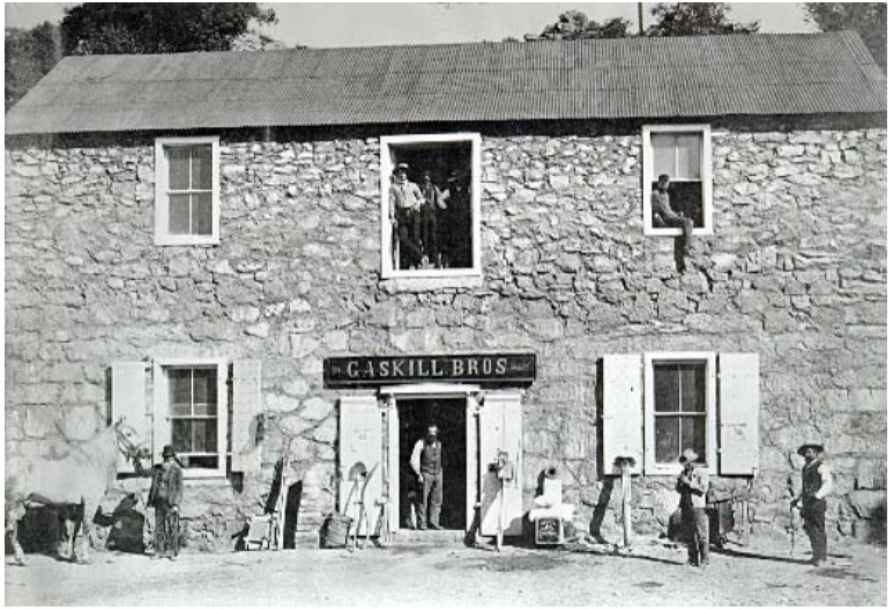
Suppliers Only

- Supplier Login
- Supplier Contract
- Supplier Procedures

Reference

- BadStore.net Manual v1.2

Welcome to BadStore.net!

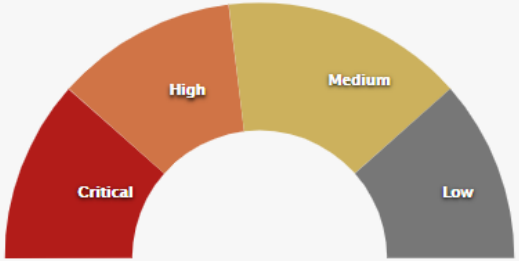


BadStore v1.2.3s - Copyright © 2004-2005

Vulnerability Scan of BadStore - Results

Results by severity level

Critical	9
High	9
Medium	12
Low	9



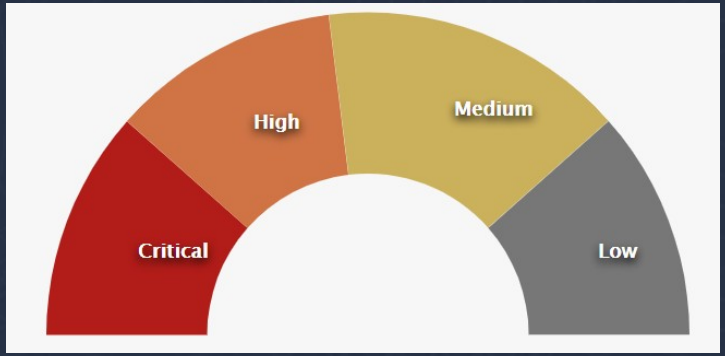
Scan Information

Start Time	April 27, 2021, 7:51 p.m.
End Time	April 27, 2021, 8:10 p.m.
Scan Time	0:19:32
Total Pages Crawled	29
Total Requests Performed	4314
Authentication	N/A
Authentication Username	N/A
Domain Verification	http://server1.cuda.link/ was verified by user1@cuda.link on April 20, 2021, 10:43 a.m.

Server Information

Server Responsive	Yes
Server Banner	Apache/2.4.25 (Debian)
Server OS	Linux
Server Technologies	Apache

Vulnerability Scan of BadStore - Results



Results by severity level

Critical	9
High	9
Medium	12
Low	9

Critical

1. Blind SQL Injection (1 instance)
2. SQL Injection (8 instances)

High

3. Reflected Cross-Site Scripting (6 instances)
4. Stored Cross-Site Scripting (3 instances)

Medium

5. Clickjacking: Missing X-Frame-Options Header (1 instance)
6. HTML Injection (5 instances)
7. HTTP OPTIONS Method Enabled (1 instance)
8. Malicious File Upload (1 instance)
9. Password is Sent Unencrypted (1 instance)
10. Sensitive File Found (2 instances)
11. Server-Side Source Code Found (1 instance)

Low

12. Autocomplete Enabled on Password Field (1 instance)
13. Email Address Found (1 instance)
14. HTML Form Without CSRF Protection (6 instances)
15. Open TCP/UDP Port Found (1 instance)

Website http://server1.cuda.link is based on BadStore.net project
It is a poorly written internet shop full of vulnerabilities

It is possible to login as admin without password (SQL Injection attack):

- Login/Register Username: `admin' OR '1'='1`

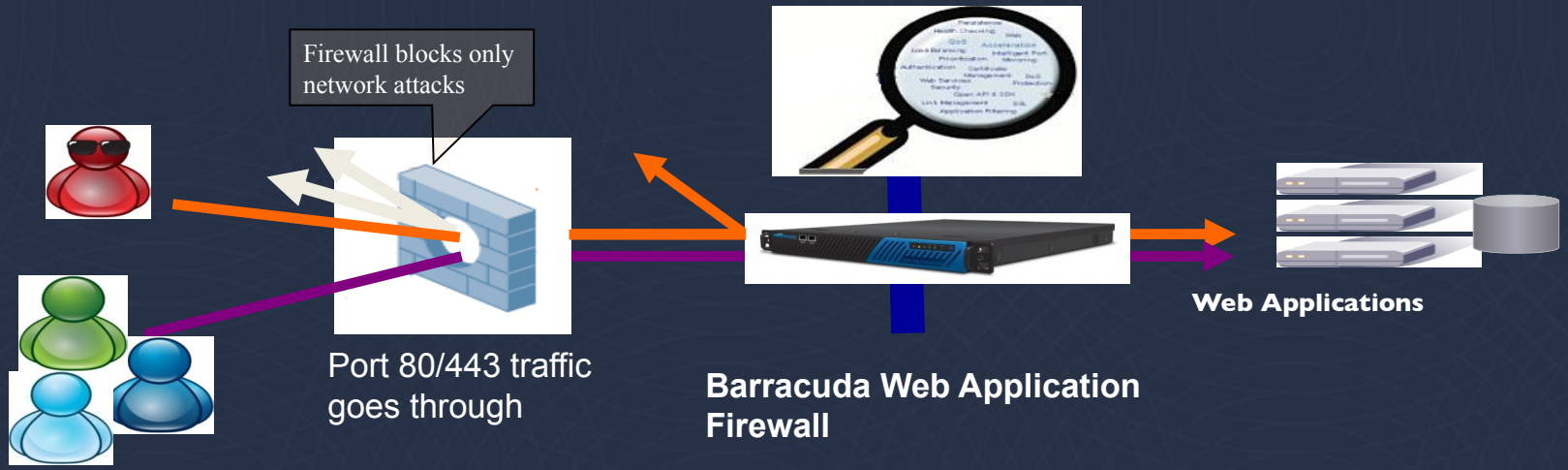
It is possible to list /etc/passwd file (SQL Injection attack with remote OS command execution):

- *Search* item field: `' UNION SELECT null,null,null,load_file('/etc/passwd')##`
- *Search* item field: `' UNION SELECT null,null,null,version()##`

It is possible to place a XSS trap in Guestbook:

- Insert a guestbook entry with `<script>alert('It is a trap!')</script>`

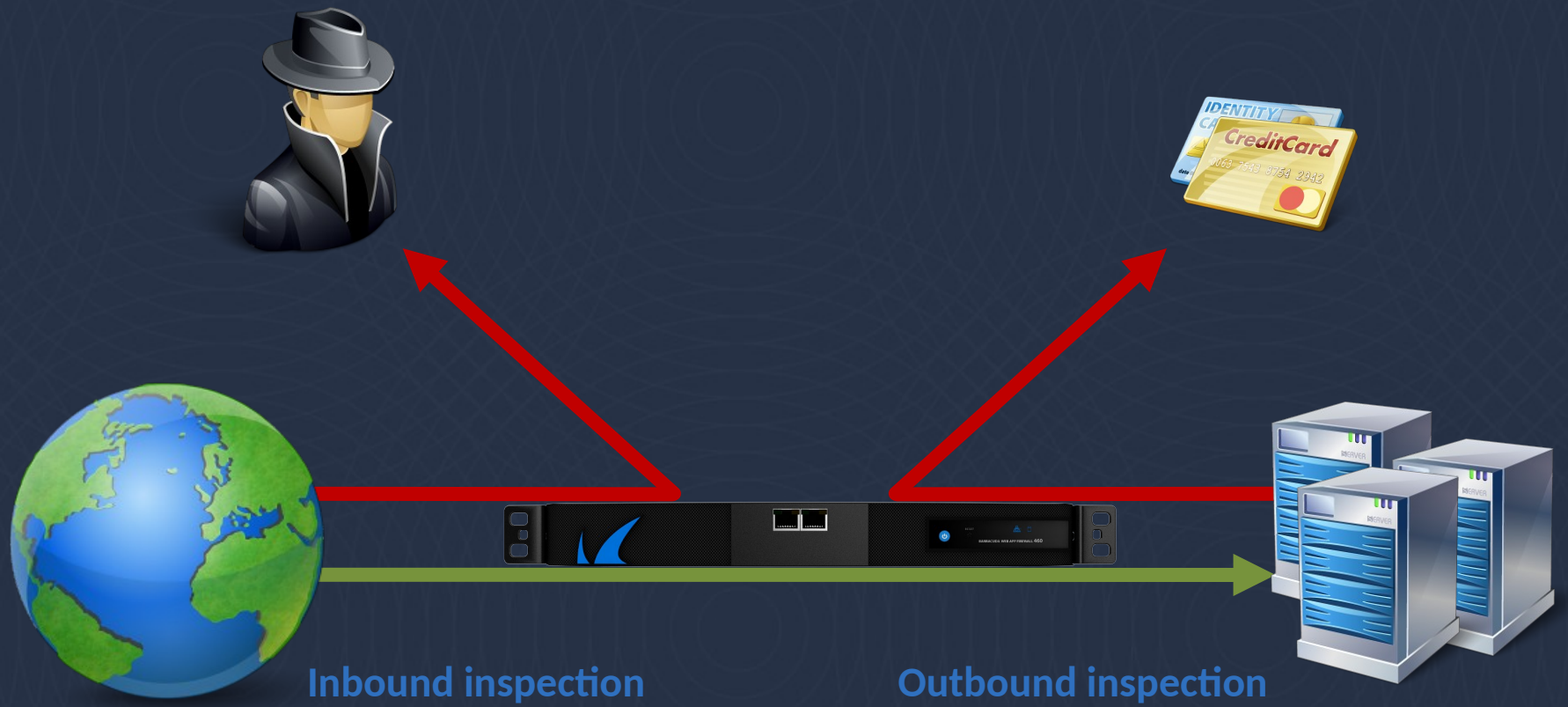
The solution: Layer 7 security



The solution: Barracuda Web Application Firewall

- ✓ Understands web traffic
- ✓ Layer 4 and Layer 7 **load balancing** for Web servers
- ✓ **Accelerates** application delivery
- ✓ **Protects** against common web attacks
- ✓ **Mitigates** broken access control

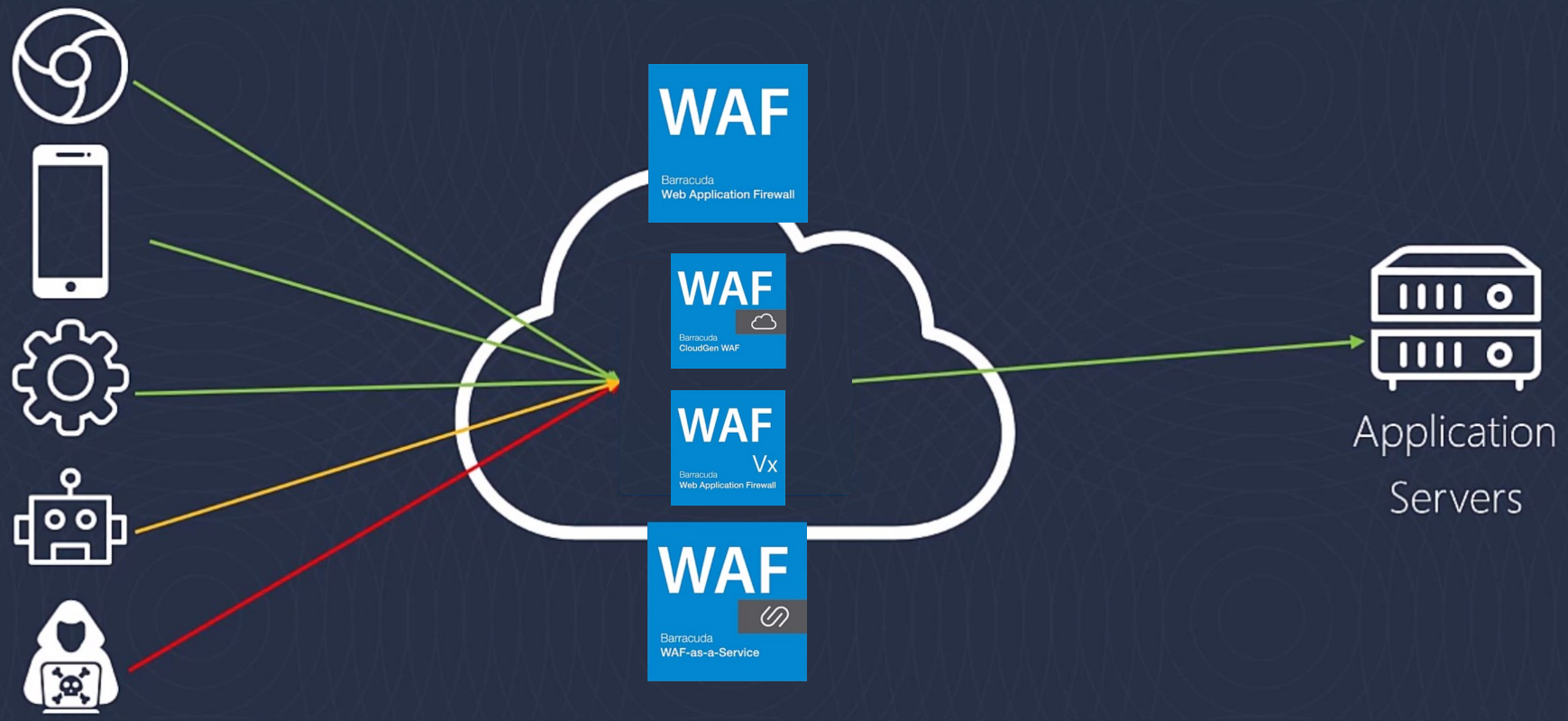
Layer 7 Web Application Firewall



Inbound inspection
(protect against layer 7 attacks)

Outbound inspection
(protect against data theft)

WAF or WAF-as-a-Service protects WWW application servers with a specific *policy* (application protection rules)



Sources of information for WAF policy

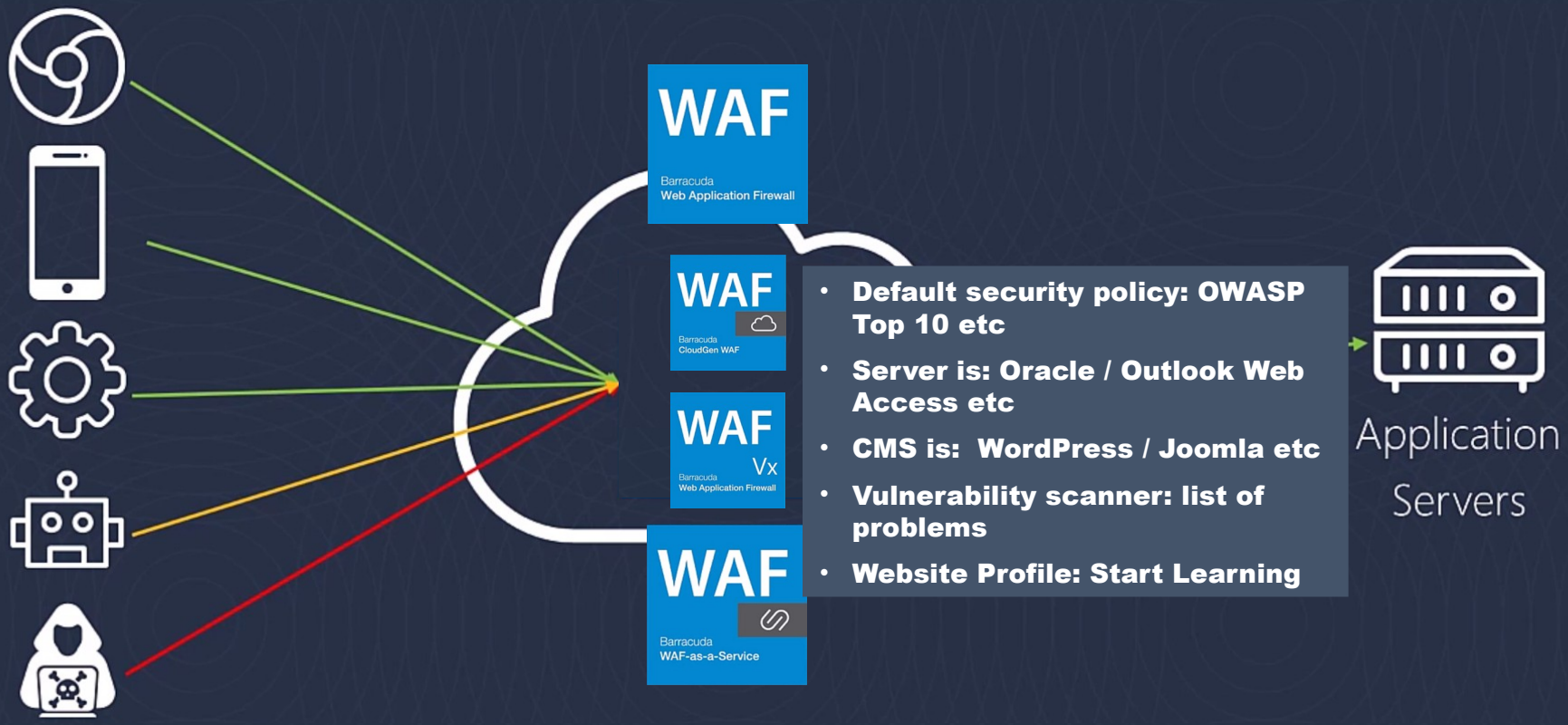
Knowledge of server side:

- General security policy applicable to any WWW server: detection of OWASP Top 10 attacks, masking response codes, blocking countries, blocking TOR anonymous access etc
- Content Management System knowledge: if my site is built on Wordpress then I can patch some well-known Wordpress vulnerabilities with specific security rules
- Application vulnerability scanning: if a vulnerability is detected then a blocking rule can be created
- Application profiling: WAF „learns” typical parameters and typical values used by application and then enforces them

Knowledge of client side (observing browsers’ behaviour):

- Fingerprinting browsers for selective blocking
- Watching attacks generated by browsers and assigning risk levels
- Classifying pseudo-browsers as good bots or bad bots

Server side knowledge and protection



- **Default security policy: OWASP Top 10 etc**
- **Server is: Oracle / Outlook Web Access etc**
- **CMS is: WordPress / Joomla etc**
- **Vulnerability scanner: list of problems**
- **Website Profile: Start Learning**

Attach Protection & Data Loss Prevention

Attack Protection:
Open Web Application Security Project (OWASP) top 10

- SQL, XSS, command injection

CSRF

Web Site Cloaking

Data Theft Protection

- Credit card, SSN, custom patterns

Session Protection

- Cookie encryption
- Parameter tampering protection

Integrated Anti-Virus

Brute Force Protection

DoS Protection

IP Reputation Blocking

- Blocking by Geo IP
- Anonymous Proxy Blocking

XML Firewall

- XML schema enforcement
- Web services security

SIEM Integration

Client side knowledge and protection

- **Client IP address**
- **Client Fingerprint**
- **Client is a browser**
- **Client is a Python script**
- **Client is a bot (Good? Bad?)**
- **Client tries to attack (Risk level?)**

++++++++++++++++++++++

Advanced Analysis Dashboard

- ***interface to BATIC Barracuda AppSec Threat Intelligence Cloud***

Application Servers

OWASP Top 10 and predefined security policies

Policy Overview

Name	Sync …	Limit Checks	Cookie Protection	URL Protection	Parameter …	Data Theft Prote…	Default Char…	Double Decoding	Allowed ACLs	Denied ACLs
default	▲	Yes	Signed	Enable	Yes	credit-cards ssn	UTF-8	No	1	8
oracle	▲	Yes	Signed	Enable	Yes	credit-cards ssn	UTF-8	No	1	1
owa	▲	Yes	Signed	Enable	Yes	credit-cards ssn	UTF-8	No	2	7
owa2010	▲	Yes	Signed	Enable	Yes	credit-cards ssn	UTF-8	No	2	7
owa2013	▲	Yes	Signed	Enable	Yes	credit-cards ssn	UTF-8	No	2	7
saml	▲	Yes	Signed	Enable	Yes	credit-cards ssn	UTF-8	No	1	8
sharepoint	▲	Yes	Signed	Enable	Yes	credit-cards ssn	UTF-8	No	1	7
sharepoint2013	▲	Yes	Signed	Enable	Yes	credit-cards ssn	UTF-8	No	1	7

Content Management Systems and predefined policy templates

BASIC | SECURITY POLICIES | WEBSITES | BOT MITIGATION | ACCESS CONTROL | NETWORKS | ADVANCED

Search help topics

Backups | Energize Updates | Firmware Update | Export Logs | System Logs | Templates | View Internal Patterns | Libraries
Admin Access Control | High Availability | Appearance | System Configuration | Secure Administration | Troubleshooting | Vulnerability Reports
CloudGen Firewall Settings | Cloud Control | Task Manager

Template Repository

Create Template | Import Template | Delete | Help

There are no templates available.

Factory Shipped Templates

Help

Show 10 entries

Search:

Name	Type	Format	Compatibility	Created	Actions
WordPress	WordPress	Bundle	11.0.0.001+	May 03 00:00:00 2021	Use
Joomla	Joomla	Bundle	11.0.0.001+	May 03 00:00:00 2021	Use
Drupal	Drupal	Bundle	11.0.0.001+	May 03 00:00:00 2021	Use
OpenCart	OpenCart	Bundle	11.0.0.003+	May 03 00:00:00 2021	Use
PrismWeb	PrismWeb	Bundle	11.0.0.001+	Mar 31 00:00:00 2021	Use
osCommerce	osCommerce	Bundle	11.0.0.001+	Mar 11 00:00:00 2021	Use
Magento	Magento	Bundle	11.0.0.001+	Mar 11 00:00:00 2021	Use
Typo3	Typo3	Bundle	11.0.0.001+	Mar 11 00:00:00 2021	Use

Showing 1 to 8 of 8 entries

Previous Next

Application profile: database of URL addresses, parameter names and acceptable values

Service | Help

Website
service-badstore (10.10.1.90:80)

Use Profile:	Yes	**Strict:**	Yes
Mode:	Learning	**Allowed Domains:**	badstore....atest.net
URLs excluded:	*.jpg, *....js, *.css	**URLs not reviewed:**	1 (Out of 1)
		Parameters not reviewed:	8 (Out of 8)

Stop Learning | Edit

Directories | More Actions | Help

- /
  - cgi-bin
    - badstore.cgi

URL Profiles | Page 1 of 1 | Filter | More Actions | Add URL | Help

URL	Hits	Last Changed	Status	Mode	Action
/cgi-bin/badstore.cgi	0	0h:1m:0s	On	Passive	Edit

Parameter Profiles | Page 1 of 1 | More Actions | Add Param | Help

Parameter	Type	Class	Action
action	Read Only	Generic	Edit
cartitem	Global Choice	Generic	Edit
email	Input	Generic	Edit
fullname	Input	Generic	Edit
passwd	Input	Generic	Edit
pwdhint	Global Choice	Generic	Edit
role	Read Only	Generic	Edit
searchquery	Input	Generic	Edit

Rules based on client behaviour

Client Details

Client IP	89.64.85.34
Client Port	52446
Client Fingerprint	88f75edf10f19cba22afdf2331d5744d
Country	PL
Host	waf1.cuda.link
User Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.182 Safari/537.36
Client Type	Unknown
Session ID	
Proxy IP	89.64.85.34
Proxy Port	52446

da.link/cg

Attack Action Name (ID)	Follow Up Action
Suspicious Client Profile Risk Threshold Breached(426)	Challenge with CAPTCHA

Attack Action Name (ID)	Follow Up Action
Bad Client Profile Risk Threshold Breached(427)	Block Client Fingerprint

WAF with out-of-the-box default security policy

https://bvm.barracudanetworks.com

Application security policy:

Default

Web Application

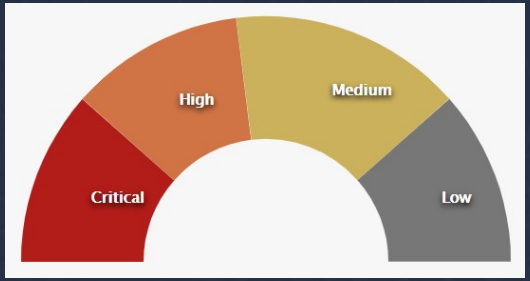
Users

Efficiency of the *Default* policy

Badstore scanned by BVM directly:

Results by severity level

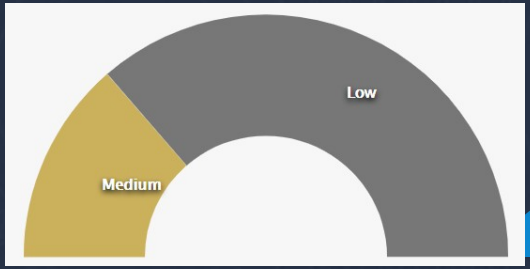
Critical	9
High	9
Medium	12
Low	9



Badstore behind *Default* security policy:

Results by severity level

Medium	3
Low	8



WAFs offer more than security

Security &
DDoS Protection

Load Balancing &
Server Health Monitoring

Logging & Reporting

Authentication &
Access Control

Session
Persistence

SSL & Performance
Acceleration

Thank You

Barracuda®
Your journey, secured.