1 For Ofer and other Reviewers

My (Achim) text for the WAFEC document starts in chapter 2 What is a WAF below. This paragraph gives some ideas behind the text and should help reviewers to understand what I've written. I've made some AH: comments in this format inside the text where I expect further discussions.

This document also contains the headlines of other “WAFEC 2 Outline” sections, so that I can simple reference them.

There is 7 References at the end, where I listed the external documents I referenced to.

Finally, I checked some resource in the web (i.e. http://en.wikipedia.org/wiki/Application_firewall) where I only agree partially. Are there any plans to update these resources to comply to our document (or the other way around)? According spelling: I capitalized all known terms; used backend, blacklist and witelist without a dash;

2 What is a WAF

This document is about Web Application Firewalls which are referred to in here with their commonly used acronym WAF.

AH: “WAFEC 2 Outline”‘s Introduction uses the acronym WAF. The definition for it is given in previous sentence here. Should this be mentioned in “Introduction”? Following the definition we provide the main Use Cases of modern WAFs.

2.1 Definition s

Simply giving a definition of WAF would miss 2 aspects: the technical definition and what the name (and acronym) stands for. So we explain technically How does a WAF work and explain historically Why the name WAF.

There are many definitions to be found elsewhere. As this document focuses on solutions to detect, mitigate, protect against attacks exploiting vulnerabilities in web applications and anything related to them, it's hard to give short definition on such versatile problems.

AH: Not sure if “versatile” is the proper adjective here, others are: multi-faceted, multifarious.

AH: Question: should we mention here, that a WAF does not filter, but log and pass or block?

[2] “A WAF is defined as a security solution on the web application level which – from a technical point of view – does not depend on the application itself.”
2.1.1 How does a WAF work (technical)

The technical definition is the most important to understand the content and purpose of the other parts of this document.

In general – as the name WAF - Web Application Firewall implies – a WAF operates on the application layer of the OSI model. The application layer protocol is HTTP, or HTTPS when it is tunneled using SSL/TLS.

As modern attacks often don't stop at one layer of the OSI model, so do modern WAFs which try to protect as much as possible. They therefore may cover other layers of the OSI model (mainly TCP/IP) also. They go beyond the OSI model and inspect data, formats and protocols transported inside HTTP (S) as well. They try to understand business logic and protect against exploits of logical flaws there.

AH: “try to understand business logic“ – this is a configuration issue in some WAFs, should we keep it here, or use “partially try to ...“?

This enables WAFs to detect attacks originated in one layer targeting another one (for example like DoS with Slowloris' "SYN flooding HTTP", SQL Injection inside JSON, REST, SOAP, ...).

Looking at the network or architecture topology, WAFs can be distinguished by following technologies:

- passive device (like an IDS)
- active device (mainly reverse proxy)
- plug-in/module in web or application server
- plug-in/module in application (embedded WAF)

Details for these deployment and operation modes will be described in 4 Deployment Options.

Looking from OSI layers, WAF detections and protections are against:

- attacking the web server (DoS on TCP/IP, HTTP)
- HTTP enforcement (W3C conformance, ...)
- SSL enforcement (ciphers, certificate usage, SSL Proxy to client and/or backend)

In all cases a WAF operates at a perimeter of a system. Such a perimeter can be next¹ to the traditional network firewall, next to load balancers, next to proxy and caching servers or close to the web or application server. They even can be used at all of these positions simultaneously.

WAFs can detect a huge amount of attacks against vulnerabilities in modern web applications and the related systems used there (multi tier architecture) and provide mitigations against these attacks. This distinguishes WAFs from patching insecure code, secure coding in each application and/or instrumenting application code by adding additional security code to enforce security checks.

In short this can be expressed as [2]:

“A WAF is defined as a security solution on the web application level which – from a technical point of view – does not depend on the application itself. ”

2.1.2 Why the name WAF (historical)

Historically the name WAF took quite a while until it was widely accepted. The historical description of the name and acronym is more informative but gives a good impression why so many functionality is provided by modern WAFs and hence make it difficult to give a brief,

---
¹ “next” can be “in front” and/or “behind” if seen from the traffic flow
unique and short definition.
After traditional network firewalls became well accepted security systems and were in common use, the lack of proper protections at the application layer was identified. A couple of names for systems securing the application layer have been used over the years until the name “Web Application Firewall” with its acronym “WAF” have been settled.

The following names [1] were already in use back in 2005:
- Adaptive Firewall
- Adaptive Proxy
- Adaptive Gateway
- Application Firewall
- Application Level Firewall
- Application Level Security Gateway
- Application Security Device
- Application Security Gateway
- Stateful Multilayer Inspection Firewall
- Web Adaptive Firewall
- Web Application Firewall
- Web Application Security Device
- Web Application Proxy
- Web Application Shield
- Web Security Firewall
- Web Security Gateway
- Web Security Proxy
- Web Intrusion Detection System
- Web Intrusion Prevention System

Today there are more names, like
- Security Gateway
- Next Generation Firewall
- Intrusion Detection Firewall
- x

All these names qualify the purpose of a WAF, at least for a special usage or as a special brand of a vendor.

Modern WAFs work as Intrusion Detection System (IDS) also (see 2.2 Use Cases below), which should be better named Web Intrusion Detection System (but again the community uses simple acronyms). This is today (2012) often named Next Generation Firewall, which may be a bit misleading.

This document is about the WAFs as described in 2.1.1 How does a WAF work (technical).

2.2 Use Cases

There are a variety of use cases for WAFs. While attack mitigation is the most common purpose for a WAF, attack detection (like an IDS) and virtual patching to mitigate against newly discovered vulnerabilities are other use cases.

2.2.1 Logging and Troubleshooting

AH: Don’t see a purpose for this paragraph. Should be described elsewhere (partially done in “Attack Detection”) ...

2.2.2 Attack Detection

AH: Note: all examples here fall far short of completeness. Should we mention this, or provide more references to other herein for details? Another option would be to avoid examples here and reference to other chapters only.

Detecting attacks is the main functionality a WAF needs to provide. Without a proper method to detect attacks, a WAF can not mitigate them, obviously. A WAF may operate in detection only mode, which then is a passive mode and does not harm any traffic. Logging all detections
helps to identify malicious data and the targets being attacked, just like an IDS. Logging detected attacks also helps to properly configure the mitigations provided by the WAF.

WAFs can detect a huge amount of attacks against vulnerabilities in modern web applications, such as:

- generic Data Validation (length, range, constants, Format String Attacks, pattern according whitelists and/or blacklists)
- Injection Attacks (namely: Cross-Site Scripting, SQL Injection, Code Injection, Command Injection, LDAP Injection, XML Injection, XPath Injection)
- URL / Path Traversal
- Forceful Browsing
- Session Hijacking
- DoS (Layer 7)
- HTTP Header Tampering (Cookie Poisoning)
- HTTP Parameter Pollution
- Information Leakage

They even can detect attacks, such as:

- HTTP Response Splitting
- HTTP Request Smuggling

which exploit improper configurations of related and connected systems.

AH: Need to add references here to other chapters herein.

In short: a WAF inspects traffic on the application level detects potential attacks based on its rule set.

### 2.2.3 Attack Mitigation

In general the purpose of a WAF – if not used in passive detection only mode – is to block inbound malicious traffic, while all legal traffic is passed unmodified. In some circumstances it can also be useful to mangle inbound traffic instead of completely blocking it.

All mitigations rely on properly detecting attacks (either due to malicious data or by exploiting improper configured systems). However, while a WAF mainly provides full generic protections against attacks (see Attack Detection above), it can only partially protect against following attacks:

- Privilege Escalation
- Bypass Business Logic
- Brute Force Attacks

WAFs have implemented different methods and models as countermeasures against these attacks and vulnerabilities. Depending on the technology of the WAF, the range of these protections differ:

- W3C conformance checks
- Data Validation according (length, range)
- Data Signing, Data Encryption
- Data Hiding (intermediate store)
- Pattern Matching with whitelists and blacklists
- URL encryption

AH: Need to add references here to other chapters herein.

2 note that most modern WAFs do not use simple signature-based blacklists (like AV software) but generic patterns, for details see <<insert chapter here>>

3 note that most modern WAFs do not use simple signature-based blacklists (like AV software) but generic patterns, for details see <<insert chapter here>>
• additional tokens (for example anti-CSRF)
• Site Usage Enforcement
• abstract session handling
• heuristic methods based on usage and provided client data
• DoS (Layer 7) blocking based on IP, user, transaction; termination of user sessions
• Output Cloaking

AH: Need to add references here to other chapters herein.

When the WAF also provides its own session handling, it can protect against
• authentication bypass
• unauthorised access to objects
• insecure direct object references

AH: Need to add references here to other chapters herein.

In short: a WAF inspects traffic on the application level and passes or blocks requests based on
its rule set.

2.2.4 Virtual Patching

“A security policy enforcement layer which prevents the exploitation of a known vulnerability.” [3]

The term virtual patching is often used in conjunction with WAFs even it's not originated with
web application security. As with WAFs itself (see 2.1.2 Why the name WAF (historical)),
there exists a few other names like Hot Patching, External Patching or Just-in-time Patching.

As “A WAF is defined as a security solution on the web application level which does not
depend on the application itself” (see 2.1.1 How does a WAF work (technical)), a virtual patch
is a (configuration) rule in the WAF which also does not depend on the actual source code of
the application itself. This rule prevents an exploitation attempt to reach the application.

With virtual patching it is possible to operate web applications with known vulnerabilities, for
example:
• newly identified vulnerabilities in applications
• known vulnerabilities in legacy applications or third-party applications
• protecting systems, which cannot be taken offline
• emergency patching for newly discovered exploits

For a more comprehensive description, please see

AH: Ryan, please add your comments here ;–)

2.3 Minimum Requirements

AH: I'm unsure what someone had in mind to describe here, hence I omitted it for
now.

AH: References I found are:
http://lists.webappsec.org/pipermail/wasc-wafec_lists.webappsec.org/2012-June/000082.html

AH: End of my text here, all following for references only.
3 Security

4 Deployment Options

4.1 Method of Delivery

4.2 Network Integration

4.3 Application Support

5 Supporting Features

6 Appendices

7 References

[1] this list was compiled 2005 by Achim Hoffmann and published by Ivan Ristić in
“Web Application Firewalls: When are they useful”
http://www.modsecurity.org/documentation/Web_Application_Firewalls_-_When_Are_They_Useful.pdf
http://www.owasp.org/images/9/9c/OWASPAppSecEU2006_WAFs_WhenAreTheyUseful.ppt


[3] Virtual Patching