



Deep Dive On F5 „on and off prem“ Application Protection

Sven Müller – Security Solution Architect



Important Trends: Patching Vulnerabilities Remain An Issue



Vulnerabilities help make Web application attacks amongst the leading causes of data breaches

86% of websites has at least 1 vulnerability and an average of 56 per website WhiteHat Security Statistics Report 2013

99% of vulnerabilities were compromised a year after the vulnerability was made public (CVE)

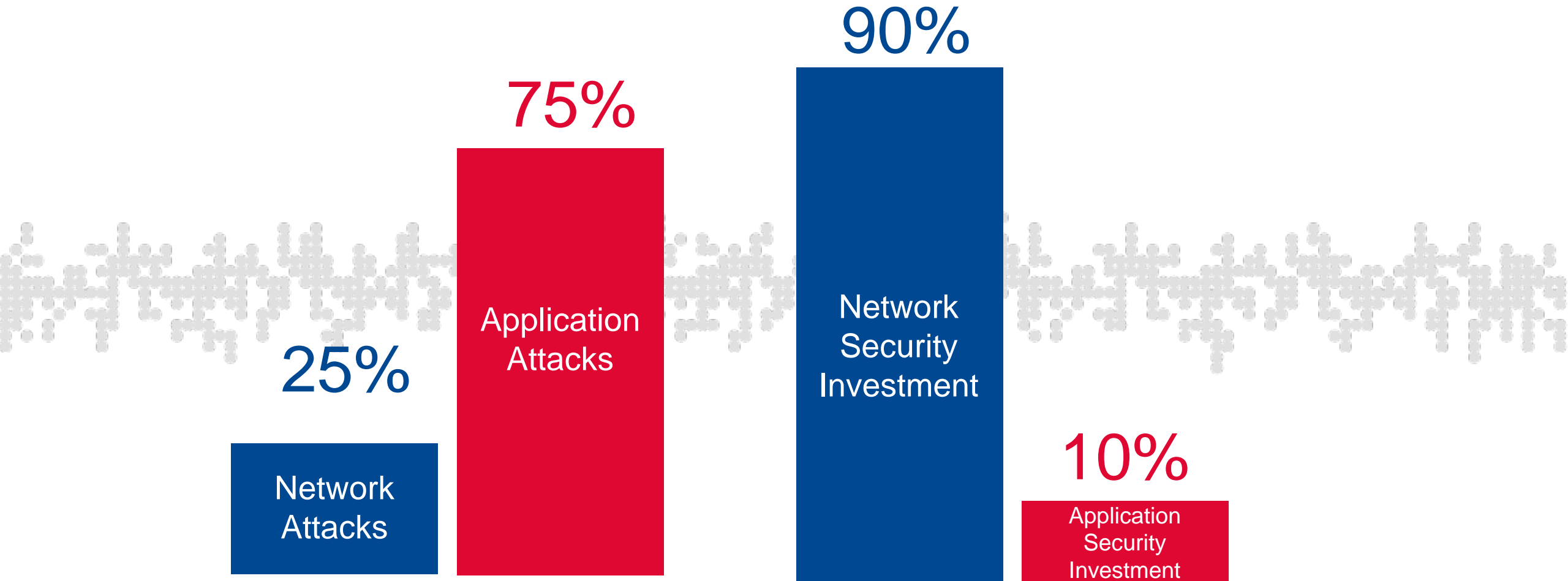
10 CVE's account for 97% of the exploits observed in 2014

Less than 49% of companies have an organized effort for patching

2015 Cisco Annual Security Report

What's Going on in the Market?

Today's Security Investment Doesn't Address the Big Problem



Source: Gartner

Application Security Manager (ASM)



BIG-IP ASM: Leadership in WAF

Traditional WAF

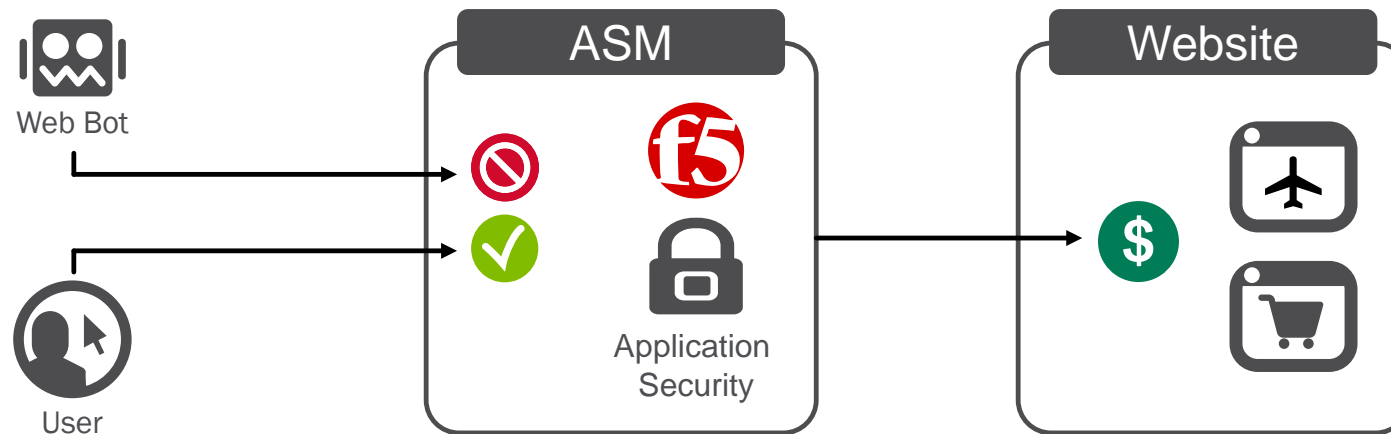
- Signatures (OWASP Top 10)
- DAST Integration
- Site Learning
- File/URL/Parameter/Header/Cookie Enforcement
- Protocol Enforcement
- Login Enforcement / Session Tracking
- Data Leak Prevention
- Flow Enforcement
- IP Blacklisting

Advanced WAF

- Bot Detection
- Client Fingerprinting
- Web Scraping Prevention
- Brute Force Mitigation
- L7 DDoS Protection
- Heavy URL Mitigation
- CAPTCHA Challenges
- HTTP Header Sanitization/Insertion
- Anti-CSRF Token Insertion
- PFS Ciphers

Highly accurate anti-bot and scanner protection

- Differentiate between script and browser
- Inspection of user interaction with browser
- Distinguish real-user from bot
- Mitigate automated attacks, scanners, botnets and intellectual property scrappers
- Detect a persistent scrapper that uses multiple ip addresses or a single request session



Fingerprinting, DeviceID

- Collects browser attributes
 - Screen resolution
 - Time zone
 - Default fonts
 - User agent
 - Installed plug-ins
 - ...
- Associates collected information and browser behavior to identify suspicious clients



<http://browserspy.dk/>

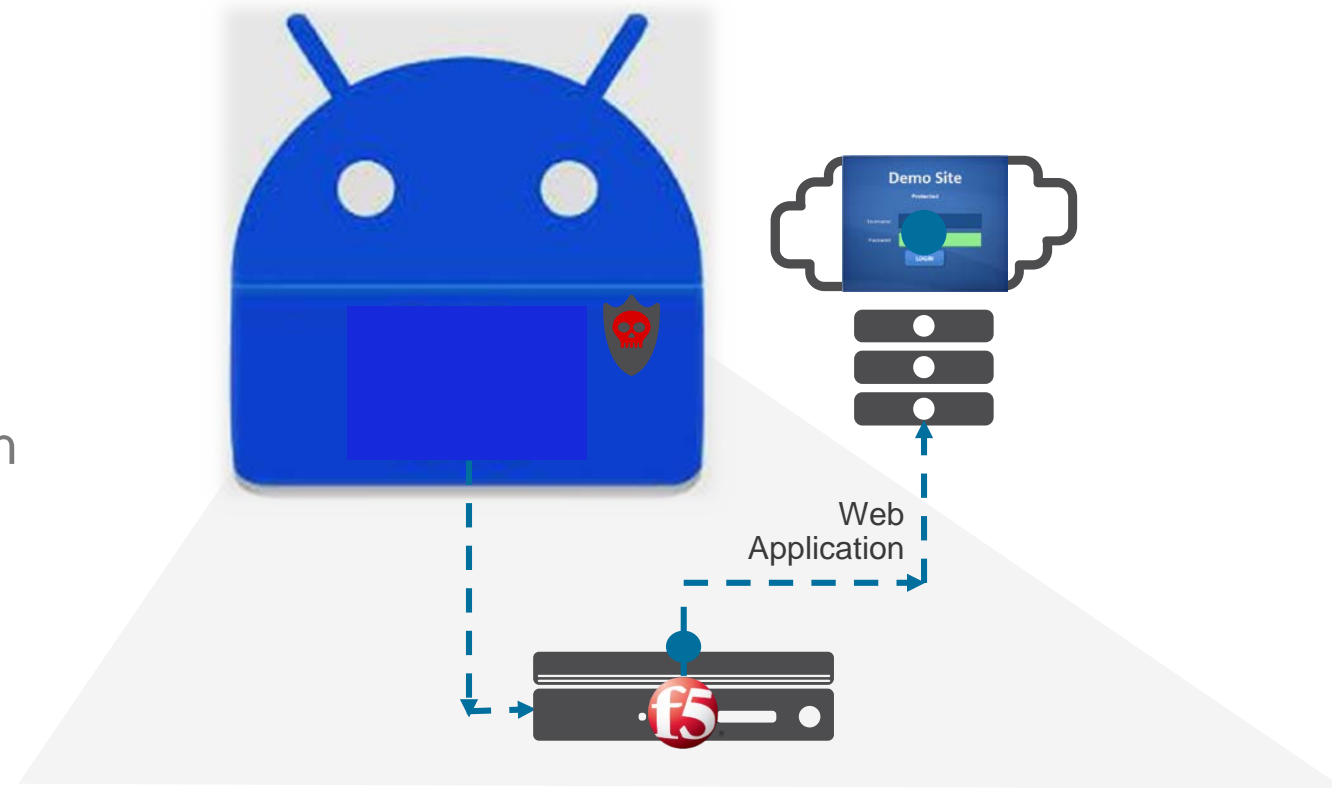
<https://panoptickick.eff.org/>

<http://jcarlosnorte.com/security/2016/03/06/advanced-tor-browser-fingerprinting.html>

ASM's unique Proactive Bot defense

Stop automated attacks from ever materializing

- Enables always-on protection that preempts attacks
- Compliments existing reactive protections
- Utilizes advances detection methods and techniques CAPTCHA challenges & geolocation enforcement
- Categorize BOTs detected by signature classification to distinguishes good Bots from malicious offenders
- Detect headless browsers that run JS



Defend against automated non-human webscraping, DDoS and Brute force attacks

Detecting bots and blocking

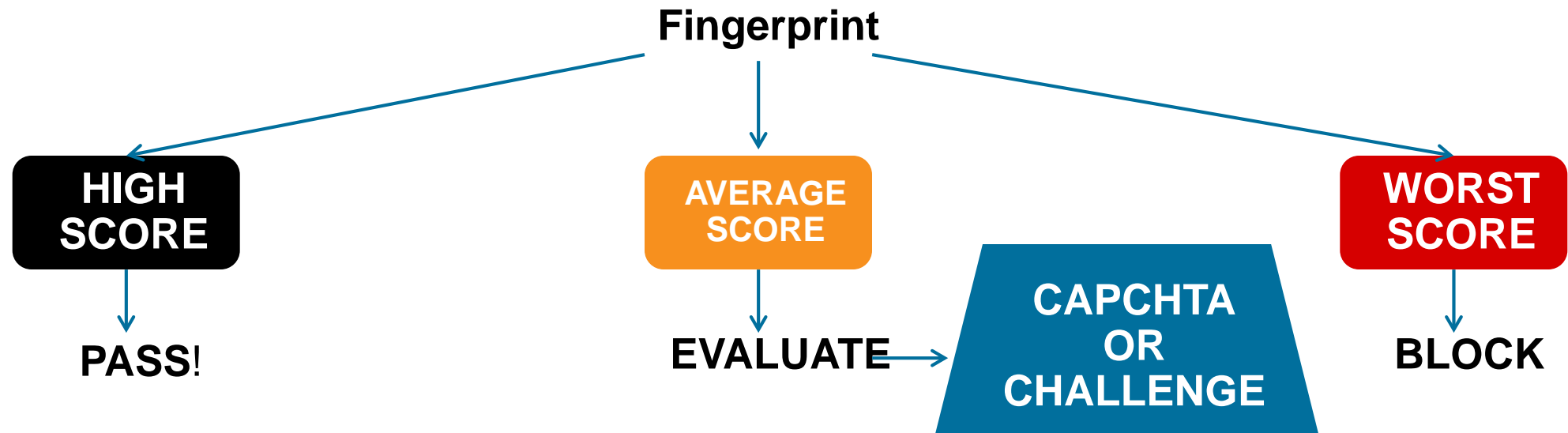
Block requests from
suspicious browsers

Strengthen the bot defense by
blocking suspicious browsers. Highly
suspicious browsers are completely
blocked, while moderately suspicious
browsers are challenged with
CAPTCHA.

- ✓ Block Suspicious Browsers
- ✓ CAPTCHA Challenge

[CAPTCHA Settings](#)

- When checked, ASM will fingerprint and score the browser and check multiple variables to determine if it is a bot



DDOS Profile

⚙️ DoS Profile Properties

Profile Information

General Settings

Application Security

General Settings

Proactive Bot Defense Off

Bot Signatures Off

TPS-based Detection

Stress-based Detection

Heavy URL Protection

Record Traffic Off

Protocol DNS

General Settings Off

Protocol SIP

General Settings Off

Network

General Settings Off

Application Security » Stress-based DoS Detection [Edit All](#)

This section configures the detection of DoS attacks based on server stress.
The system automatically detects an increase in server stress and mitigate DoS attacks causing it.

Operation Mode	Specifies how the system reacts when it detects an attack.	<div>Blocking ▼</div>	Close
How to detect attackers and which mitigation to use	By Source IP	Mitigation: Request Blocking (Rate Limit)	Edit
	By Device ID	Consider a Device as an attacking entity if the following conditions occur: TPS increased by: <div>500</div> % and reached at least <div>40</div> transactions per second OR TPS reached: <div>200</div> transactions per second Set default criteria	Close
	Select mitigation methods to use on the attacking Device's: <div><input type="checkbox"/> Client Side Integrity Defense <input type="checkbox"/> CAPTCHA Challenge <input type="checkbox"/> Request Blocking</div>		
	By Geolocation	No mitigation	Edit
	By URL	Mitigation: Request Blocking	Edit
Site Wide	No mitigation Edit		
	Behavioral	<input checked="" type="checkbox"/> Enabled Enables traffic behavior, server's capacity learning, and anomaly detection. <div>No mitigation ▼</div> Learns and monitors traffic behavior, but no action is taken.	Close
Prevention Duration	Specifies the time spent in each mitigation step until it is stopped, and the next one is started.	Escalation Period: 120 seconds De-escalation Period: 7200 seconds	Edit

Traffic Learning

Security » Application Security : Policy Building : Traffic Learning

Traffic Learning

Enforcement Readiness

Learning and Blocking Settings

Current edited policy: testen (blocking) Apply Policy

☐ Q

Score

Highest

Attack signature detected

Parameter: q

20%

Attack signature detected

Policy Signature: textarea tag (Parameter)

20%

Attack signature detected

Parameter: q

20%

Attack signature detected

Policy Signature: CreateTextFile() (Parame...

20%

Attack signature detected

Parameter: q

20%

Attack signature detected

Policy Signature: OpenAsTextStream() (Pa...

20%

Attack signature detected

Parameter: q

20%

Attack signature detected

Policy Signature: asfunction: (Parameter)

20%

Attack signature detected

Parameter: q

20%

Attack signature detected

Policy Signature: livescript (Parameter)

20%

Attack signature detected

Parameter: q

20%

Attack signature detected

Parameter: q

20%

Accept Suggestion

Delete Suggestion

Ignore Suggestion

Related Suggestions

Action: Add textarea tag (Parameter) (disabled on the parameter) to Overridden Attack Signatures.
Matched Parameter: q
Matched Attack Signature: 200001414 - textarea tag (Parameter)

2 sample requests out of 2 that triggered the suggestion from 2015-06-18 02:17:57 until 2015-06-28 00:34:02
Average Request Violation Rating 2.0 At least 1 different IP / 1 different session

[HTTP] /search.php

192.168.188.58

3

[HTTP] /search.php

192.168.188.58

1

General Data

Request

Response

Attack signature detected

Requested URL	[HTTP] /search.php
Support ID	8189134291251246184
Time	2015-06-18 02:30:42
Request Status	Blocked
Severity	Error
Violation Rating	3 Request needs further examination
Response Status Code	N/A
Attack Types	Cross Site Scripting (XSS)
Username	N/A
Session ID	cad6ad854dee47f9
Source IP Address	192.168.188.58:59043
Destination IP Address	172.29.46.36:80
Geolocation	N/A

Demo: Evasion Techniques



WebSocket Security



HTTP and Real Time Updates

Why does HTTP run into difficulties?

- HTTP is half duplex
- It is primarily designed for document sharing and not for interactive applications
- The protocol overhead is big, especially if the message (payload) is a short

Introducing WebSocket

- TCP based, bi-directional, full-duplex messaging
- Part of HTML5
- IETF-defined Protocol: RFC 6455
- W3C defined JavaScript API
- Uses HTTP upgrade handshake
- Supports HTTP proxies, filtering, authentication and intermediaries

How does it work?

- Starting from an HTTP connection the clients needs to “update” the connection to another protocol, which is WebSocket

Request:

GET ws://echo.websocket.org/?encoding=text HTTP/1.1

Host: echo.websocket.org

Upgrade: websocket

Connection: Upgrade

Origin: http://websocket.org

Cookie: __utma=99as

Sec-WebSocket-Key: uRovscZjNol/umbTt5uKmw==

Sec-WebSocket-Version: 13

} Request upgrade to WebSocket connection

} WebSocket Handshake headers

How does it work?

- **Response:**

HTTP/1.1 101 WebSocket Protocol Handshake

Date: Fri, 10 Feb 2012 17:38:18 GMT

Connection: Upgrade

Server: Kaazing Gateway

Upgrade: WebSocket

Access-Control-Allow-Origin: <http://websocket.org>

Access-Control-Allow-Credentials: true

Sec-WebSocket-Accept:

rLHCkw/SKsO9GAH/ZSFhBATDKrU=

Access-Control-Allow-Headers: content-type

At this point the HTTP connection breaks down and is replaced by the WebSocket connection over the same underlying TCP/IP connection. The WebSocket connection uses the same ports as HTTP (80) and HTTPS (443), by default.

Demo: WebSocket Security



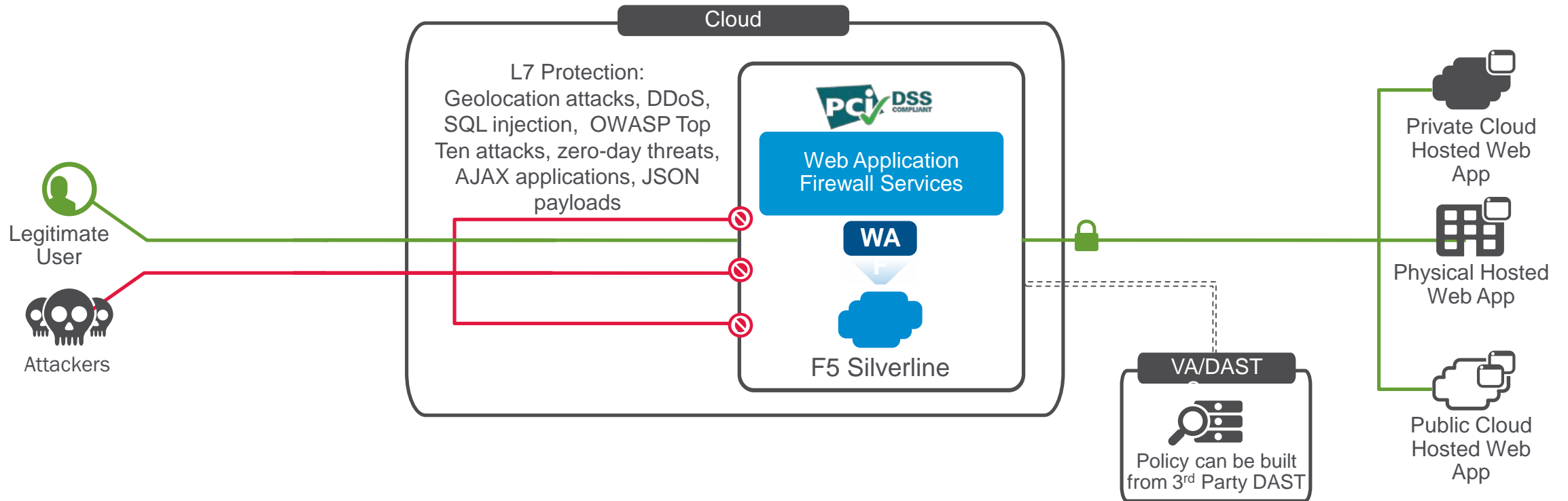
Silverline: WAFaaS



Silverline Web Application Firewall

Proven security effectiveness as a convenient cloud-based service

Protect web applications and data from layer 7 attacks, and enable compliance, such as PCI DSS, with the Silverline Web Application Firewall service which is built on BIG-IP Application Security Manager and backed by 24x7x365 support from F5 experts.



Global Coverage



24/7 Support

F5 Security Operations Center (SOC) is available 24/7 with security experts ready to respond to DDoS attacks within minutes

- Seattle, WA US
- Warsaw, Poland

Global Coverage

Fully redundant and globally distributed data centers world wide in each geographic region

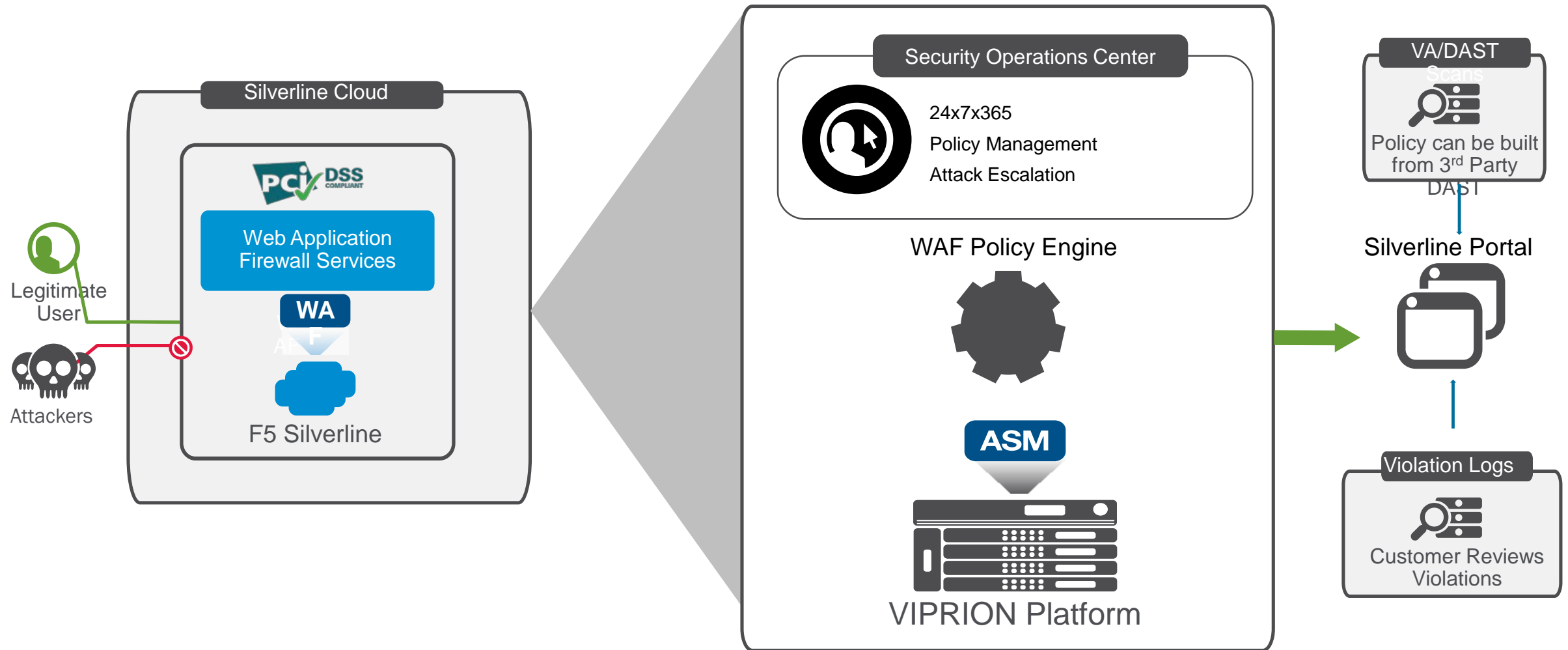
- San Jose, CA US
- Ashburn, VA US
- Frankfurt, DE
- Singapore, SG

Industry-Leading Bandwidth

- Attack mitigation bandwidth capacity over 2.0 Tbps
- Scrubbing capacity of over 1.0 Tbps
- Guaranteed bandwidth with Tier 1 carriers

Silverline Web Application Firewall

Proven security effectiveness as a convenient cloud-based service





SOLUTIONS FOR AN APPLICATION WORLD