Application Protection Report:
2019 Data Breach Landscape

Sander Vinberg,
Threat Research Evangelist
Application Protection Research Series

Using data to unite tactics and strategy in risk-based security
Application Protection Research Series

2019 Conclusions

- PHP, the weak point of the Internet
- Attack methods follow business models
- Injection, rejuvenated
- Access attacks predominant
- APIs changing the landscape
Applications are the business
Applications are the gateway to your data.
How Are Applications Targeted?

- Dynamic web page generators
- HTTP headers and cookies
- Admin interfaces
- Helper apps on client (java, flash)
- Backend connections through the server (injection)
- Data/active content pools— the data that populates and drives pages
- Server-side features such as search
- Sub domains hosting other versions of the main application site
- Cookies/state tracking mechanisms
- Web service methods
- Data entry forms
- Events of the application— triggered server-side code
- Shells, Perl/PHP
- APIs
- Administrative and monitoring stubs and tools
- Apps/files linked to the app
- Web pages and directories
- Backend connections through the server
How Can We Organize This Better?

SERVICES
- Dynamic web page generators
- Admin interfaces
- HTTP headers and cookies
- Data entry forms
- Events of the application triggered server-side code
- Web service methods
- Cookies/state tracking mechanisms
- Server-side features such as search
- APIs
- Administrative and monitoring stubs and tools
- Data/active content pools—the data that populates and drives pages

ACCESS
- Sub domains hosting other versions of the main application site
- Apps/files linked to the app
- Helper apps on client (java, flash)

TLS/SSL
- Backend connections through the server (injection)
- Data/active content pools

NETWORK
- Web pages and directories
- Shells, Perl/PHP
- Data entry forms
- Admin interfaces
- Server-side features such as search
Certificate spoofing
Protocol abuse
Session hijacking
Key disclosure
DDoS
Eavesdropping
Man-in-the-middle
Session hijacking
Malware
Cross-site request forgery
Cross-site scripting
Man-in-the-browser
Session hijacking
Malware
DDoS
DNS hijacking
Dictionary attacks
DDoS
DDoS
Eavesdropping
Protocol abuse
Man-in-the-middle
API attacks
Injection
Malware
Cross-site request forgery
Man-in-the-middle
Abuse of functionality
Credential theft
Credential stuffing
Session hijacking
Brute force
Phishing
DDoS
Key disclosure
Protocol abuse
Session hijacking
Certificate spoofing
TLS
The unauthorized access involved the insertion of rogue code into our checkout page.
The unauthorized access involved the insertion of rogue code into our checkout page.
Caveats for Public Breach Notifications

- Small $n$
- Limited detail
- Reductive categories
- Reductive causes
2019 US Breaches by Cause (%)

- Access related: 51.8%
- Web: 18.9%
- Accidental: 13.0%
- Physical: 6.1%
- Malware: 6.6%
- Insider: 3.1%
- Third-party compromised: 0.5%
2018 Breach Root Causes

<table>
<thead>
<tr>
<th>Industry</th>
<th>Government</th>
<th>Manufacturing</th>
<th>Tech</th>
<th>Non-profit</th>
<th>Health</th>
<th>Education</th>
<th>Services</th>
<th>CPA*</th>
<th>Finance</th>
<th>Retail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>36.40%</td>
<td>39.40%</td>
<td>41.94%</td>
<td>4.76%</td>
<td>7.08%</td>
<td>3.23%</td>
<td>14.29%</td>
<td>1.50%</td>
<td>71.88%</td>
<td></td>
</tr>
<tr>
<td>Access</td>
<td>22.70%</td>
<td>33.30%</td>
<td>33.28%</td>
<td>42.86%</td>
<td>42.48%</td>
<td>51.61%</td>
<td>52.75%</td>
<td>78.30%</td>
<td>66.91%</td>
<td>15.63%</td>
</tr>
</tbody>
</table>
2019 Breach Root Causes

- **Web (mostly injection)**
  - Government: 26.7%
  - Manufacturing: 36.5%
  - Tech: 18.8%
  - Non-profit: 10.8%
  - Health: 2.4%
  - Education: 3.8%
  - Services: 11.8%
  - CPA*: 2.7%
  - Finance: 0.0%
  - Retail: 82.4%

- **Access (mostly phishing and email)**
  - Government: 46.7%
  - Manufacturing: 32.7%
  - Tech: 59.4%
  - Non-profit: 56.8%
  - Health: 57.6%
  - Education: 67.3%
  - Services: 62.5%
  - CPA*: 84.0%
  - Finance: 62.3%
  - Retail: 9.3%
Web Attacks:

Don’t fix it if it ain’t broke
Stolen data exfiltrated via HTTPS to a drop server.

Targeted Site

Malicious PHP Code

Payment Card Info Breached

Injects usually due to weak input filters common in PHP, SEO CMS.

Can add fake fields to page

Targeted Site
A typical shopping site
Third party widgets/content

All linked off main app site but hosted elsewhere...
Third party widget injects not seen by WAF

1. Attacker discovers PHP vulnerabilities on widget
2. Attacker injects code into widget
3. Code runs on widget
4. Customer enters payment card info on site
5. Payment card info is sent offsite to attacker
Access Attacks:
Primary cause of breach

- Brute force
- Credential Stuffing
- Phishing
2018 Access Attacks Broken Down

- 20.0% Email cited as cause
- 19.6% Phishing to gain access to email
- 4.0% Stolen credentials
- 2.2% Credential stuffing and brute force
2019 Access Attacks Broken Down

- **32.8%** Email cited as cause
- **14.2%** Phishing gained access to email
- **1.9%** Stolen credentials
- **2.2%** Credential stuffing and brute force
Brute Force attacks by industry from reported 2018 F5 SIRT incidents

- Telecom: 22.2%
- Retail: 23.1%
- Tech: 23.5%
- Service: 25.0%
- Education: 27.3%
- Health: 41.7%
- Finance: 47.8%
- Public: 50.0%
Breach Data Conclusions

- Access attacks predominant except for retail
- Retail breaches increasingly dominated by formjacking
- Breach modes driven more by application architecture than by traditional sector
API breaches

API incident
Trends
Apps Expose their APIs

Server 1 → Ship → Elastic

Server 2 → Ship → Logstash

Server 3 → Ship → Logstash

Logstash → Parse filter and transform

Elastic → Visualize data → Kibana

© 2020 F5 Networks
API Breaches

**Basic Security Fails**
1. Authentication
2. Injection
3. Permissions

**Attack**
1. Mobile Apps
2. Direct APIs

---

RESTful API allowed Facebook credential spoofing attack on Tinder via public API.

© 2020 F5 Networks
API Mobile Attacks

User

Attacker

Desktop

OR

Mobile

Brute Force, Injects, ETC

Data

© 2020 F5 Networks
Protecting Applications

Survey (2018)

Recommendations
Confidence in Application Inventory

- Very Confident: 16%
- Confident: 22%
- Somewhat Confident: 24%
- No Confidence: 38%

F5 Ponemon Survey

© 2020 F5 Networks
App Vulnerability Test Frequency

- Annually: 9%
- Twice a Year: 5%
- Every 3 Months: 7%
- Every Month: 5%
- Every Week: 9%
- Every Time the Code Changes: 14%
- Not Pre-scheduled: 24%
- Unsure: 2%
- None: 25%
Barriers to Better App Defense

- Lack of Visibility in the App Layer: 59%
- Migration to the Cloud Environment: 44%
- Lack of Skilled or Expert Personnel: 44%
- Proliferation of Mobile Devices: 41%
- Turf/Silo Issues Between IT Security and App Dev: 32%
- Lack of Budget: 30%
- Insufficient Software Testing Tools: 19%
- Not a Priority: 13%
- Lack of Exec Level Support: 8%
- Lack of Leadership: 7%
- Other: 4%

F5 Ponemon Survey
Mitigation Recommendations

- **Program maturity**
  - Inventory
  - Vulnerability management
  - Change control
  - Access control
  - Training
  - Monitoring and Logging

- **Multifactor Authentication**

- **WAF / WAAP**
Mobile apps do not support the same security capabilities as web browsers.

Mobile APIs are often left under-protected.

Automated attacks targeting mobile applications:
- Content scraping
- Denial of serving
- API attacks
Training Reduces Phishing Success!

33% without training.

13% with training.

Phishing success without training.

Phishing success with training.
Tell us what you want to read about

@F5Labs