Identification of potentially harmful requests directed at web sites

Marek Zachara (http://marek.zachara.name/en/)

OWASP AppSecEU 15
Amsterdam, The Netherlands

The sad state of web security

- IVIZ: 99% of tested applications (5000) have at least one vulnerability, 35 on average
- WhiteHat: 86% of tested applications are vulnerable
- Symantec: 78% websites with vulnerabilities, 16% critical
  - In just one month (05.2012) LizaMoon was responsible for at least a million successful SQL Injection attacks

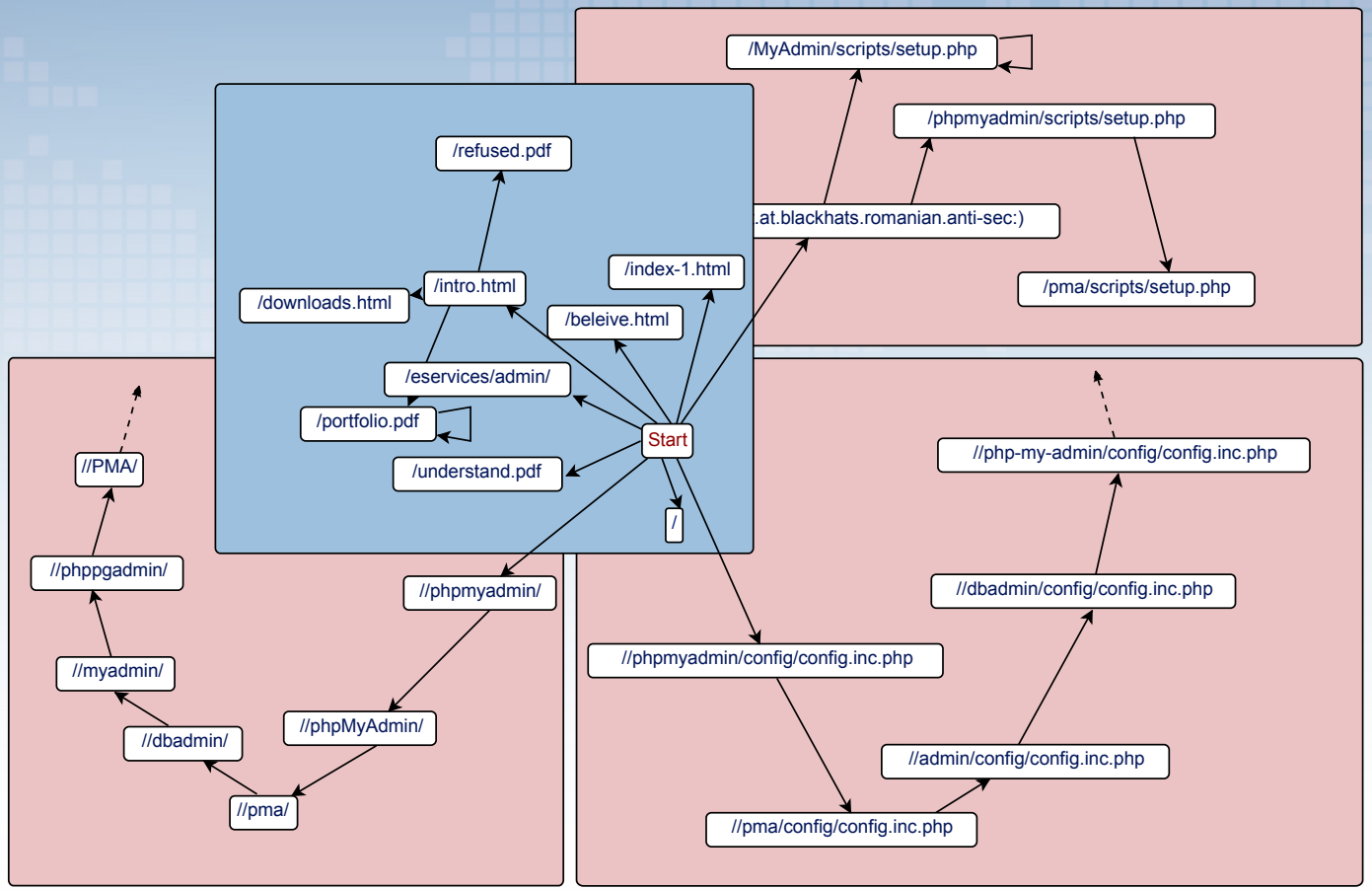
Scope of the method

- There are several classes of the attacks
- The method discussed focuses on the abuse of path traversal:
  - Multiple attempts
  - Forceful browsing
  - Unusual usage patterns

Part 1

Behavior modeling

A real-life scenario

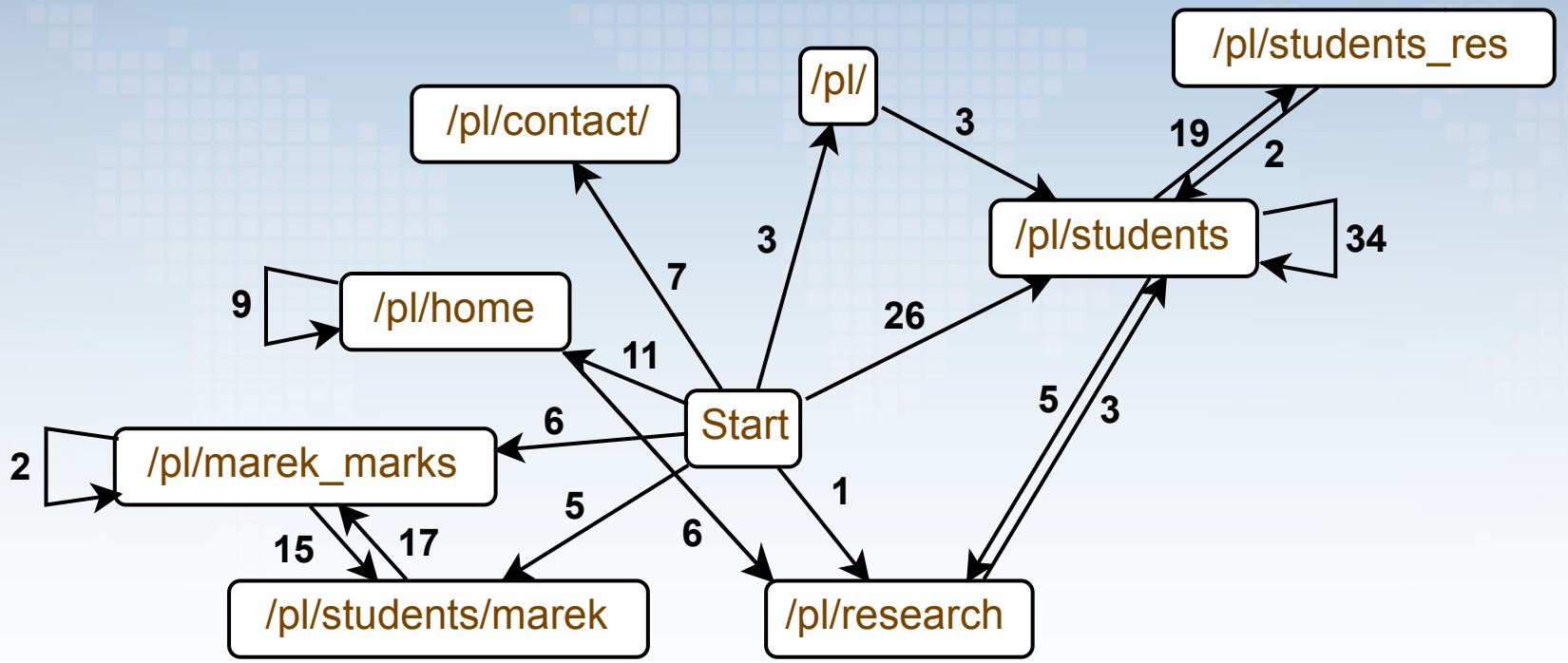


OWASP AppSecEU 15
Amsterdam, The Netherlands

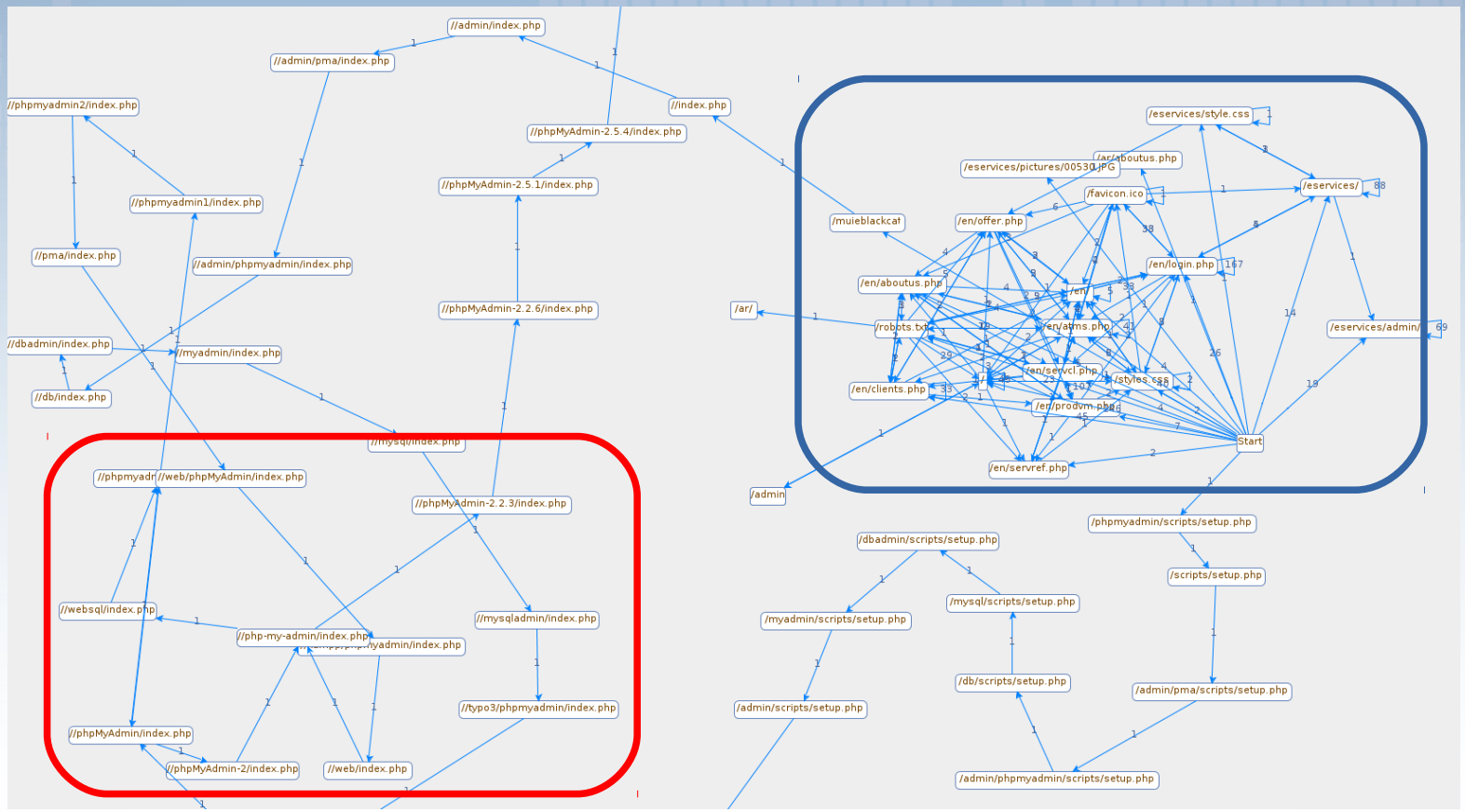
Behavior analysis

- With large enough sample of data it is possible to identify typical usage patterns
- These patterns can be derived from the frequency of transitions between pages (URLs), which in turn can be represented as weights of a graph's edges
- Unusual behavior can be detected by the lack of supporting graph edge (or its low weight)

Sample log-based graph

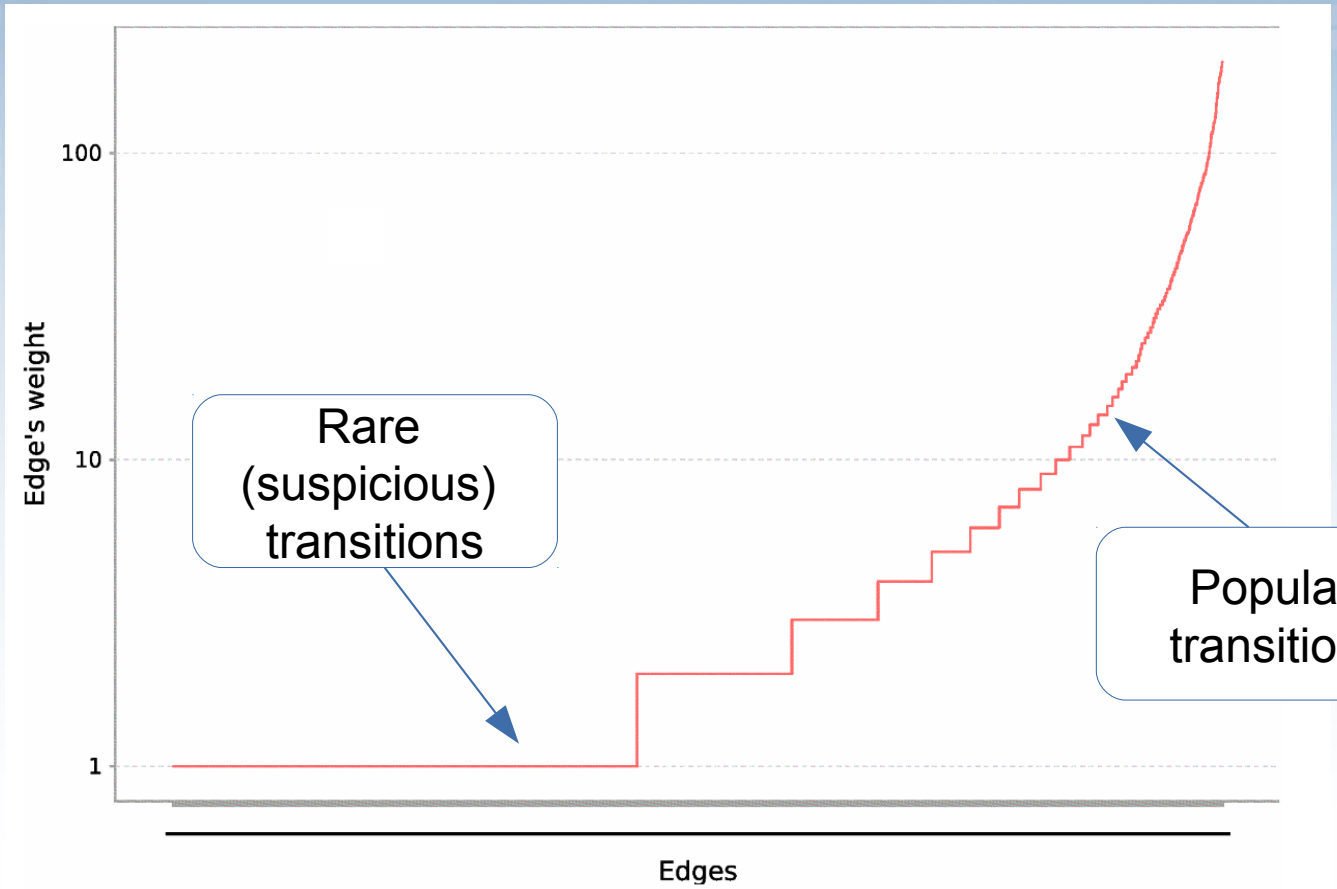


Another sample transition graph



OWASP AppSecEU 15
Amsterdam, The Netherlands

Distribution of edges' weights



Benefits of the graph structure

- Effective representations of large log data sets
- On-line analysis and detection of (some) suspicious behavior
- Identification of unusual requests, marking of potentially harmful requests
- Also, optimization of site's performance:
  - Pre-caching of expected web pages
  - Path-to-target minimization

Problems & issues

- Assembly of sessions from the requests
  - SID is the most reliable, but not always available
- Filtering out of follow-up requests (CSS, images)
- 'Unusual' but legitimate transitions
  - Introduction of 'trust earned'

Part 2

Collective assessment of suspicious requests

Rationale

- Lot of attack attempts are more or less automated. They are performed by malware or 'script-kiddies'
- As a result, (near) identical requests appear at different servers
- If the servers share the information about such requests, they can identify attack attempts with greater accuracy

Securing the confidentiality

- The requests (URLs) cannot be made publicly available, as this could lead to a leak of sensitive information
- However, it is sufficient to publish and share a cryptographic hash (e.g. MD5) of the request URL
- URL needs to be stripped of server specific part (domain, etc.), and likely request parameters

Sample data exchange format

```
{ C:O, T:M, A:57, MD5:2cf1d3c7fe2eadb66fb2ba6ad5864326 }
{ C:O, T:M, A:53, MD5:2370f28edae0afcd8d3b8ce1d671a8ac }
{ C:F, T:M, A:32, MD5:2f42d9e09e724f40cdf28094d7beae0a }
{ C:F, T:M, A:31, MD5:8f86175acde590bf811541173125de71 }
{ C:F, T:M, A:24, MD5:eee5cd6e33d7d3deaf52cadeb590e642 }
{ C:O, T:B, A:17, MD5:bd9cdbfedca98427c80a41766f5a3783 }
```

Additional algorithm-tuning information

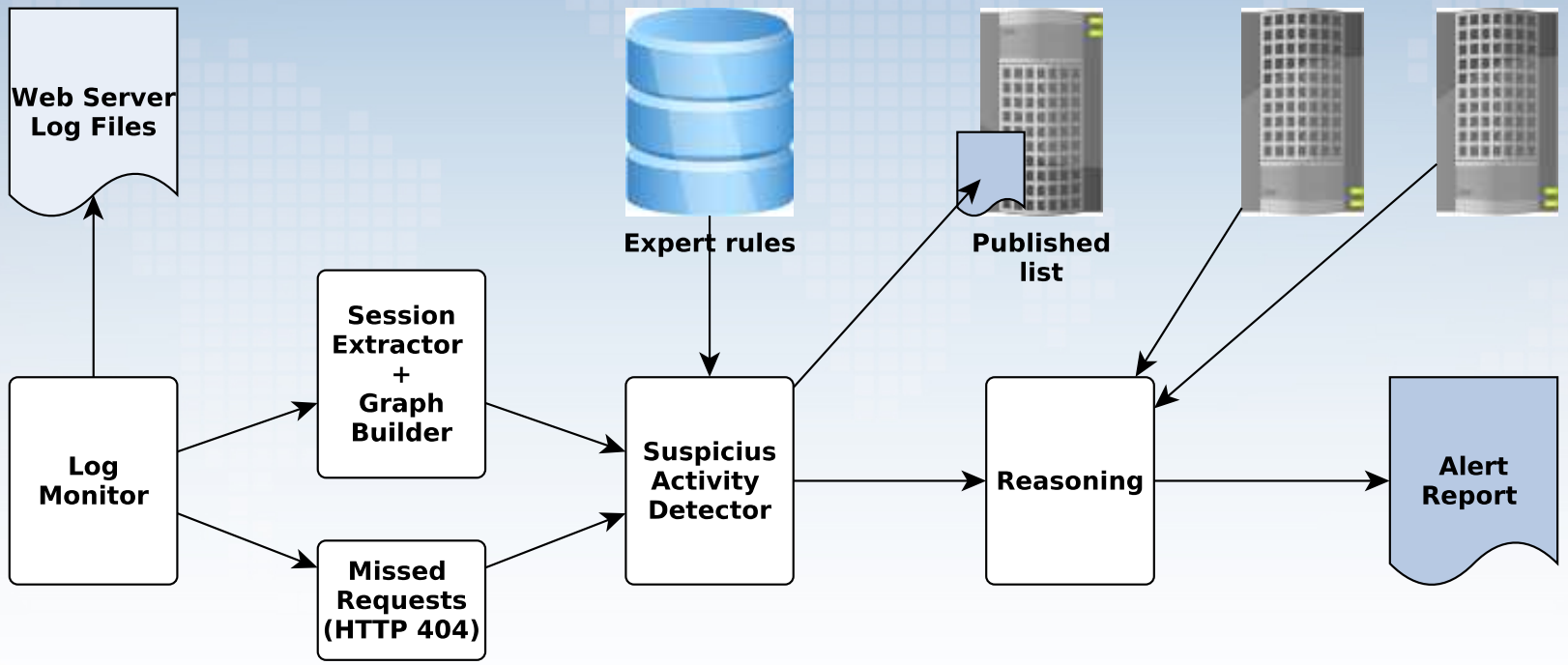
Age of the event

Type of report

M: missing resource

B: Behavioral anomaly

Reference implementation



Initial results

- Tests ran for a year on three (small) servers
- Approximately 30% of the attacks detected (compared to semi-manual log analysis)
- The ratio is expected to raise with the number of servers involved
- The results required no a-priori training
- Most importantly: **no false positives**

Thank you for your attention

Any questions are welcome

OWASP AppSecEU 15
Amsterdam, The Netherlands