McPAD & HMM-Web
two different approaches
for the detection of attacks
against Web Applications

Davide Ariu, Igino Corona, Giorgio Giacinto
Dept. of Electrical and Electronic Engineering
University of Cagliari, Italy
Outline

• Web services
• Motivations and purposes of our work
• Top Ten Web application attacks
• HMM-Web: an application-specific IDS
• McPAD: a payload-based IDS
Web Services - basics

Messages follow the HyperText Transfer Protocol (HTTP) and may flow inside an encrypted channel (e.g. SSL)

GET /search.php?attr1=val1&attr2=val2 HTTP/1.1
Host: www.example.com
Connection: keep-alive

HTTP/1.1 200 OK
Date: Thu, 26 Jun 2008 16:37:55 GMT
Connection: close
Content-Type: text/html

web application output (i.e. html page)
Top Ten¹ Web Application Attacks

1. Cross Site Scripting (XSS). Allows attackers to execute scripts in the victim’s browser.

2. Injection Flaws. Exploits an input validation flaw to make a server executing unintended commands or changing data.

3. Malicious File Execution. Allows attackers to include hostile code and data.

¹Source: Open Web Application Security Project, OWASP Top Ten 2007
Motivations and Purposes

• Today, so many services are offered through the web: e-commerce, flight booking, home banking …NOT only simple web sites…

• The security of (the information carried by) these services is a challenge, because:
  – The exposure to attacks is very high, because services are public, and the importance of information is often strategic this means high RISK
  – The threats, that is, the possible ways an attacker may follow, are multiple and with low cost for the adversary
  – security VULNERABILITY = RISK*THREAT

• Problem formulation:
  – Detect malicious HTTP requests (i.e. attacks known or unknown) to be able to take suitable counteractions
HTTP Guardian
Web Intrusion Protection System

Decision Module
(9) Flags situation
(9) Probability of normality

Probabilities Correlation
(8) Probabilities

Flags Correlation
(8) All Error Flags

Correlation Module
For each model, a probability threshold and its reliability are taken into account

Web IDS CONTROLLER
(7) Probabilities & Analysis Error Flags
(6) Analyse Features
(5) Features & Extraction Error Flags
(2) Get features of current HTTP request
(11) Store HTTP Request (and server response)

Apache
(10) Action
Encrypted Channel

Mod_security
(10) Action
Encrypted Channel

Mysql
(3) Query
(4) Data

Feature Extraction

Analysis Module
Model 1
Model N

HMM-Web, McPAD
HMM-WEB
HMM-Web overview

What is HMM-Web?

• An Intrusion Detection System capable to detect both simple and sophisticated input validation attacks against installed web applications.

• It exploits a sample of Web application queries to model normal (i.e. legitimate) queries.

• Attacks can be detected evidencing anomalous (not normal) web application queries.
HMM-Web overview

- HMM-Web is made up of a set of application-specific modules
  - Each module is made up of an ensemble of Hidden Markov Models, trained on a specific web application
  - During the detection phase, each web application query is analysed by the corresponding module

- A decision module classifies each analysed query as suspicious or legitimate
  - A different threshold is set for each application-specific module based on
    - the confidence on the legitimacy of the set of training queries
    - the proportion of training queries on the corresponding web application
HMM-Web
operational phase example
HMM-Web
application-specific modules

Each application-specific module analyses through a HMM ensemble:
- The input of each attribute, as a sequence of symbols, by generalising letters and numbers
- The attribute sequence, e.g. \{cat, key\}

Outputs of independent HMM ensembles are fused using the minimum rule to produce the query probability
HMM-Web
application-specific modules

- The decision module applies a threshold that is set independently for each web application

- This threshold depends on the confidence on the legitimacy of training queries, and on the proportion of queries for each web application
Datasets used for performance evaluation

- The training dataset (D) is extracted from a production web server of our institution.
- We estimated a proportion of attack queries around 0.1% in D.
- We labelled this dataset to evaluate the effectiveness of HMM-Web in front of attacks inside the training set.

- Attacks (dataset A) built using known attacks and typical vulnerabilities

### Table 1

<table>
<thead>
<tr>
<th>Data set D</th>
<th>Attack set A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries Time</td>
<td>Admin Pub</td>
</tr>
<tr>
<td>154,036</td>
<td>183 days</td>
</tr>
</tbody>
</table>

Principal characteristics of datasets D and A. For dataset D are shown: number of queries, collection time interval, number of web applications for administration (Admin) and public (Pub) services, total percentage of attacks $\alpha^* = \sum_{i=1}^{M} \alpha_i^*$. For dataset A are shown: number W of violated web applications, number of SQL Injection and Cross-Site Scripting (XSS) attacks.

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Exploit N.</th>
<th>Paper N.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Injection</td>
<td>6512, 6510, 6502, 6490, 6469, 6467, 6465, 6449, 6336, 3490, 3507</td>
<td>16, 174, 202, 215</td>
</tr>
<tr>
<td>XSS</td>
<td>2776, 2881, 2987, 3405, 3490, 4681, 4989, 6332</td>
<td>162, 173, 192</td>
</tr>
</tbody>
</table>

**TABLE 1**

References for attacks inside A, see http://www.milw0rm.com.
HMM-Web features and performances

- In the experiments HMM-Web detected all performed attacks (set A) with very low false positive rates (FPR < 0.4%), and it has been able to spot the 96% of attacks similar to those “erroneously” present in the training set, with a FPR < 1%.
- With respect to currently proposed systems, HMM-web is able to effectively perform an unsupervised training dealing with attacks inside the training set... This is very important, because it is really difficult to obtain an attack-free training set!
- The proposed query codification and modelling allows for a very fast training and a reliable detection of anomalous (i.e. attack) queries
- Future work will involve the development of automatic processes to also throw off possible attacks from the training set
Mc-PAD
McPAD
Multiple Classifiers Payload-based Anomaly Detector

- **McPAD** is a network-based IDS that analyses the payload of HTTP packets.
- It is based on the Multiple Classifier Systems paradigm, in order to attain both high Detection Rate and False Positive Rate.
- It is written in Java and it is freely available from our website.
- It has been developed by Davide Ariu (PRA Group) and Roberto Perdisci (GTISC).
McPAD
How does it work?

• McPAD applies a 2-v–analysis to extract statistics from the payload (i.e. frequencies of pairs of bytes)
• The 2-v–analysis is an “improvement” of the n-gram analysis aimed at dealing with the curse of dimensionality problem.
• While the n-gram analysis takes into account sequences of consecutive bytes, the 2-v–analysis considers just pairs of bytes ν positions away from each other
2ν-gram analysis a simple example

Figure 1: ν-gram with ν=0

Figure 2: ν-gram with ν=1

Figure 3: ν-gram with ν=2
**McPAD**

**How does it work?**

- Having a window that slides over the payload and considering a range of possible values of ν (from 0 to n)
  - A classifier is trained to model frequencies of pairs of bytes ν positions away from each other.
  - Combining classifiers built on different values of ν, it is possible to “approximate” a window of width n.
- An n-gram analysis is infeasible for n > 2 (the dimensionality of the features’ space is 256^n)
- Even if we increase the value of ν, our IDS always works in a features space of size 256^2
  - We also developed techniques to further reduce the number of features
McPAD: Datasets

**Georgia Tech Dataset**

<table>
<thead>
<tr>
<th>Day</th>
<th>Training</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>307.929</td>
<td>76.654</td>
<td>350.849</td>
</tr>
<tr>
<td>2</td>
<td>171.750</td>
<td>43.418</td>
<td>385.247</td>
</tr>
<tr>
<td>3</td>
<td>289.649</td>
<td>72.320</td>
<td>354.637</td>
</tr>
<tr>
<td>4</td>
<td>263.498</td>
<td>65.260</td>
<td>361.189</td>
</tr>
<tr>
<td>5</td>
<td>195.192</td>
<td>48.653</td>
<td>379.610</td>
</tr>
<tr>
<td>6</td>
<td>184.572</td>
<td>45.949</td>
<td>380.895</td>
</tr>
<tr>
<td>7</td>
<td>296.425</td>
<td>74.218</td>
<td>352.119</td>
</tr>
</tbody>
</table>

**DARPA Dataset**

<table>
<thead>
<tr>
<th>Day</th>
<th>Training</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>161.202</td>
<td>40.057</td>
<td>137.997</td>
</tr>
<tr>
<td>2</td>
<td>196.605</td>
<td>48.905</td>
<td>131.738</td>
</tr>
<tr>
<td>3</td>
<td>189.362</td>
<td>46.957</td>
<td>133.133</td>
</tr>
<tr>
<td>4</td>
<td>268.250</td>
<td>67.593</td>
<td>121.999</td>
</tr>
<tr>
<td>5</td>
<td>150.847</td>
<td>37.639</td>
<td>139.869</td>
</tr>
</tbody>
</table>

**Attack Dataset**

- For each dataset the number of packets that it contains is reported.
- Georgia Tech and DARPA dataset contain normal traffic.
- For the Attack dataset the number of different attacks of each type is reported.

<table>
<thead>
<tr>
<th>Type</th>
<th>Attack</th>
<th>Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic</td>
<td>66</td>
<td>205</td>
</tr>
<tr>
<td>Shellcode</td>
<td>11</td>
<td>93</td>
</tr>
<tr>
<td>Clet</td>
<td>96</td>
<td>792</td>
</tr>
<tr>
<td>PBA</td>
<td>6.339</td>
<td>71.449</td>
</tr>
<tr>
<td>Total</td>
<td>6.512</td>
<td>72.539</td>
</tr>
</tbody>
</table>
**McPAD**

**Experimental Setup**

- For each value of \( \nu = 0, \ldots, 10 \) a One-Class SVM is trained.
- A feature clustering algorithm has been applied in order to reduce the size of the features’ space.
  - Five different values for the number of clusters (10, 20, 40, 80, 160)
- SVM outputs have been combined using different fusion rules: Majority Voting, Maximum, Minimum, Average and Product of Probabilities.
- Cross validation to evaluate performances (DARPA dataset and Georgia Tech dataset).
A comparison:
PAYL performances

![Graph showing the comparison of PAYL performances with different models and parameters.](image)
A comparison:
McPAD performances
McPAD

Results Resume

• It performs better than previous solutions in terms of false positive rate (fundamental in order to obtain a high Bayesian Detection Rate)

\[ P(I \mid A) = \frac{P(A \mid I)P(I)}{P(A \mid I)P(I) + P(A \mid \bar{I})P(\bar{I})} \]

• High detection rate against shellcode and CLET-Based Attacks

• Detects Polymorphic Blending Attacks if the attack is not spread over a large number of packets.