Host based anomaly detection for webservers

RP1
Sudesh Jethoe
Overview

1. Introduction
2. Problem description
3. Research Questions & Method
4. Analyze
5. Solutions
6. Result
7. Conclusion
Introduction

HSBC has restored its online banking services after a distributed denial of service (DDoS) attack.

HSBC said servers had come under a DDoS attack which affected HSBC websites around the world.

The DDoS attack on HSBC did not affect any customer data, but did prevent customers using HSBC online services, including internet banking.

“We are cooperating with the relevant authorities and will cooperate with other organisations that have been similarly affected by such criminal acts.” HSBC said.
Byte Internet

- Since 1999
- Managed hosting
  - Shared hosting
- 10,000+ sites
Overview

1. Introduction
2. Problem description
3. Research Questions & Method
4. Analyze
5. Solutions
6. Result
7. Conclusion
Problem description

Facts:

● Sites get hacked
● Sites get abused
  ○ spam
  ○ malware distribution
  ○ (d)dos
Cause?

old versions of:
• frameworks
• plugins
weak passwords
What can customers do

- Update web application frameworks
  - Joomla, Wordpress
- Avoid buggy plugins
  - guestbook, photoalbum
- Use encrypted channels for data-transport
  ssh vs ftp
Why customers do not:

Dependency on customers

- Unaware
- Don't know how
- Don't want to risk it
- Unable/unwilling to pay for security measures
Overview

1. Introduction
2. Problem description
3. Research Questions & Method
4. Analyze
5. Solutions
6. Result
7. Conclusion
Research Questions

Can we develop a method which detects interactive malware (for example a webshell) running on servers in a shared hosting environment?

○ What are the characteristics of this kind of malware?
○ How can the characteristics be used to detect this malware?
○ How do existing solutions detect this malware?
○ Can we make use of existing frameworks for the detection and prevention in a hosting providers environment?
Method

- Collect malware
- Run it in a controlled environment
- Collect logs
- Review existing solutions
- Integrate method in a suitable solution
Overview

1. Introduction
2. Problem description
3. Research Questions & Method
4. Analyze
5. Solutions
6. Result
7. Conclusion
Cases (1/3) johanstegels.nl & webcast.nl

<form method="POST" action="{$fstring}&amp;action=save&amp;chdir={$chdir}&amp;file={$file}">

randomstream.nl 188.142.*.* - -
&action=cmd&amp;chdir=/home/users/randrftp/randomstream.nl/ HTTP/1.1" 200 3835 "http://randomstream.nl/webshell.php?http://www.education.zp.ua/images/down.jpg?&action=cmd&amp;chdir=/home/users/randrftp/randomstream.nl/" "Mozilla/5.0 (X11; Linux x86_64; rv:16.0) Gecko/20100101 Firefox/16.0"
Cases (2/3) florian.nl

```php
indx.php

switch($_POST['action'])
{
    case "upload":UploadFile($_FILES['file']);
    break;
    case "stop":stoped();
    break;
    **snip**
}
```

46.21.*.* web10.c4 www.florian.nl - - [18/Oct/2012:14:34:19 +0200] "POST /shop//langs/nl/indx.php HTTP/1.1" 200 - "-" "-" "-" "-" 46.21.145.228 florian.nl pid:31699 1608779 0 0 32002 36002
Klachten of onenigheid kunnen beter direct met de desbetreffende personen gecommuniceerd worden, zie voor de telnurnmers bij Organisatie of Contact.

Onderwerp *
Naam *
Email *

Inhoud *

Path: p

Beeld verificatie *

Type the two words:
Cases (3/3) liverunning.nl

199.15.*.* web8.c2 liverunning.nl - - [18/Oct/2012:12:05:39 +0200] "POST /index.php?option=com_phocaguestbook&view=phocaguestbook&id=2&Itemid=248 HTTP/1.0" 200 25805 "Mozilla/5.0 (Windows NT 5.1; rv:9.0.1) Gecko/20100101 Firefox/9.0.1" "-" "-" 199.15.*.* liverunning.nl
Analyze

1. Hacker abuses exploit
2. Hacker uploads malicious script
3. Hacker instructs script
   a. POST is used
      i. no character limit
      ii. content not shown in log
4. Malicious script is executed
## Detect?

### POST analysis

**7 sites, 7 days**

<table>
<thead>
<tr>
<th>Site</th>
<th>urls POSTed to</th>
<th>real files POSTed to</th>
</tr>
</thead>
<tbody>
<tr>
<td>sc*****</td>
<td>451</td>
<td>13</td>
</tr>
<tr>
<td>it*****</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>fa*****</td>
<td>198</td>
<td>12</td>
</tr>
<tr>
<td>de*****</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>dm*****</td>
<td>410</td>
<td>0</td>
</tr>
<tr>
<td>aa*****</td>
<td>344</td>
<td>1</td>
</tr>
<tr>
<td>aa*****</td>
<td>130</td>
<td>2</td>
</tr>
</tbody>
</table>
Overview

1. Introduction
2. Problem description
3. Research Questions & Method
4. Analyze
5. Solutions
6. Result
7. Conclusion
Solutions (Hosting Provider)

- Network Intrusion Detection Systems (NIDS)
- Web Application Firewalls (WAF)
- Host Intrusion Detection Systems
Byte Internals

Diagram:
- Visitor (WAN) connected to Loadbalancer
- Visitor 1 connected to Web1
- Visitor 2 connected to Web2
- Visitor 3 connected to Web3
- Web1, Web2, Web3 connected to NFS
- NFS connected to Storage
Solutions (Hosting Provider)

Network Intrusion Detection System
+ Can detect (and block) uploads in early stages
- Does not work on encrypted channels
- Depends on signatures (only detects known malware)

Web Application Firewalls
+ Can be finetuned to look for specific instructions
- Inspection takes time and slows visitor experience

Host Intrusion Detection Systems
+ Integrated tools for checking various system variables (files, logs)
- Not suitable for working over a LAN
Overview

1. Introduction
2. Problem description
3. Research Questions & Method
4. Analyze
5. Solutions
6. Result
7. Conclusion
Result

byte-security-POST-IDS
1. generate whitelist of files which can be posted to
2. tail access.log
3. grep POST
4. test files for:
   a. included in whitelist
      i. modifications
5. alert
Overview

1. Introduction
2. Problem description
3. Research Questions & Method
4. Analyze
5. Solutions
6. Result
7. Conclusion
Conclusion

- malicious scripts can be detected
- not suitable for attacks on indirect urls
Future work

- Tweak whitelist flagging
  - Who maintains the whitelist?
    - Site maintainers
    - The hosting provider
    - An algorithm?

- Read rewrite rules to find more files
  - For example by enabling mod_rewrite logging in Apache