Custom Intrusion Detection Techniques for Monitoring Web Applications

Matthew Olney
Sourcefire VRT
molney@sourcefire.com

AppSec DC
November 13, 2009

The OWASP Foundation
http://www.owasp.org
GIVE YOUR ANALYST A CHANCE
As we know, there are known knowns. There are things we know we know. We also know there are known unknowns. That is to say we know there are some things we do not know. But there are also unknown unknowns, the ones we don't know we don't know.

-- Donald Rumsfeld, Feb 12, 2002
The **attacker** holds a major information advantage

...but that makes the small advantages we do have that much more important.
Part One:  
Signature Based Detection
alert tcp $EXTERNAL_NET any -> $HOME_NET any 
(msg:”Jesus Christ it’s a lion GET IN THE CAR”; 
content:”LION”; nocase; classtype: attempted-bite-your-head; 
sid: 1;)

Jesus Christ it’s a lion

GET IN THE CAR
But the VRT doesn’t know…

your network,

your systems,

your applications

Maybe, just maybe, you do.
SNORT – The Open Source Intrusion Detection System

Signature based detection

Frag & Stream Reassembly

Substantial HTTP preprocessing

Multiple protocol decoding

http://www.snort.org
Matt’s 30 Second OWASP Rule Writing Class

Part 1:
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:”LOCAL #{Message}”; flow: to_server, established;

Part 2:
uricontent:”#{SecretSauce}”; nocase;

Part 3:
class-type: attempted-admin; sid: #{UniqueLocalSID};)

For Example:

alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:”LOCAL Admin page access attempt”; flow: to_server, established; uricontent:”admin”; nocase; class-type:attempted-admin; sid: 42098729;)

### Hypertext Transfer Protocol

- POST /cgi/comments.pl HTTP/1.1
  - [Expert Info (Chat/Sequence): POST /cgi/comments.pl HTTP/1.1]
    - [Message: POST /cgi/comments.pl HTTP/1.1]
      - [Severity level: Chat]
      - [Group: Sequence]
    - Request Method: POST
    - Request URI: /cgi/comments.pl
    - Request Version: HTTP/1.1
    - [truncated] Accept: image/gif, image/jpeg, image/png, image/png, ap
    - Referer: http://www.fark.com/cgi/comments.pl
    - Accept-Language: en-us
    - User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/)
    - Content-Type: application/x-www-form-urlencoded
  - Accept-Encoding: gzip, deflate
  - Host: www.fark.com
  - Content-Length: 344
    - [Content length: 344]
    - Connection: Keep-Alive
    - Cache-Control: no-cache
    - [truncated] Cookie: __qca=1231175793-89433651-15190562; __utma=20092591
  - Line-based text data: application/x-www-form-urlencoded
    - [truncated] tok=H39q5Kp_rPBU3s7-hi12Xk8_j_XCKdnDBvT2AyitM_O3tJkq7aTND2p

<table>
<thead>
<tr>
<th>0250</th>
<th>0260</th>
<th>0270</th>
<th>0280</th>
<th>0290</th>
<th>02a0</th>
<th>02b0</th>
<th>02c0</th>
<th>02d0</th>
</tr>
</thead>
<tbody>
<tr>
<td>6e 63 6f 64 65 64 0d 0a</td>
<td>41 63 63 65 70 74 2d 49</td>
<td>48 6f 73 74 3a 20 77 77</td>
<td>77 2e 6e 61 72 65 2e 63</td>
<td>65 6e 74 2d 4c 69 6e 67</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>70 2d 41 6c 69 6e 67</td>
<td>6f 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
</tr>
<tr>
<td>65 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>65 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>65 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>65 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>65 6e 66 69 63 6f 64 0d</td>
</tr>
<tr>
<td>77 77 77 77</td>
<td>77 77 77 77</td>
<td>77 77 77 77</td>
<td>77 77 77 77</td>
<td>77 77 77 77</td>
<td>77 77 77 77</td>
<td>77 77 77 77</td>
<td>77 77 77 77</td>
<td>77 77 77 77</td>
</tr>
<tr>
<td>65 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>65 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>65 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>65 6e 66 69 63 6f 64 0d</td>
<td>0a 43 6f 6e 66 69 63 6f</td>
<td>65 6e 66 69 63 6f 64 0d</td>
</tr>
</tbody>
</table>
i.e. (or is it e.g.)?
HTTP_INSPECT

- If you’re going to inspect http, it should be on.
- Already does some anomaly detection:
  - Directory traversal
  - Double decoding
  - Oversize URI Requests
  - Oversize chunk encoding
  - Oversize header size
- Check out the http_inspect config
- Read the README.http_inspect document (seriously)
- Remember to configure it to monitor any ports that handle HTTP traffic (80, 8080, custom web management consoles, etc…)
TEST YOUR RULES
(and find some more samples at http://vrt-sourcefire.blogspot.com/)
Where to find Snort help:

Snort-Users mailing list:
https://lists.sourceforge.net/lists/listinfo/snort-users

Snort-Sigs mailing list:
https://lists.sourceforge.net/lists/listinfo/snort-sigs

Webcasts (Writing Effective Rules Parts I and II):
http://www.sourcefire.com/resources/snort-webcast-access

IRC:
#snort on freenode

VRT Blog:
http://vrt-sourcefire.blogspot.com/

Twitter:
http://twitter.com/vrt_sourcefire
Your code and application flow defines how the client should request information.

Abuse that fact to “give your analysts a chance” at finding a problem.
Part Two:
Anomaly Based Detection
Different demands an explanation
Netflow:
The Instant WTF Generation Specialist
Analysis of Network Flow Statistics

For each network conversation we get the following data:

- Source and destination IP address and ports
- Total number of packets
- Total number of layer 4 bytes
- IP protocol number

Individually the information is mildly interesting. But within the context of all other data at that moment, or all other data ever gathered, individual flows can become very interesting.
Typically, you’ll have to play nice with the router guys:
A simple top talker chart (by Total Octets):

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Total Flows</th>
<th>Total Octets</th>
<th>Total Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.4.12.226</td>
<td>68.177.102.20</td>
<td>7</td>
<td>1252024</td>
<td>10413</td>
</tr>
<tr>
<td>10.4.12.60</td>
<td>10.4.128.99</td>
<td>10136</td>
<td>624726</td>
<td>10136</td>
</tr>
<tr>
<td>10.4.12.226</td>
<td>216.34.181.48</td>
<td>5167</td>
<td>421623</td>
<td>5310</td>
</tr>
<tr>
<td>10.4.12.60</td>
<td>216.34.181.48</td>
<td>4891</td>
<td>397892</td>
<td>5024</td>
</tr>
<tr>
<td>10.4.12.201</td>
<td>68.177.102.20</td>
<td>4437</td>
<td>358588</td>
<td>4673</td>
</tr>
</tbody>
</table>
Tools Exists to Present the Data in Different Forms:

Image of FlowScan Output

Flow-Tools Suite
Flow-report output
Things to Look For:

• Service detection
• Statistical deviation from the norm
• Connections to unexpected networks
• Local Outliers (top Talker)
• External Outliers (Top Listener)
• Any condition that makes your brow furrow.

Bonus: Encrypted Traffic Doesn’t Affect Netflow
Different demands an explanation

Examining an unexplained outlier may be the “break” that gives you a chance to catch the bad guy.
Some Netflow Resources

Flow Tools (PERL folks go here)
http://www.splintered.net/sw/flow-tools/

Cisco’s Netflow Site

Collection of Netflow Analysis Tools
http://www.networkuptime.com/tools/netflow/

Caida’s Flowscan Netflow Visualization Tool
http://www.caida.org/tools/utilities/flowscan/
Questions?