Defending Computer Networks

Lecture 13: NIDS/HTTP

Stuart Staniford
Adjunct Professor of Computer Science
Logistics

• No lecture Tuesday
• Quiz 2 next time (Thursday 15\textsuperscript{th})
  – Networking through DDOS
• Midterm November 3\textsuperscript{rd}
• Guest lectures:
  – October 20\textsuperscript{th}, Cornell ITSO
  – Nov 5\textsuperscript{th}, Tim Dawson
  – One more pending from a security vendor
Revealed: The 100 car models at risk of being stolen due to security 'flaw'

A scientist from Birmingham has won a two-year legal battle to finally publish research that reveals a major security flaw that could leave scores of car models at risk of being stolen.

Volkswagen had used its lawyers to keep under wraps the research of University of Birmingham computer scientist Flavio Garcia and his colleagues Baris Ege and Roel Verdult from Radboud University Nijmegen in the Netherlands.

They discovered more than 100 models of cars produced by 26 car manufacturers are at risk of theft by hackers who could crack codes to produce fake keys - thanks to flaws in a device designed to prevent vehicles from being stolen.
Main Goals for Today

• More Network Intrusion Detection
• Start on HTTP
Overall Snort Architecture

- Packets captured by libpcap
- Packet decoder
- Pre-processors
- Detection engine
- Ruleset
- Alert database
- Alert generation
- Alerts
Snort Detection Engine Data Structure

Figure 3: Rule Chain logical structure.
Snort Rule Example 1

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"SERVER-WEBAPP HyperSeek hsx.cgi directory traversal attempt";
flow:to_server,established; content:"/hsx.cgi"; http_uri; content:"../../";
http_raw_uri; content:"%00"; distance:1; http_raw_uri; metadata:ruleset community, service http; reference:bugtraq,2314; reference:cve,2001-0253;
Variants

Clearly we have to reassemble TCP before looking for “SIGNATURE”
How Fragmentation Works

All fragments of a given packet have same id

Offset of this segment in the original ip data, in eight byte blocks

MF flag bit says whether to expect any more packets

Note that fragmentation can be used to break up the TCP header, not just the TCP data
Evading NIDS: Mac address

• Only works if on same L2 network as NIDS
• Add extra packets directed to bad Mac address
  – But with correct destination IP
  – If IDS is not careful, it will process promiscuously
    • Where end-client won’t
• Note there are possible legit reasons for Mac address to change during a connection
  – Eg route flapping
  – So just looking for a changing Mac will have some FPs.
### Evading NIDS: TTL

<table>
<thead>
<tr>
<th>Field</th>
<th>Bit Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>0 - 3</td>
</tr>
<tr>
<td>IHL</td>
<td>4 - 7</td>
</tr>
<tr>
<td>Type of Service</td>
<td>8 - 11</td>
</tr>
<tr>
<td>Total Length</td>
<td>12 - 15</td>
</tr>
<tr>
<td>Identification</td>
<td>16 - 19</td>
</tr>
<tr>
<td>Flags</td>
<td>20 - 23</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>24 - 27</td>
</tr>
<tr>
<td>Time to Live</td>
<td>28 - 31</td>
</tr>
<tr>
<td>Protocol</td>
<td>0 - 15</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>16 - 31</td>
</tr>
<tr>
<td>Source IP Address</td>
<td>0 - 31</td>
</tr>
<tr>
<td>Destination IP Address</td>
<td>0 - 31</td>
</tr>
<tr>
<td>Options</td>
<td>0 - 31</td>
</tr>
<tr>
<td>Padding</td>
<td>0 - 31</td>
</tr>
</tbody>
</table>
Evading NIDS: TTL Field
Fragmentation Variant Strategy

• Similar to TTL
• There is a DF bit in “Flags” field in IP header
• Means “Don’t Fragment”
• On certain packets, set this then set packet size greater than MTU at some part of route
• Routers will drop those packets, not deliver
• Can be used as an evasion strategy
IP Timestamp Option Evasion

- IP Options allow additional fields to be added to IP packet header
  - For special purposes
  - IHL field > 5 signals presence of options
- Timestamp recording (RFC 781)
- Packet will be dropped if timestamp option malformed
Effects of Evasions

• Force the IDS to know a great deal about the network
  – Distance to end points (TTL)
  – MTUs in physical networks (DF bit)
  – Nature of end-client (reassembly algorithms)

• OTOH
  – Many of these strategies are themselves somewhat suspicious
  – IDS can use them as evidence
    • maybe, care needed on FPs
Strategies for Defeating Evasions

• Target based
  – IDS needs to figure out nature of all machines on network
    • Active fingerprinting (integration with vuln scanner)
    • Passive fingerprinting
    • Manual, static
      – not scalable unless network pretty homogeneous
  – Do TCP, Frag, etc reassembly however appropriate
    • IDS implementors have a lot of work to do
Strategies for Evasions (2)

• Normalization
  – If IDS is inline (IPS = Intrusion Prevention System)
  – Then IPS can rewrite packet stream to make it unambiguous
  – Solves problem pretty well in principle
  – Places different set of demands on IPS
    • Better not break anything in rewriting those packets!
    • Latency
    • Reliability – MTF
      – Disks on box
    • Typically customers start in non-inline mode, and then move to inline as they gain confidence
At Experian, Security Attrition Amid Acquisitions

T-Mobile disclosed last week that some 15 million customers had their Social Security numbers and other personal data stolen thanks to a breach at Experian, the largest of the big American consumer credit bureaus. But this actually wasn’t the first time that a hacking incident at Experian exposed sensitive T-Mobile customer data, and that previous breach may hold important clues about what went wrong more recently.

HTTP 1.1

• Main protocol that web runs over
• By far most important application protocol on Internet
• RFC 2616 (1999)
  – Obsoletes RFC 1945 for HTTP 1.0
  – HTTP originally dates back to Tim Berners Lee/CERT in 1991 (v 0.9)
• Text based request/response protocol
  – Originally primarily to identify/download files
  – Also provides for web applications
Protocol Layering

• HTTP runs over TCP
• In HTTP 1.1, one TCP connection can have a series of HTTP requests
  – Reverse direction carries responses
• NB: connection structure is not that meaningful to HTTP
  – Different browser may use one or multiple TCP connections
    • And spread requests between them differently.
  – Proxies can rearrange requests/responses to different connections.
Protocol Layering Names/Numbers

- **Application**: HTTP (5-7)
- **Transport**: TCP (4)
- **Network**: IP (3)
- **Datalink**: Ethernet (2)
- **Physical**: Copper wire/fiber optic (1)
HTTP Request

GET /dumprequest HTTP/1.1\r\nHost: djce.org.uk\r\nConnection: keep-alive\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/30.0.1599.101 Safari/537.36\r\nDNT: 1\r\nReferer: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CD4QFjAC&url=http%3A%2F%2Fdjce.org.uk%2Fdumprequest&ei=835lUpjEM5Xb4APEglGoDA&usg=AFQjCNExeAn5wSZMp_y_oTmOKonq482sS9A&sig2=pSajtDK-YYIvE4HFDqmRfA&bvm=bv.54934254,d.dmg\r\nAccept-Language: en-US,en;q=0.8\r\n\r\nTry it at http://www.procato.com/my+headers/
HTTP Header Basics

• Text lines separated by \r\n
  – Servers often accept “\n” only, but protocol is “\r\n”

• Header is terminated by a blank line (\r\n\r\n)

• Initial request line
  – GET /dumprequest HTTP/1.1\r\n
    • Other methods include POST, CONNECT, HEAD, DELETE, etc.
    • Focus on GET for now

• Followed by headers of form
  – Header: Value...\r\n
  – No request headers are actually required
Let’s try it

• Telnet to [www.google.com](http://www.google.com) 80 and try a manually entered request for nosuchpage.html
A Few Popular Request Headers

• Host:
  – Used to specify domain (server might have several).

• User-Agent:
  – Gives browser specifics (allows server to customize responses to browser)

• Referer:
  – What page (etc) sent us here

• Accept-Language:
  – We speak English, or...

• Accept:
  – media formats we accept (eg text/html)
HTTP Response Basics

• Text lines separated by \r\n
• Header is terminated by a blank line (\r\n\r\n)

• Initial response line
  – HTTP/1.1 404 Not Found\r\n    • Indicates status of request.

• Followed by headers of form
  – Header: Value...\r\n
  – No response headers are actually required
    • Though hard to get much done without them...
Important Response Codes

• 200 OK
• 301 Moved Permanently
• 304 Not Modified
• 400 Bad Request
• 404 Not Found
• 500 Internal Server Error
A Few Popular Response Headers

• Content-Type:
  – Media-type of entity attached after header

• Content-Length:
  – Length of same (in bytes)

• Content-encoding:
  – ‘gzip’ means compression applied

• Date:

• Server: software being run on the server
Let’s try it

• telnet on 80 to a few popular websites
  – www.cnn.com
  – www.yahoo.com
  – www.nytimes.com
**Entity Body**

- **Follows header**
  - either request or response, but more consistently in response direction
  - Can be any media type:
    - text/html, text/plain, image/jpeg, audio/mpeg
    - [http://www.iana.org/assignments/media-types](http://www.iana.org/assignments/media-types)
  - Three methods to delineate length:
    - Content-length
    - Transfer-encoding: chunked
    - Connection: close
Detecting Attacks on Web Servers

• Has been a major industry for 15+ years
• Exploits on the servers themselves
• Exploits on cgi scripts,
  — other server-side plugins
• SQL Injection
• Cross-site scripting
• Also HTTP command-and-control
  — Similar issues of detecting bad HTTP requests
Top Snort Rule Files

Stuarts-MacBook-Pro:rules stuarts$ du -s -k *.rules |sort -n -r |head -10
6152   deleted.rules
1216   browser-plugins.rules
792    malware-cnc.rules
688    blacklist.rules
568    server-webapp.rules
392    file-identify.rules
348    file-office.rules
344    server-other.rules
328    pua-adware.rules
316    browser-ie.rules
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"MALWARE-CNC Win.Trojan.Zbot variant in.php outbound connection"; flow:to_server,established; urilen:7; content:"/in.php"; http_uri; content:".ru|0D 0A|User-Agent|3A 20|Mozilla/4.0|0D 0A|"; fast_pattern:only; http_header; content:"|0A|Content-Length|3A 20|"; http_header; metadata:policy balanced-ips drop, policy security-ips drop, ruleset community, service http; reference:url,zeustracker.abuse.ch/monitor.php?ipaddress=195.22.26.231; classtype:trojan-activity; sid:26023; rev:3;)

Snort Example 4

## 89861: D-Link Multiple Router command.php cmd Parameter Remote Command Execution

<table>
<thead>
<tr>
<th>Views This Week</th>
<th>Views All Time</th>
<th>Added to OSVDB</th>
<th>Last Modified</th>
<th>Modified (since 2008)</th>
<th>Percent Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>574</td>
<td>9 months ago</td>
<td>about 1 month ago</td>
<td>14 times</td>
<td>100%</td>
</tr>
</tbody>
</table>

### Timeline
- **Disclosure Date**: 2013-02-04
- **Time to Exploit**: 52 days
- **Time to Vendor Response**: 7 days

### Description
Multiple D-Link routers contain a flaw that is triggered when input passed via the 'cmd' parameter is not properly sanitized before being used in the command.php script. This may allow a remote attacker to execute arbitrary commands.

### Classification
- **Location**: Remote / Network Access
- **Attack Type**: Input Manipulation
- **Impact**: Loss of Integrity
- **Solution**: Solution Unknown
- **Exploit**: Exploit Public
- **Disclosure**: Vendor Disputed, Third-party Verified
- **OSVDB**: Web Related

### Solution
OSVDB is not currently aware of a solution for this vulnerability.

### Products

<table>
<thead>
<tr>
<th>Products</th>
<th>DIR-300</th>
<th>DIR-600</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-Link Corporation/D-Link Systems, Inc.</td>
<td>2.12b02</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td>2.13b01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.14b01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.13</td>
<td></td>
</tr>
</tbody>
</table>