

# Analyzing Potential False Positive Alerts In Snort

Alex Kirk

Sourcefire VRT



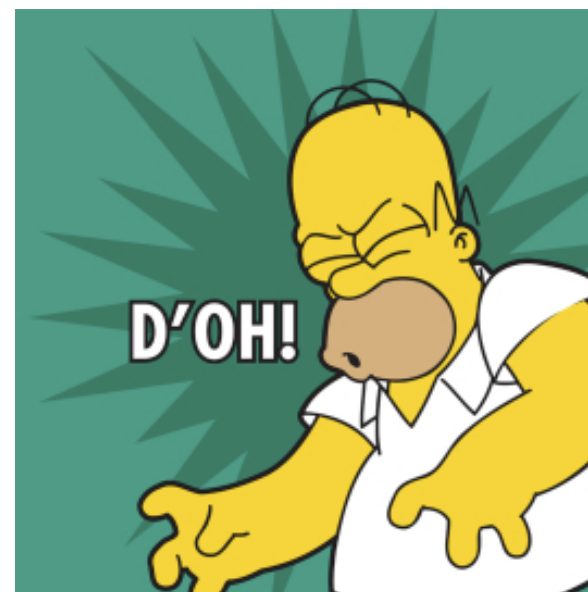
# About the Sourcefire VRT

- Founded in 2001
- 20 team members
  - Core team members based in Columbia, Maryland (USA)
  - ClamAV team members based in Poland, Italy and Germany
  - Threat Feed team members based in Columbia and India
- Responsibilities include:
  - Publishing new Snort rules and Sourcefire Protection Updates
  - Publishing new ClamAV signatures
  - Development of the ClamAV Engine



# False Positives Happen

- Attack could be coming into a non-vulnerable system
- Rule could be written too broadly
- Analyst could have misunderstood the vulnerability
- Malicious behavior might be indistinguishable from legitimate traffic



# Addressing False Positives

- Tuning Snort and your active rules is first step
  - Many people err on the side of running rules
  - Better to run a small set of quality rules
    - Less noise = more detection
- No matter how good your tuning, some questionable alerts will remain
- How the VRT analyzes them, through examples



# First, A Useful Tool – Session-Wrap

- Snort logs a single packet with alerts
- Not useful for testing
  - Can't run back through Snort and get an alert
- Session-wrap to the rescue:
  - `perl session-wrap <filename>`
  - Can specify `--client` or `--server`
  - Available soon on <http://labs.snort.org/>



# SID 13865 – Adobe BMP Image Handler Overflow

```
0030 77 18 50 10 ff ff 24 98 00 00 48 54 54 50 2f 31
0040 2e 31 20 32 30 30 20 4f 4b 0d 0a 44 61 74 65 3a
0050 20 54 68 75 2c 20 31 30 20 44 65 63 20 32 30 30
0060 39 20 31 37 3a 33 31 3a 35 31 20 47 4d 54 0d 0a
0070 53 65 72 76 65 72 3a 20 41 70 61 63 68 65 0d 0a
0080 4c 61 73 74 2d 4d 6f 64 69 66 69 65 64 3a 20 57
0090 65 64 2c 20 31 32 20 41 75 67 20 32 30 30 39 20
00a0 32 32 3a 35 32 3a 33 36 20 47 4d 54 0d 0a 45 54
00b0 61 67 3a 20 22 32 38 34 36 66 37 2d 34 39 64 2d
00c0 34 61 38 33 34 37 62 34 22 0d 0a 41 63 63 65 70
00d0 74 2d 52 61 6e 67 65 73 3a 20 62 79 74 65 73 0d
00e0 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a
00f0 20 31 31 38 31 0d 0a 4b 65 65 70 2d 41 6c 69 76
0100 65 3a 20 74 69 6d 65 6f 75 74 3d 31 2c 20 6d 61
0110 78 3d 31 32 32 0d 0a 43 6f 6e 6e 65 63 74 69 6f
0120 6e 3a 20 4b 65 65 70 2d 41 6c 69 76 65 0d 0a 43
0130 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 69 6d 61
0140 67 65 2f 70 6e 67 0d 0a 0d 0a 89 50 4e 47 0d 0a
0150 1a 0a 00 00 00 0d 49 48 44 52 00 00 00 19 00 00
0160 00 19 08 03 00 00 00 f3 37 75 51 00 00 00 03 73
0170 42 49 54 08 08 08 db e1 4f e0 00 00 02 01 50 4c
0180 54 45 ff ff ff 66 66 66 42 4d 66 43 43 43 00 00
0190 00 00 00 00 1b 1b 1b 00 00 00 a5 a5 a5 a1 a4 9c
01a0 99 99 99 95 95 95 00 00 00 89 89 86 43 43 43 99
01b0 99 99 1b 1b 1b 05 05 05 21 21 21 1b 1b 1b 21 21
01c0 21 10 10 10 00 00 00 29 29 29 c3 c1 bd b6 b6 b6
```

```
w.P...$. ..HTTP/1
.1 200 O K..Date:
Thu, 10 Dec 200
9 17:31: 51 GMT..
Server: Apache..
Last-Mod ified: w
ed, 12 A ug 2009
22:52:36 GMT..ET
ag: "284 6f7-49d-
4a8347b4 "..Accep
t-Ranges : bytes.
.Content -Length:
1181..K eep-Alive
e: timeo ut=1, ma
x=122..C onnectio
n: keep- Alive..C
ontent-T ype: ima
ge/png.. ...PNG..
.....IH DR.....
..... /uQ....S
BIT..... O.....PL
TE...fff BMFCCC..
.....
..... .CCC.
..... !!!...!!
!.....) ).....
```

# File Format Mismatch

- Rule is looking for malformed Bitmaps
- Alert is on a PNG file
- Common occurrence – multiple files downloaded over a single TCP session
  - Flowbit http.bmp is set
  - Not unset for new files

# Flowbit “Grouping” Feature

- alert tcp \$EXTERNAL\_NET \$HTTP\_PORTS -> \$HOME\_NET any (msg:“WEB-MISC bitmap file download request”;  
flow:established,to\_client; uricontent:“.bmp”;  
nocase; **flowbits:set,http.bmp,http.files;**  
flowbits:noalert; metadata: policy security-ips  
alert, service http; classtype:misc-activity; sid:  
16205)
- Available in Snort 2.8.6 Beta

# SID 7070 – Encoded XSS

- Rule is simple – looks for “<SCRIPT” in a URI
- Works reliably against a lot of basic XSS
- Customer said it was alerting every time they accessed Lotus Notes



# Wireshark Is Your Friend

The screenshot displays the Wireshark interface with a packet list on the left, a packet bytes pane in the middle, and a context menu open over the packet bytes pane. The packet list shows a single packet (Frame 22) with details for Ethernet II, Internet Protocol, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII. The context menu includes options like 'Expand Subtrees', 'Apply as Filter', 'Copy', and 'Decode As...'. The 'Decode As...' menu is also open, showing options like 'Description', 'Fieldname', 'Value', and 'Bytes (Hex Stream)'.

Frame 22 (1514 bytes on wire, 1500 bytes captured)

- Ethernet II, Src: Cisco\_1e:6c:32 (00:11:bb:1e:6c:32), Dst: crossbea\_11:5e:fe (00:03:d2:11:5e:fe)
- Internet Protocol, Src: 172.16.11.13 (172.16.11.13), Dst: 10.10.4.131 (10.10.4.131)
- Transmission Control Protocol, Src Port: 41229 (41229), Dst Port: http (80), Seq: 3927, Ack: 4905,
- Hypertext Transfer Protocol
- Data (1446 bytes)

Data: 2532352532354D6F64446174653026302235253235506673  
[Length: 1446]  
[Packet size limited during capture: 1446]

Offset	Hex	ASCII
0030	2b ba 26 36 00 00 25 32 35 25 32	
0040	61 74 65 3d 26 25 32 35 25 32 35	
0050	68 61 72 73 65 74 3d 49 53 4f 2d	
0060	31 26 68 5f 53 63 65 6e 65 43 6f	
0070	3d 70 75 74 41 77 61 79 25 35 42	
0080	62 6c 69 73 68 41 63 74 69 6f 6e	
0090	44 25 32 36 25 32 36 25 32 36 25	
00a0	25 32 36 70 75 74 41 77 61 79 25	
00b0	70 75 62 6c 69 73 68 46 6f 6c 64	
00c0	6c 65 25 32 37 25 35 44 25 32 36	
00d0	36 25 32 36 25 32 36 25 32 36 70	
00e0	79 25 35 42 25 32 37 4d 45 25 32	
00f0	32 36 25 32 36 25 32 36 25 32 36	
0100	36 70 75 74 41 77 61 79 25 35 42	
0110	62 6c 69 73 68 46 6f 6c 64 65 72	
0120	6e 69 64 25 32 37 25 35 44 25 32	
0130	32 36 25 32 36 25 32 36 25 32 36	
0140	61 79 25 35 42 25 32 37 74 6f 63	
0150	69 6f 6e 25 32 37 25 35 44 25 32	
0160	32 36 25 32 36 25 32 36 25 32 36	
0170	61 79 25 35 42 25 32 37 74 6d 70	
0180	32 37 25 35 44 25 32 36 25 32 36	
0190	36 25 32 36 25 32 36 70 75 74 41	
01a0	42 25 32 37 73 65 6c 65 63 74 65	
01b0	65 72 49 6e 64 65 78 25 32 37 25 35	
01c0	25 32 36 25 32 36 30 25 32 36 25 32	
01d0	70 75 74 41 77 61 79 25 35 42 25 32	
01e0	25 32 37 25 35 44 25 32 36 25 32 36	
01f0	32 36 25 32 36 25 32 36 26 68 5f 45	
0200	63 74 69 6f 6e 3d 68 5f 4e 65 78 74	
0210	65 74 45 64 69 74 43 75 72 72 65 6e	
0220	6e 65 3d 73 5f 53 74 64 50 61 67 65	
0230	26 68 5f 53 65 74 50 75 62 6c 69 73	
0240	64 65 72 73 3d 26 68 5f 41 6c 74 65	
0250	65 4e 61 6d 65 3d 26 68 5f 43 75 72	
0260	46 6f 6c 64 65 72 44 6f 63 75 6d 65	
0270	68 5f 43 75 72 72 65 6e 74 46 6f 6c	
0280	61 6d 65 3d 26 68 5f 65 74 45 64 69	
0290	78 74 53 63 65 65 3d 26 68 5f 53 65	
02a0	74 75 72 6e 55 52 4c 3d 25 33 43 73	
02b0	74 25 33 45 70 61 72 65 6e 74 2e 73	
02c0	69 74 5f 4f 6e 4f 70 65 72 61 74	
02d0	6d 70 6c 65 74 65 25 32 38 25 32 39	

25ModD  
5PostC  
-8859-  
ontext  
8%27pu  
n%27%5  
%26%26  
%58%27  
4erT3+

Copy  
Export Selected Packet Bytes...  
Wiki Protocol Page  
Filter Field Reference  
Protocol Preferences  
Decode As...  
Disable Protocol...  
Resolve Name  
Go to Corresponding Packet

Description  
Fieldname  
Value  
As Filter  
Bytes (Offset Hex Text)  
Bytes (Offset Hex)  
Bytes (Printable Text Only)  
Bytes (Hex Stream)  
Bytes (Binary Stream)

Data (data.data), 1446 bytes | Packets: 33 Displayed: 33 Marked: 0

# Seriously, Lotus, WTF?

- Offending text:
  - `&h_SetReturnURL=%3Cscript%3Eparent.stdEdit_OnOperationComplete%28%29%3B%3C%2Fscript%3E`
  - `%3C` normalizes to “<”
- Examination of PCAP shows calling JavaScript is common practice in legit Lotus traffic



# Local Stupid = Local Fix

- Create a local flowbit rule
  - alert tcp \$EXTERNAL\_NET any -> \$HOME\_NET \$HTTP\_PORTS (msg:“LOCAL Lotus Notes webmail flowbit set”; flow:established,to\_server; content:“Host|3A| webmail.company.com”; nocase; **flowbits:set,lotus.webmail;** flowbits:noalert;)
- Make existing rule check that flowbit

# SID 12741 – Quicktime TCP RTSP sdp type buffer overflow

- QuickTime buffer overflow
- Looks like:

```
RTSP/1.0 200 OK
```

```
Content-Type:
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAA <shellcode goes here>
```

# Evasion Case

RTSP/1.0 200 OK

```
Content-Type      : AAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAA  
AAAAAAAAAAAAAA  
AAAAAAAAAAAAAA  
AAAAAAAAAAAAAA  
AAAAAAAAAAAAAA  
AAAAAAAAAAAAAA  
AAAAAAAAAAAAAA <shellcode goes  
here>
```

Spaces between “Content-Type” and “:”

# Existing Rule

```
alert tcp $EXTERNAL_NET 554 -> $HOME_NET
  any (msg:"EXPLOIT Apple Quicktime TCP RTSP
  sdp type buffer overflow attempt";
  flow:to_client,established; content:"RTSP";
  depth:4; content:"Content-Type"; nocase;
  content:"|3A|"; distance:0; isdataat:
  256,relative; content:!"|0A|"; within:256;
  content:!"|3A|"; within:256; metadata:service
  rtsp; reference:bugtraq,26549; reference:cve,
  2007-6166; classtype:attempted-user; sid:
  12741; rev:9;)
```

# Rule Problem

- Goal - Allow for spaces between “Content-Type” and “:”
- Actual result – any “:” following “Content-Type” in the packet matches

```
RTSP/1.0 200 OK
```

```
Content-Type: application/sdp
```

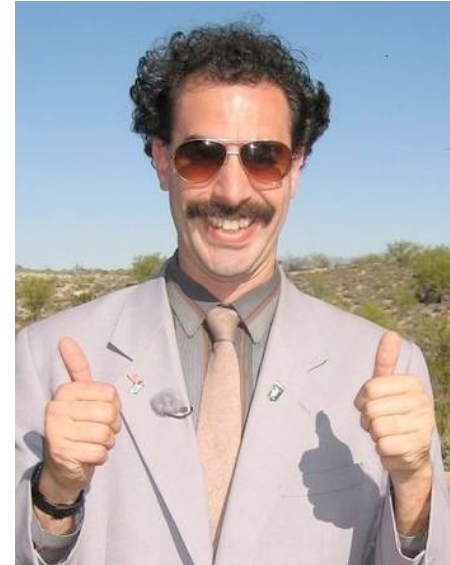
```
a=pgmpu:data :application/x-wms- contentdesc,  
8, language,  
31, 0, , 42, WMS_CONTENT_DESCRIPTION_PLAYLIST_ENTRY  
_URL,  
31, 1, /, 58, WMS_CONTENT_DESCRIPTION_COPIED_METADATA  
FROM_PLAYLIST_FILE,  
3, 1, 1, 47, WMS_CONTENT_DESCRIPTION_PLAYLIST_ENTRY  
_DURATION...
```

# Fixed Rule

```
alert tcp $EXTERNAL_NET 554 -> $HOME_NET any
(msg:"EXPLOIT Apple Quicktime TCP RTSP sdp
type buffer overflow attempt";
flow:to_client,established; content:"RTSP"; depth:
4; fast_pattern; content:"Content-Type"; nocase;
isdataat:257,relative; content:!"|0A|"; within:257;
pcre:"/Content-Type\s*\x3A[^\n\x3A]{256}/smi";
metadata:service rtsp; reference:bugtraq,26549;
reference:cve,2007-6166; classtype:attempted-
user; sid:12741; rev:10;)
```

# Improvements

- PCRE validates spaces – more accurate
- Uses new “fast\_pattern” keyword
  - “Content-Type” is common
  - “RTSP” is not
  - Matching less common content means other rule options get evaluated less, so rule is faster



# SID 15912 – Small TCP Window

- Yes, it's a restricted SO rule
- You still have some information available
  - Msg: "small or zero-sized tcp window"
  - MS advisory: "excessive number of established TCP connections"
  - That's a DDoS



# TCP Flags

Protocol	Info
TCP	44674 > https [RST] Seq=1 win=0 Len=0

- Sure doesn't look like an ESTABLISHED connection to me
- Reported to the VRT; subsequently fixed



# SID 14896 – Conficker Coverage

stub data (680 bytes)																	
0090	00	03	10	00	00	00	00	02	00	00	01	00	00	a8	02	.....	.....
00a0	00	00	00	00	1f	00	2c	15	be	00	06	00	00	00	00	.....	.....
00b0	00	00	06	00	00	00	48	00	48	00	44	00	48	00	48	.....H.	H. D. H. H.
00c0	00	00	31	01	00	00	00	00	00	00	31	01	00	00	5c	..1.....	..1....\.
00d0	69	76	4e	4a	50	76	44	46	48	75	79	64	4d	50	4e	ivNJPvDF	HuydMPNl
00e0	70	61	5a	68	65	66	5a	7a	57	59	54	51	47	4a	72	paZhefZz	WYTQJrx
00f0	49	70	50	7a	74	72	4f	58	59	77	4b	68	68	53	43	IpPztrOX	YwKhHSCI
0100	4d	6f	4c	52	78	42	54	43	75	75	4e	6b	43	61	72	MoLRxBTC	uunkCary
0110	66	53	67	6a	6d	70	4a	71	42	4a	42	63	75	48	58	fSgjmpJq	BJBcuHXf
0120	55	75	7a	70	64	42	52	48	66	68	53	6e	53	44	73	UuzpdBRH	fhsnsDsz
0130	77	54	4d	7a	e8	ff	ff	ff	ff	c1	5e	8d	4e	10	80	WTMz....	..^..N..1
0140	c4	41	66	81	39	45	50	75	f5	ae	c6	9d	a0	4f	85	.Af.9EPu	.....O..
0150	4f	84	c8	4f	84	d8	4f	c4	4f	9c	cc	49	72	58	c4	O..O..O.	O..Irx..
0160	c4	2c	ed	c4	c4	c4	94	26	3c	4f	38	92	3b	d3	57	.....&	<08.;.WG
0170	02	c3	2c	dc	c4	c4	c4	f7	16	96	96	4f	08	a2	03	.....	...O....
0180	bc	ea	95	3b	b3	c0	96	96	95	92	96	3b	f3	3b	24	.....;	.....;.\$i
0190	95	92	51	4f	8f	f8	4f	88	cf	bc	c7	0f	f7	32	49	..QO..O.	.....2I.
01a0	77	c7	95	e4	4f	d6	c7	17	f7	04	05	04	c3	f6	c6	w...O...	.....
01b0	44	fe	c4	b1	31	ff	01	b0	c2	82	ff	b5	dc	b6	1b	D...1...	.....0
01c0	95	e0	c7	17	cb	73	d0	b6	4f	85	d8	c7	07	4f	c0	.....s..	O....O.T
01d0	c7	07	9a	9d	07	a4	66	4e	b2	e2	44	68	0c	b1	b6	.....fn	..Dh....
01e0	a9	ab	aa	c4	5d	e7	99	1d	ac	b0	b0	b4	fe	eb	eb	....]	.....
01f0	f4	ea	f5	f3	f6	ea	f4	ea	f5	f6	f4	fe	f2	fc	f5	.....	.....
0200	eb	b0	ae	b0	a2	ab	a9	b7	b7	c4	45	50	59	70	5a	.....	..EPYpZz
0210	4c	66	6a	45	4b	46	6b	63	45	4d	55	52	4c	64	53	LfjEKfKc	EMURldSH
0220	4a	78	52	78	75	41	6c	66	51	74	73	6b	57	5a	74	JxRxuAlf	QtskwZtw
0230	53	42	59	4e	62	52	66	42	4a	68	65	71	79	76	6e	SBYnBrFB	Jheqyvni
0240	42	54	52	65	43	74	78	53	4c	79	57	75	48	4d	5a	BTRectXS	LywuhMzX
0250	52	6e	72	5a	45	6c	74	55	74	76	54	6a	59	79	49	RnrZElTU	tvTjYyIn
0260	79	4b	46	66	49	4a	49	71	4f	67	69	66	7a	68	46	yKfFIJiQ	OgifzhFB
0270	54	4b	78	4d	58	59	4a	63	76	51	52	65	7a	6d	41	TKxMXyJc	vQRezmAx
0280	71	50	41	69	42	41	76	74	58	61	52	6b	4b	6c	4d	qPAiBAvt	XarkKlMH
0290	51	48	7a	51	64	72	50	76	78	51	77	68	58	4d	72	QHzQdrPv	xQwhXMrf
02a0	74	4b	58	6e	64	52	73	44	47	4d	78	49	62	62	66	tKXndRsD	GmXIbbfN
02b0	6b	67	78	58	47	73	56	64	4f	43	45	64	69	45	78	kgxXGsVd	OCediEXD
02c0	69	52	50	4c	5c	00	2e	00	2e	00	5c	00	2e	00	2e	iRPL\...	..\.....
02d0	5c	00	41	00	55	00	59	00	4c	00	4f	00	47	00	51	\.A.U.Y.	L.O.G.Q.
02e0	08	04	02	00	e2	16	5c	59	41	51	59	59	27	f7	5b	.....\Y	AQYY'. [Y
02f0	4c	55	51	52	51	4f	46	4d	4e	44	4c	55	55	58	46	LUQRQOFM	NdlUUXFM
0300	4d	5a	52	4f	55	47	58	51	55	59	4d	4c	46	4e	4a	MZROUGXQ	UYMLFNJG
0310	47	4d	51	51	43	4a	54	45	4d	54	92	4a	24	b6	97	GMQQCJTE	MT.J\$....
0320	f5	37	eb	62	45	49	4d	41	58	57	59	49	46	5a	00	.7.BEIMA	XWYIFZ..
0330	00	00	1f	03	00	00	02	00	00	00	00	00	00	00	02	.....	.....
0340	00	00	5c	00	00	00	01	01	00	00	00	00	00	00	00	..\.....	.....

# Looks Suspicious

- Data isn't Base64 encoded, doesn't look like standard MS SMB encoding
- Special characters, unprintable bytes, etc.
- Passed to Lurene Grenier for review
- **TRUE POSITIVE!**
  - Shellcode analysis at <http://vrt-sourcefire.blogspot.com/2008/12/ms08-067-in-wild.html>

# SID 16367 – CVE-2010-0249

- Yes, it's an SO rule
- Public exploits exist
  - Metasploit
  - Packetstorm
- All the data you need lives there
  - **Especially** if you compare the two



# Look At The Exploits: Metasploit

```
function
```

```
  zNIqTPCdKtymiCswAxZfRjrUTvdYrnwsy(evt){
```

```
NvOdcddDpxZEqrTgOGXRVVEzTyGFOPAhnuGqeyHh  
KHubknwfPhvCIVRybvUJHxXmgeiUGceuVVZHpRh  
YUy());
```

```
  UlegxolkleHlsIndYBGlcGysJvYGNFeOXnsOItwTDCy
```

```
  = document.createElement(evt);
```

```
  document.getElementById("SjzmlxGxjrsqoLmPPc  
xwmFihjChvjEpPog").innerHTML = "";
```

```
  window.setInterval(qNVUoIxBSKRVsuLhGuTKAciN  
OEcNKrynBJzmrrGu, 50);
```

```
}
```

# Look At The Exploits: Packetstorm

```
function ev1(evt)
{
    event_obj =
    document.createElement(evt);
    document.getElementById("sp1")
        .innerHTML = "";
    window.setInterval(ev2, 1);
}
```



# Questions?

Email: [alex.kirk@sourcefire.com](mailto:alex.kirk@sourcefire.com)

IRC: #snort on freenode

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

Mailing Lists: <https://lists.sourceforge.com/lists/listinfo/snort-users>

<https://lists.sourceforge.com/lists/listinfo/snort-sigs>

Twitter: [http://www.twitter.com/vrt\\_sourcefire](http://www.twitter.com/vrt_sourcefire)

