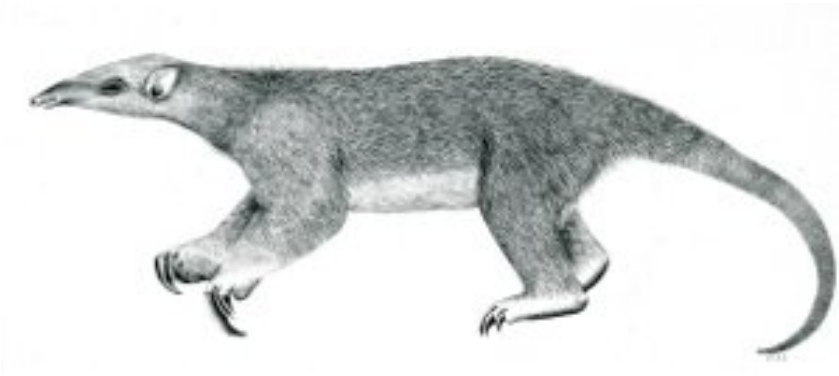
Tranalyzer2

httpSniffer

Hypertext Transfer Protocol (HTTP)



Tranalyzer Development Team

Contents

1 httpSniffer

The httpSniffer plugin processes HTTP header and content information of a flow. The idea is to identify certain HTTP features using flow parameters and to extract certain content such as text or images for further investigation. The httpSniffer plugin requires no dependencies and produces only output to the flow file. User defined compiler switches in *httpSniffer.h* produce optimized code for the specific application.

1.1 Configuration Flags

The flow based output and the extracted information can be controlled by switches and constants listed in the table below. They control the output of host, URL and method counts, names and cookies and the function of content storage. **WARNING:** The amount of being stored on disk can be substantial, make sure that the number of concurrent file handles is large enough, use `ulimit -n`.

Name	Default	Description	Flags
HTTP_MIME	1	mime types	
HTTP_STAT	1	status codes	
HTTP_MCNT	1	mime count: get, post	
HTTP_HOST	1	hosts	
HTTP_URL	1	URLs	
HTTP_COOKIE	1	cookies	
HTTP_IMAGE	1	image names	
HTTP_VIDEO	1	video names	
HTTP_AUDIO	1	audio names	
HTTP_MSG	1	message names	
HTTP_APPL	1	application names	
HTTP_TEXT	1	text names	
HTTP_PUNK	1	post/else/unknown names	
HTTP_BODY	1	analyse body and print anomalies	
HTTP_BDURL	1	refresh and set-cookie URLs	HTTP_BODY=1
HTTP_USRAG	1	user agents	
HTTP_XFRWD	1	X-Forward	
HTTP_REFRR	1	Referer	
HTTP_VIA	1	Via	
HTTP_LOC	0	Location	
HTTP_SERV	1	Server	
HTTP_PWR	1	Powered by	
HTTP_STATA	1	aggregate status response codes	
HTTP_HOSTAGA	1	aggregate hosts	
HTTP_URLAGA	1	aggregate URLs	
HTTP_USRAGA	1	aggregate user agents	
HTTP_XFRWDA	1	aggregate X-Forward-For	
HTTP_REFRRA	1	aggregate Referer	
HTTP_VIAA	1	aggregate Via	
HTTP_LOCA	1	aggregate Location	
HTTP_SERVA	1	aggregate Server	
HTTP_PWRA	1	aggregate Powered by	

Name	Default	Description	Flags
HTTP_SAVE_IMAGE	0	save all images	
HTTP_SAVE_VIDEO	0	save all videos	
HTTP_SAVE_AUDIO	0	save all audios	
HTTP_SAVE_MSG	0	save all messages	
HTTP_SAVE_TEXT	0	save all texts	
HTTP_SAVE_APPL	0	save all applications	
HTTP_SAVE_PUNK	0	save all else	
HTTP_RM_PICDIR	0	delete directories at T2 start	

Aggregate mode is on by default to save memory space. Note that HTTP_SAVE_* refers to the *Content-Type*, e.g., HTTP_SAVE_APPL, will save all payload whose Content-Type starts with application/ (including forms, such as application/x-www-form-urlencoded). The maximum memory allocation per item is defined by HTTP_DATA_C_MAX listed below. The path of each extracted http content can be set by the HTTP_XXXX_PATH constant. HTTP content having no name is assigned a default name defined by HTTP_NONAME_IMAGE. Each name is appended by the findex, packet number and an index to facilitate the mapping between flows and its content. The latter constant has to be chosen carefully because for each item: mime, cookie, image, etc, HTTP_MXFILE_LEN * HTTP_DATA_C_MAX * HASHCHAINTABLE_SIZE * HASHFACTOR bytes are allocated. The filenames are defined as follows:
Filename_Flow-Dir(0/1)_findex_#Packet-in-Flow_#Mimetype-in-Flow
So they can easily being matched with the flow or packet file. If the flow containing the filename is not present Filename = HTTP_NONAME, defined in httpSniffer.h.

Name	Default	Description
HTTP_PATH	"/tmp/"	Root path
HTTP_IMAGE_PATH	HTTP_PATH"httpPicture/"	Path for pictures
HTTP_VIDEO_PATH	HTTP_PATH"httpVideo/"	Path for videos
HTTP_AUDIO_PATH	HTTP_PATH"httpAudio/"	Path for audios
HTTP_MSG_PATH	HTTP_PATH"httpMSG/"	Path for messages
HTTP_TEXT_PATH	HTTP_PATH"httpText/"	Path for texts
HTTP_APPL_PATH	HTTP_PATH"httpAppl/"	Path for applications
HTTP_PUNK_PATH	HTTP_PATH"httpPunk/"	Path for put/else
HTTP_NONAME_IMAGE	"nudel"	File name for unnamed content
HTTP_DATA_C_MAX	20	Maximum dim of all storage array: # / flow
HTTP_CNT_LEN	13	max # of cnt digits attached to file name
HTTP_FINDEX_LEN	20	string length of findex in decimal format.
HTTP_MXFILE_LEN	80	Maximum image name length in bytes
HTTP_MXUA_LEN	400	Maximum user agent name length in bytes
HTTP_MXXF_LEN	80	Maximum x-forward-for name length in bytes

1.2 Flow File Output

The default settings will result in six tab separated columns in the flow file where the items in column 4-6 are sequences of strings separated by ';'. Whereas an item switch is set to '0' only the occurrence of this item during the flow is supplied. It is a high speed mode for large datasets or real-time operation in order to produce an initial idea of interesting flows

maybe by script based post processing selecting also by the information supplied by first three columns.

Column	Type	Description	Flags
httpStat	H16	Status	
httpAFlags	H16	Anomaly flags	
httpMethods	H8	HTTP methods	
httpHeadMimes	H16	HEADMIME-TYPES	
httpCFlags	H8	HTTP content body info	HTTP_BODY=1
httpGet_Post	2U16	Number of GET and POST requests	HTTP_MCNT=1
httpRSCnt	U16	Response status count	HTTP_STAT=1
httpRSCode	RU16	Response status code	HTTP_STAT=1
httpURL_Via_Loc_Srv_ Pwr_UAg_XFr_ Ref_Cky_Mim	10U16	Number of URL, Via, Location, Server, Powered-By, User-Agent, X-Forwarded-For, Referer, Cookie and Mime-Type	
httpImg_Vid_Aud_Msg_ Txt_App_Unk	7U16	Number of images, videos, audios, messages, texts, applications and unknown	
httpHosts	RS	Host names	HTTP_HOST=1
httpURL	RS	URLs (including parameters)	HTTP_URL=1
httpMimes	RS	MIME-types	HTTP_MIME=1
httpCookies	RS	Cookies	HTTP_COOKIE=1
httpImages	RS	Images	HTTP_IMAGE=1
httpVideos	RS	Videos	HTTP_VIDEO=1
httpAudios	RS	Audios	HTTP_AUDIO=1
httpMsgs	RS	Messages	HTTP_MSG=1
httpAppl	RS	Applications	HTTP_APPL=1
httpText	RS	Texts	HTTP_TEXT=1
httpPunk	RS	Punk	HTTP_PUNK=1
httpBdyURL	RS	Body: Refresh, set_cookie URL	HTTP_BODY=1&& HTTP_BDURL=1
httpUsrAg	RS	User-Agent	HTTP_USRAG=1
httpXFor	RS	X-Forwarded-For	HTTP_XFRWD=1
httpRefrr	RS	Referer	HTTP_REFRR=1
httpVia	RS	Via (Proxy)	HTTP_VIA=1
httpLoc	RS	Location (Redirection)	HTTP_LOC=1
httpServ	RS	Server	HTTP_SERV=1
httpPwr	RS	Powered-By / Application	HTTP_PWR=1

1.2.1 httpStat

The httpStat column is to be interpreted as follows:

httpStat	Description
2^0 (=0x0001)	Warning: HTTP_DATA_C_MAX entries in flow name array reached
2^1 (=0x0002)	Warning: Filename longer than HTTP_MXFILE_LEN
2^2 (=0x0004)	Internal State: pending url name
2^3 (=0x0008)	HTTP Flow

httpStat	Description
2^4 (=0x0010)	Internal State: Chunked transfer
2^5 (=0x0020)	Internal State: HTTP Flow detected
2^6 (=0x0040)	Internal State: http header parsing in process
2^7 (=0x0080)	Internal State: sequence number init
2^8 (=0x0100)	Internal State: header shift
2^9 (=0x0200)	Internal State: PUT payload sniffing
2^{10} (=0x0400)	Internal State: Image payload sniffing
2^{11} (=0x0800)	Internal State: video payload sniffing
2^{12} (=0x1000)	Internal State: audio payload sniffing
2^{13} (=0x2000)	Internal State: message payload sniffing
2^{14} (=0x4000)	Internal State: text payload sniffing
2^{15} (=0x8000)	Internal State: application payload sniffing

1.2.2 httpAFlags

The `httpAFlags` column denotes HTTP anomalies regarding the protocol and the security. It is to be interpreted as follows:

httpAFlags	Description
2^0 (=0x0001)	Warning: POST query with parameters, possible malware
2^1 (=0x0002)	Warning: Host is IPv4
2^2 (=0x0004)	Warning: Possible DGA
2^3 (=0x0008)	Warning: Mismatched content-type
2^4 (=0x0010)	Warning: Sequence number mangled or error retry detected
2^5 (=0x0020)	Warning: Parse Error
2^6 (=0x0040)	Warning: header without value, e.g., `Content-Type: [missing]`
2^7 (=0x0080)	
2^8 (=0x0100)	Info: X-Site Scripting protection
2^9 (=0x0200)	Info: Content Security Policy
2^{10} (=0x0400)	
2^{11} (=0x0800)	
2^{12} (=0x1000)	Warning: possible exe download, check also mime type for conflict
2^{13} (=0x2000)	Warning: possible ELF download, check also mime type for conflict
2^{14} (=0x4000)	Warning: HTTP 1.0 legacy protocol, often used by malware
2^{15} (=0x8000)	

1.2.3 httpMethods

The aggregated `httpMethods` bit field provides an instant overview about the protocol state and communication during a flow. It can also be used during post processing in order to select only flows containing e.g. responses or delete operations.

httpMethods	Type	Description
(=0x00)	RESPONSE	Response of server identified by URL
2^0 (=0x01)	OPTIONS	Return HTTP methods that server supports for specified URL

httpMethods	Type	Description
2^1 (=0x02)	GET	Request of representation of specified resource
2^2 (=0x04)	HEAD	Request of representation of specified resource without BODY
2^3 (=0x08)	POST	Request to accept enclosed entity as new subordinate of resource identified by URI
2^4 (=0x10)	PUT	Request to store enclosed entity under supplied URI
2^5 (=0x20)	DELETE	Delete specified resource
2^6 (=0x40)	TRACE	Echo back received request
2^7 (=0x80)	CONNECT	Convert request connection to transparent TCP/IP tunnel

1.2.4 httpHeadMimes

The aggregated `httpHeadMimes` bit field provides an instant overview about the content of the HTTP payload being transferred during a flow. Thus, the selection of flows with certain content during post processing is possible even when the plugin is set to count mode for all items in order to conserve memory and processing capabilities. The 16 Bit information is separated into Mime Type (MT) and Common Subtype Prefixes (CSP) / special Flags each comprising of 8 Bit. This is experimental and is subject to change if a better arrangement is found.

httpHeadMimes	MT / CSP	Description
2^0 (=0x0001)	application	Multi-purpose files: java or post script, etc
2^1 (=0x0002)	audio	Audio file
2^2 (=0x0004)	image	Image file
2^3 (=0x0008)	message	Instant or email message type
2^4 (=0x0010)	model	3D computer graphics
2^4 (=0x0020)	multipart	Archives and other objects made of more than one part
2^5 (=0x0040)	text	Human-readable text and source code
2^6 (=0x0080)	video	Video stream: Mpeg, Flash, Quicktime, etc
2^8 (=0x0100)	vnd	vendor-specific files: Word, OpenOffice, etc
2^9 (=0x0200)	x	Non-standard files: tar, SW packages, LaTex, Shockwave Flash, etc
2^{10} (=0x0400)	x-pkcs	public-key cryptography standard files
2^{11} (=0x0800)	—	—
2^{12} (=0x1000)	pdf	—
2^{13} (=0x2000)	java	—
2^{14} (=0x4000)	—	—
2^{15} (=0x8000)	allelse	All else

1.2.5 httpCFlags

The `httpCFlags` contain information about the content body, regarding to information about rerouting. They have to be interpreted as follows:

httpBodyFlags	MT / CSP	Description
2^0 (=0x0001)	STCOOKIE	http set cookie
2^1 (=0x0002)	REFRESH	http refresh detected
2^2 (=0x0004)	HOSTNAME	host name detected
2^3 (=0x0008)	BOUND	Post Boundary marker

httpBodyFlags	MT / CSP	Description
2^4 (=0x0010)	PCNT	Potential HTTP content
2^5 (=0x0020)	—	
2^6 (=0x0040)	QUARA	Quarantine Virus upload
2^{15} (=0x8000)	—	

1.3 Plugin Report Output

The following information is reported:

- Max number of file handles (only if HTTP_SAVE=1)
- Number of HTTP IPv4/6 packets
- Number of HTTP #GET, #POST, #GET/#POST ratio
- Aggregated status flags (httpStat)
- Aggregated mimetype flags (httpHeadMimes)
- Aggregated anomaly flags (httpAFlags)
- Aggregated content flags (httpCFlags, only if HTTP_BODY=1)

The GET/POST ratio is very helpful in detecting malware operations, if you know the normal ratio of your machines in the network. The file descriptor gives you an indication of the maximum file handles the present pcap will produce. You can increase it by invoking `uname -n mylimit`, but it should not be necessary as we manage the number of handle being open to be always below the max limit.