erik zandvliet <s.p.the.ghost@gmail.com>

Chat with Jaron Viëtor

1 message

Jaron Viëtor <jaron@vietors.com> **Mon, Dec 27, 2010 at 9:18 PM**
To: s.p.the.ghost@gmail.com

8:36 PM **Jaron**: boe!
8:37 PM je wou wat uitleg over de Flash dinges parser parser?
err.. maar dan goed gespelt
8:38 PM AMF, dat
8:39 PM het AMFType datatype is een recursief gebeuren
8:40 PM **me**: yeah
hoe kan ik daar een beetje logisch info uit halen?
Jaron: regels 102 t/m 106 van parsechunks.cpp geven een voorbeeld van hoe je er info uit haalt
de functie ParseAMF geef je een datablock en de lengte daarvan op
8:41 PM **me**: okee
Jaron: en die returned een AMFType
getContent(i) haalt de i'ste content daarvan op
getContentP*
typo
me: okee
Jaron: actually, beide
P returned een pointer (en is dus sneller)
8:42 PM zonder P returned een AMFType object
me: okee
Jaron: wat je moet weten:
AMF objecten zijn basically wrappers voor data
ze hebben altijd een array-achtige vorm...
8:43 PM maar er zijn 2 soorten die je kan tegenkomen:
- de eerste is een "standaard" type. Deze geven altijd dezelfde hoeveelheid elementen terug, en altijd dezelfde types elementen. Alles is in principe een "standaard" type, behalve:
8:44 PM - het tweede type is een **echt** array type. Deze heeft indices (die strings zijn), waarvan je niet van te voren weet welke er bestaan en welke types ze zijn.
bijv
op de regels die ik net gaf, wordt er een standaard type geladen
me: okee
Jaron: flash heeft als conventie dat de eerste content altijd een string is, die het type bericht aangeeft
8:45 PM dus ik doe:
getContentP(0)->StrValue()
dus: pointer naar eerste (0e, want we beginnen bij 0) content opvraagt, en daarvan de string value.
8:46 PM als dat connect is (een object waarvan flash zelf aangeeft dat het een connect packet is - dit hoef je niet te snappen voor FLV parsing though)
dan haalt ie de 3e content op, wat toevallig de capabilities content is in alle connect packages

dit is een array type - en we halen daarvan de indice "videoCodecs" op, wat weer een getal is
met andere woorden:
getContentP(2)->getContentP("videoCodecs")->NumValue()
8:47 PM 3e content, daarvan de indice "videoCodecs", daarvan de int waarde.
makes sense so far?
me: yeah
Jaron: je kan deze AMF dingen ook opbouwen, maar dat heb je niet nodig geloof ik, right?
8:48 PM dus, wat je moet doen:
8:49 PM als je de metadata eenmaal hebt, gebruik je ParseAMF om deze om te zetten naar een handelbaar object
me: nee, heb ik niet nodig idd :p
Jaron: mijn tip is dan om even op dat object de Print() functie aan te roepen
dus Object->Print()
dan krijg je op je stderr een mooie extreem leesbare opsomming van de inhoud ervan met types en alles
8:50 PM **me**: okee :)
maar dan weet ik in ieder geval waar ik moet gaan zoeken enzo
Jaron: en je kan dus aannemen dat als indices niet genamed zijn, ze altijd in die volgorde met die types voorkomen
en als ze wel named zijn, moet je getContent("naam") gebruiken
me: okee :)
Jaron: (of getContentP("naam"))
die laatste is slimmer, aangezien die een 0 returned als ie niet bestaat
8:51 PM en de eerste een leeg object
me: 0 is af te handelen etc
Jaron: (je kan dus alleen crashen als je aanneemt dat de P versie altijd geldige pointers returned - maar kan bij de non-P versie geen onderscheid maken tussen een int 0 of een niet-bestaand object)
8:52 PM de P versie is dus de enige waarbij je zeker kan zijn dat een indice wel of niet bestaat :)
me: okee
Jaron: nog vragen na dit alles, of denk je verder te kunnen?
remember, vraag gerust ipv zelf tijd te verspillen :P
8:53 PM **me**: yeah idd, dat was ook waarom ik besloot maar verder te gaan :p
Jaron: precies
me: ik kan voorlopig zowiezo verder, het afmaken van die boxes moet zowiezo eerst af zijn, voor ik nuttig flv kan gaan parsen...
Jaron: zometeen komt btw de HTTP basis in de GIT te staan
ofc
me: maar ik zat er gewoon even naar te kijken, en kwam er daarna achter dat jij het al had geschreven
Jaron: hehe
8:54 PM yeah dat scheelt
want AMF is fucked up
me: dus toen dat geprobeerd uit te vinden, om te kijken of het makkelijk te implementen was
wat helaas even niet zo was :p
dus gingen we maar verder
8:55 PM **Jaron**: ik ben best tevreden over mn AMF implementatie
als je t eenmaal snapt issie heel chill om mee te werken :)
omdat alle types op 1 manier toegangkelijk zijn en zo
8:56 PM je doet ofwel getContentP/getContent, ofwel NumValue, ofwel StringValue

je kan btw geen verkeerde aanroep doen
StringValue op een numeriek type returned gewoon de lege string ipv te crashen
andersom returned 0 ipv een crash
en getContent als er geen contents zijn is nullpointer of een leeg AMFType
8:57 PM daadwerkelijk zitten er wel 20 types achter
maar hij doet de typemagie zelf achter de schermen :P
me: niceness :p
Jaron: dus alles is of een string, of een getal, of een container voor contents.
8:58 PM **me**: nice
Jaron: je moet btw een AMFType gemaakt door ParseAMF wel deleten na gebruik
anders heb je een memory leak :)
8:59 PM **me**: okee
Jaron: actually
exuser-moi
dat heb ik helemaal verkeerd onthouden
parseAMF returned het object zelf, geen pointer
9:00 PM ik declareer er zo eentje, aan het begin van mn functie/code: static AMFType amfdata("empty", (unsigned char)0xFF);
en daarna kan je die gewoon setten met = ParseAMF(data, len)
de =-operator is namelijk gedefined en doet zelf alle cleanup en zo
9:01 PM en omdat je geen new hebt gedaan is een delete daarna ook niet nodig
gaat vanzelf (C++ is nice)
(amfdata is hier ofc de naam van de variabele, die je mag setten naar wat je maar wilt...)
9:02 PM (en hij hoeft niet static te zijn, maar in een functie die zo vaak als de mijne wordt aangeroepen scheelt dat wat overhead van wissen/maken)
9:03 PM btw, als je amf.cpp gaat gebruiken, symlink hem dan even naar je directory ipv een kopie te maken :)
ish netjes en zo
en krijg je ook automagisch eventuele bugfixes en zo
9:04 PM **me**: ish gut
9:06 PM **Jaron**: je kan dus trouwens, mocht je dat nuttig vinden, AMF dingen gewoon kopieren met een =-statement
me: okee
9:07 PM **Jaron**: ik heb er zelf geen use voor gevonden
maar who knows :P
me: ik moet enkel de metadata omzetten naar f4v boxed
boxes*
Jaron: wat zit daarin, anyway? geen AMF, right?
me: nee, gewone strings
9:08 PM maar de data halen we uit flv
as far as I still know?
9:09 PM **Jaron**: yeah dat klopt
en dat is inderdaad in AMF format
in de metadata tags
9:11 PM en zo te zien ga je alleen de duration en encodings moeten gebruiken
of mis ik iets? :P
9:12 PM **me**: width, height, dpi en ongein
9:13 PM is wat ik nu al tegen ben gekomen :p
9:14 PM **Jaron**: dpi, staat dat in de metadata dan? nou ja, whatev :P
dpi kan je ook een default value voor nemen ofc
9:15 PM **me**: ja, er is een default value voor
maar width en height moet je wel encoden enzo
9:18 PM

Jaron: ofc, ofc
en die heb je ook :P
