The Listening

Email Client Backdoor

Esteban Guillardoy
esteban@immunityinc.com

Introduction

- This presentation will focus on a backdoor implementation based on Thunderbird 3.x
- Different approach taking advantage of the addon/extension features
- How to make it persistant and hide the C&C by using steganography

Demo

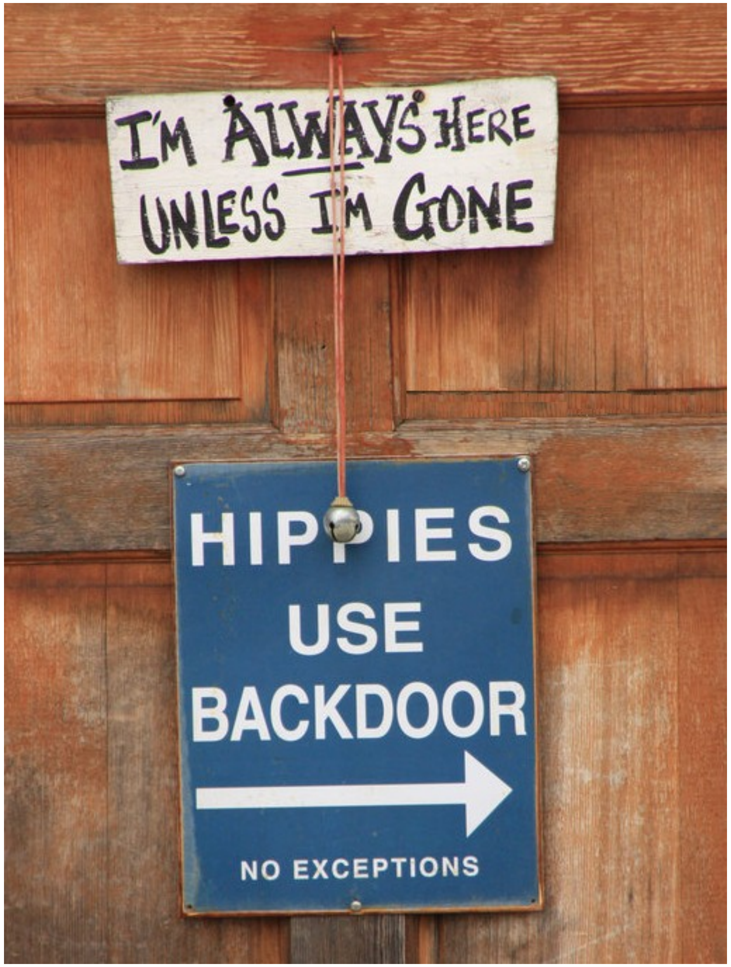
How cool is this presentation?
It is starting with a demo :)

How all this started

- Never leave the office without locking your session – FAIL!
- Malicious Brainstorming...

Adapting the idea

- Web Browsers are commonly targeted
- But Email Clients are not
- Why not using this as a real backdoor?

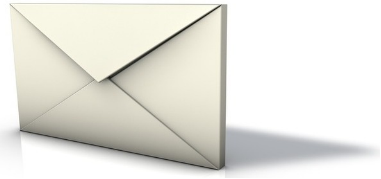


The challenge

- Targets go on and off
- Covertness without losing reliability
- Routing the data
- Stealthiness
- Resistance to traffic analysis
- No suspicious open ports
- Avoid antiviruses & scanners
- Thinking of future trojans

Why an email client

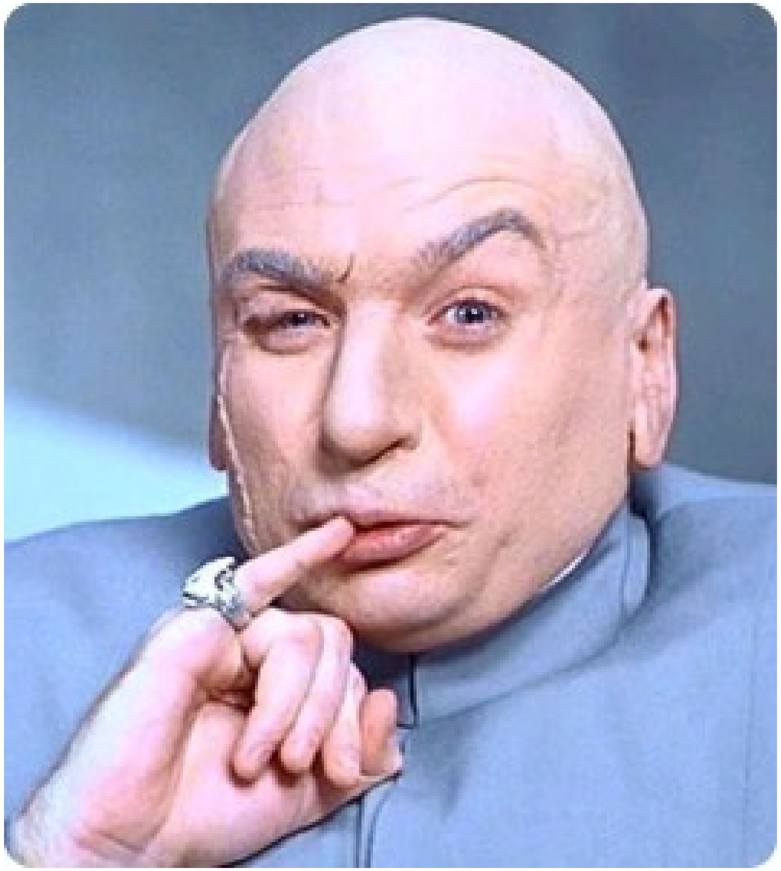
Don't you use one? Is it Thunderbird?



Email Client Extensions

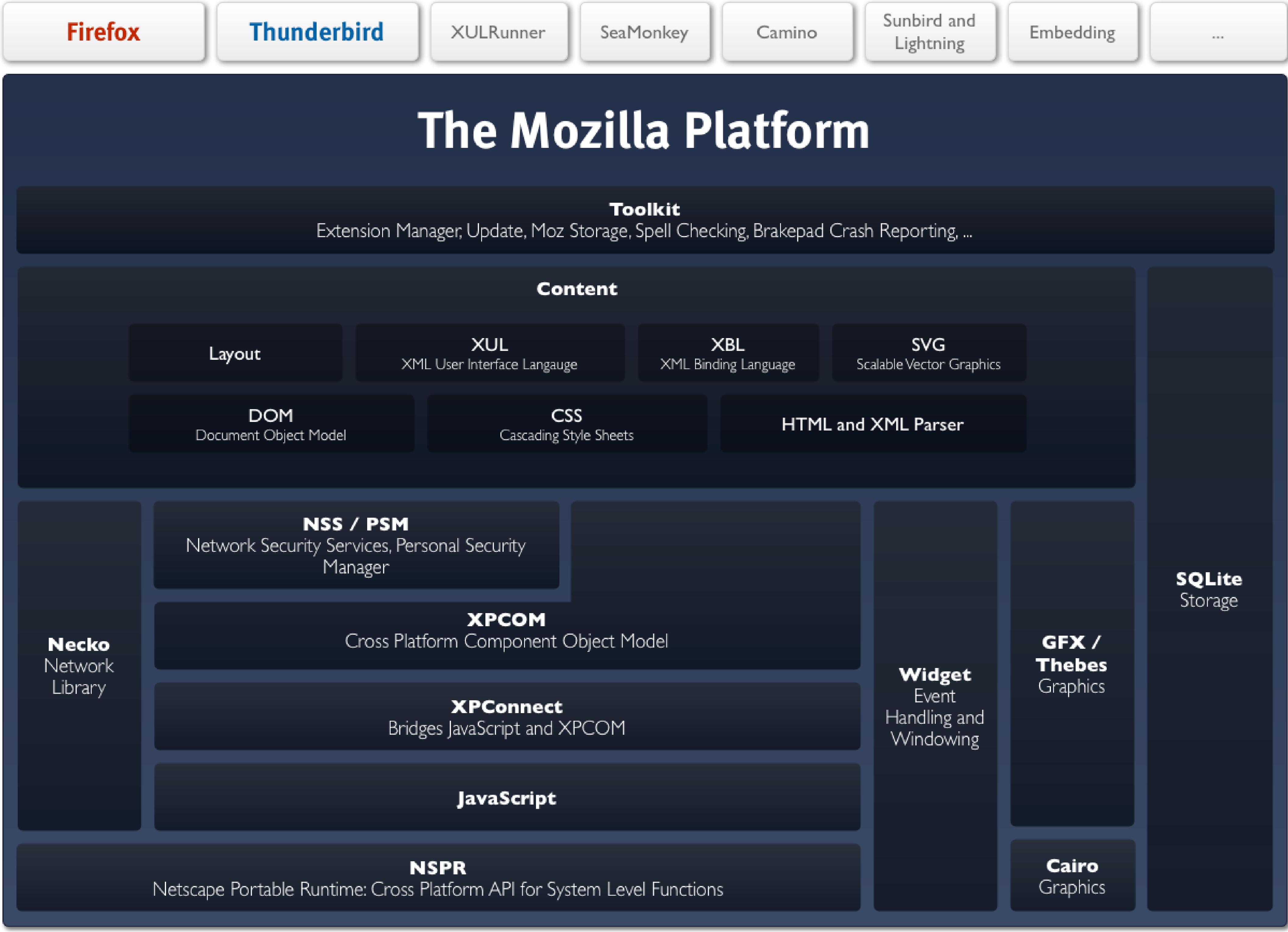
- Only Thunderbird 3.x for now
  - **multiplatform backdoor** out of the box
- Trusted code
- Full access to all client functions
- Program execution
- Easy development
- Solve us part of the challenge

Backdoor controlled by simply sending emails



Features

- Doesn't require user interaction
- Hidden C&C using steganography on images
- Encryption using public & private key
- Processes every email that arrives to the client
- Predefined Actions
- Command execution with output retrieval



Firefox

Thunderbird

XULRunner

SeaMonkey

Camino

Sunbird and Lightning

Embedding

...

The Mozilla Platform

Toolkit
Extension Manager, Update, Moz Storage, Spell Checking, Brakepad Crash Reporting, ...

Content

Layout

XUL
XML User Interface Langauge

XBL
XML Binding Language

SVG
Scalable Vector Graphics

DOM
Document Object Model

CSS
Cascading Style Sheets

HTML and XML Parser

SQLite
Storage

Necko
Network Library

NSS / PSM
Network Security Services, Personal Security Manager

XPCOM
Cross Platform Component Object Model

XPConnect
Bridges JavaScript and XPCOM

JavaScript

Widget
Event Handling and Windowing

GFX / Thebes
Graphics

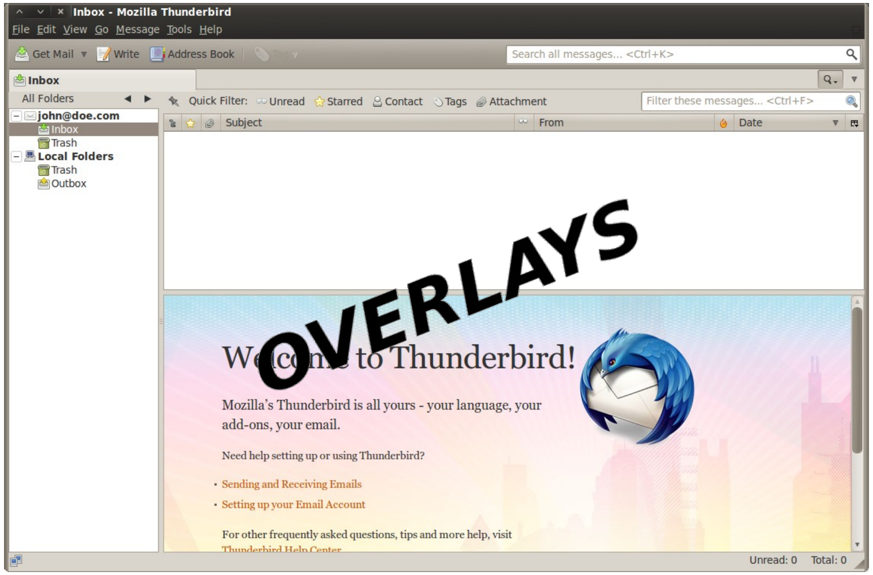
Cairo
Graphics

NSPR
Netscape Portable Runtime: Cross Platform API for System Level Functions

Mozilla Addons/Extensions

Basic structure:

```
    /components/*
    /content or /chrome/content
    /defaults/preferences/*.js
    /chrome.manifest
    /install.rdf
```



Welcome to Thunderbird!

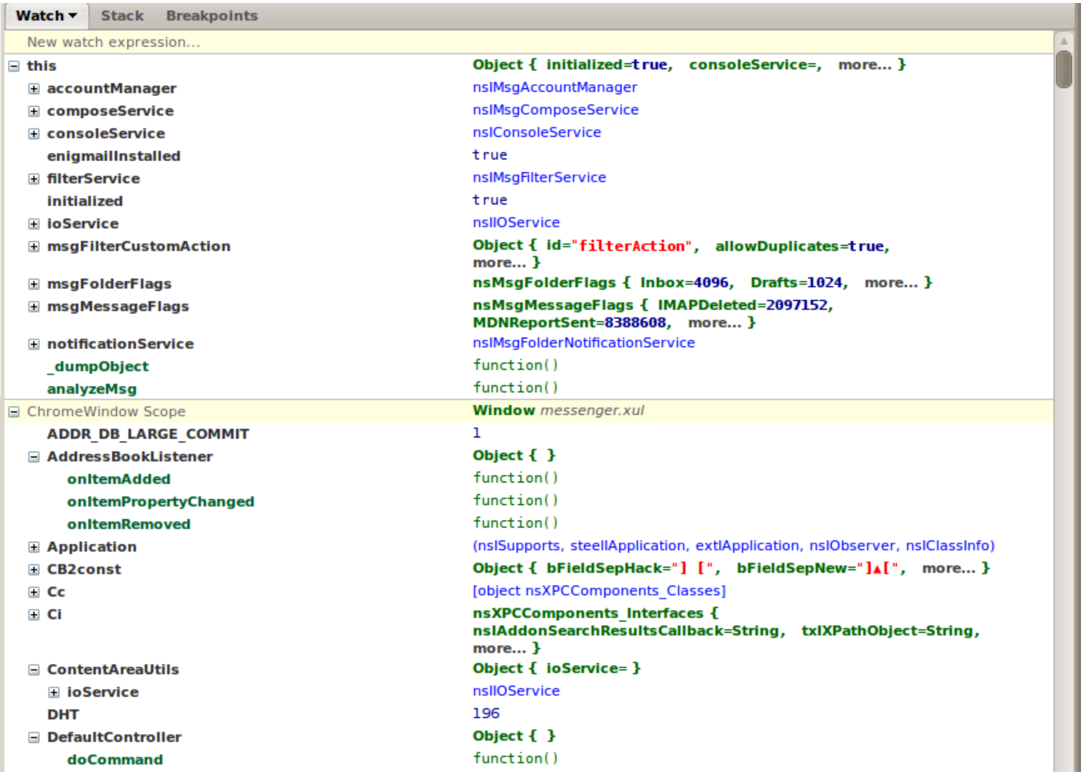
Mozilla's Thunderbird is all yours - your language, your add-ons, your email.

Need help setting up or using Thunderbird?

- Sending and Receiving Emails
- Setting up your Email Account

For other frequently asked questions, tips and more help, visit Thunderbird Help Center

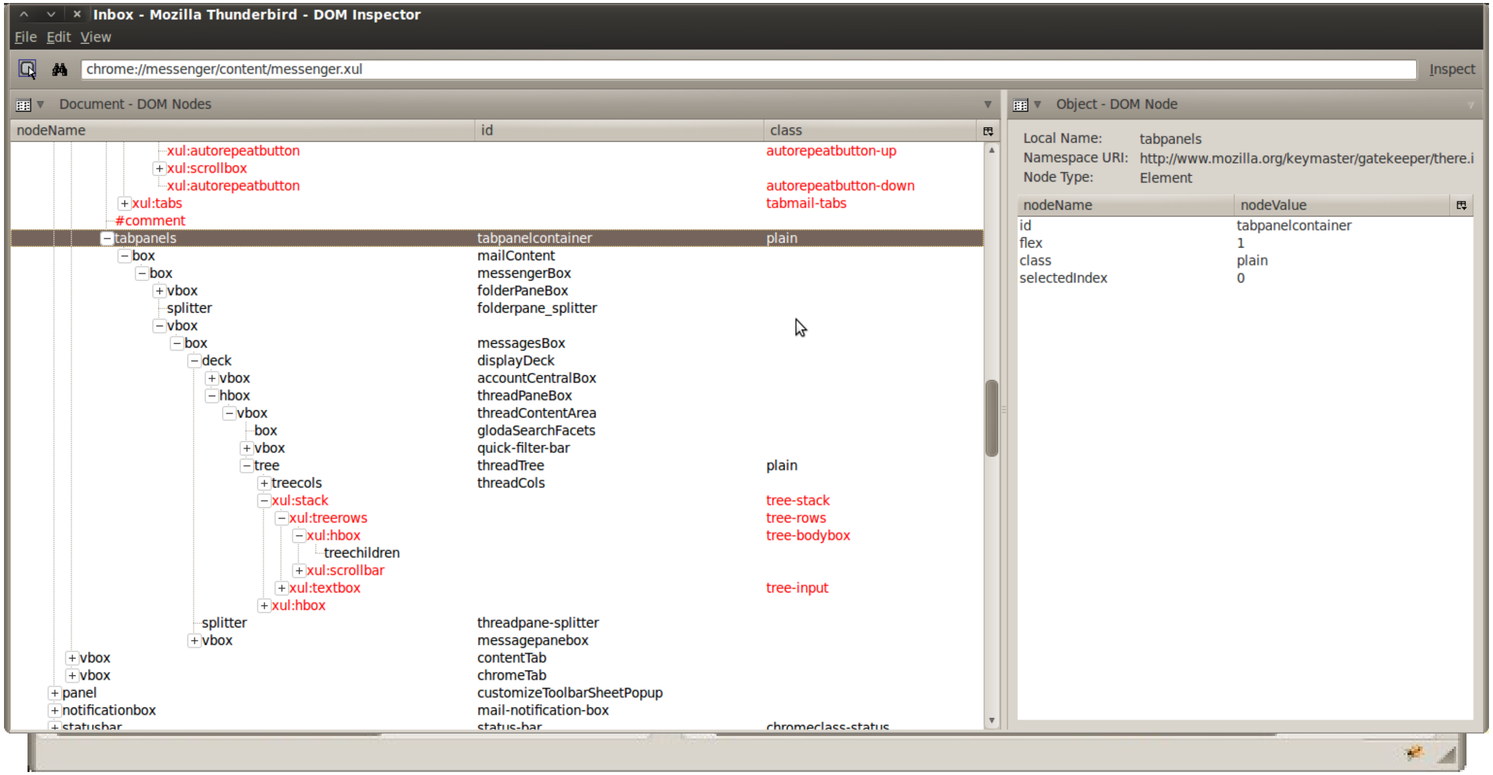
OVERLAYS



```
⊟ this                          Object { initialized=true,  consoleService=,  more... }
   ⊞ accountManager             nsIMsgAccountManager
   ⊞ composeService             nsIMsgComposeService
   ⊞ consoleService             nsIConsoleService
     enigmailInstalled          true
   ⊞ filterService              nsIMsgFilterService
     initialized                true
   ⊞ ioService                  nsIIOService
   ⊞ msgFilterCustomAction      Object { id="filterAction",  allowDuplicates=true,
                                more... }
   ⊞ msgFolderFlags             nsMsgFolderFlags { Inbox=4096,  Drafts=1024,  more... }
   ⊞ msgMessageFlags            nsMsgMessageFlags { IMAPDeleted=2097152,
                                MDNReportSent=8388608,  more... }
   ⊞ notificationService        nsIMsgFolderNotificationService
     _dumpObject                function()
     analyzeMsg                 function()
⊟ ChromeWindow Scope            Window messenger.xul
     ADDR_DB_LARGE_COMMIT       1
   ⊟ AddressBookListener        Object { }
       onItemAdded              function()
       onItemPropertyChanged    function()
       onItemRemoved            function()
   ⊞ Application                (nsISupports, steelIApplication, extIApplication, nsIObserver, nsIClassInfo)
   ⊞ CB2const                   Object { bFieldSepHack="] [",  bFieldSepNew="]▲[",  more... }
   ⊞ Cc                         [object nsXPCComponents_Classes]
   ⊞ Ci                         nsXPCComponents_Interfaces {
                                nsIAddonSearchResultsCallback=String,  txIXPathObject=String,
                                more... }
   ⊟ ContentAreaUtils           Object { ioService= }
     ⊞ ioService                nsIIOService
     DHT                        196
   ⊟ DefaultController          Object { }
       doCommand                function()
```

Development

- "Must have" tools
  - Firebug + ChromeBug
  - Chrome List
  - Console2
  - DOM Inspector
  - Event Spy
  - Extension Developer
  - Extension Manager Extended
  - Inspector Widget
  - MozRepl
  - XPCOMViewer



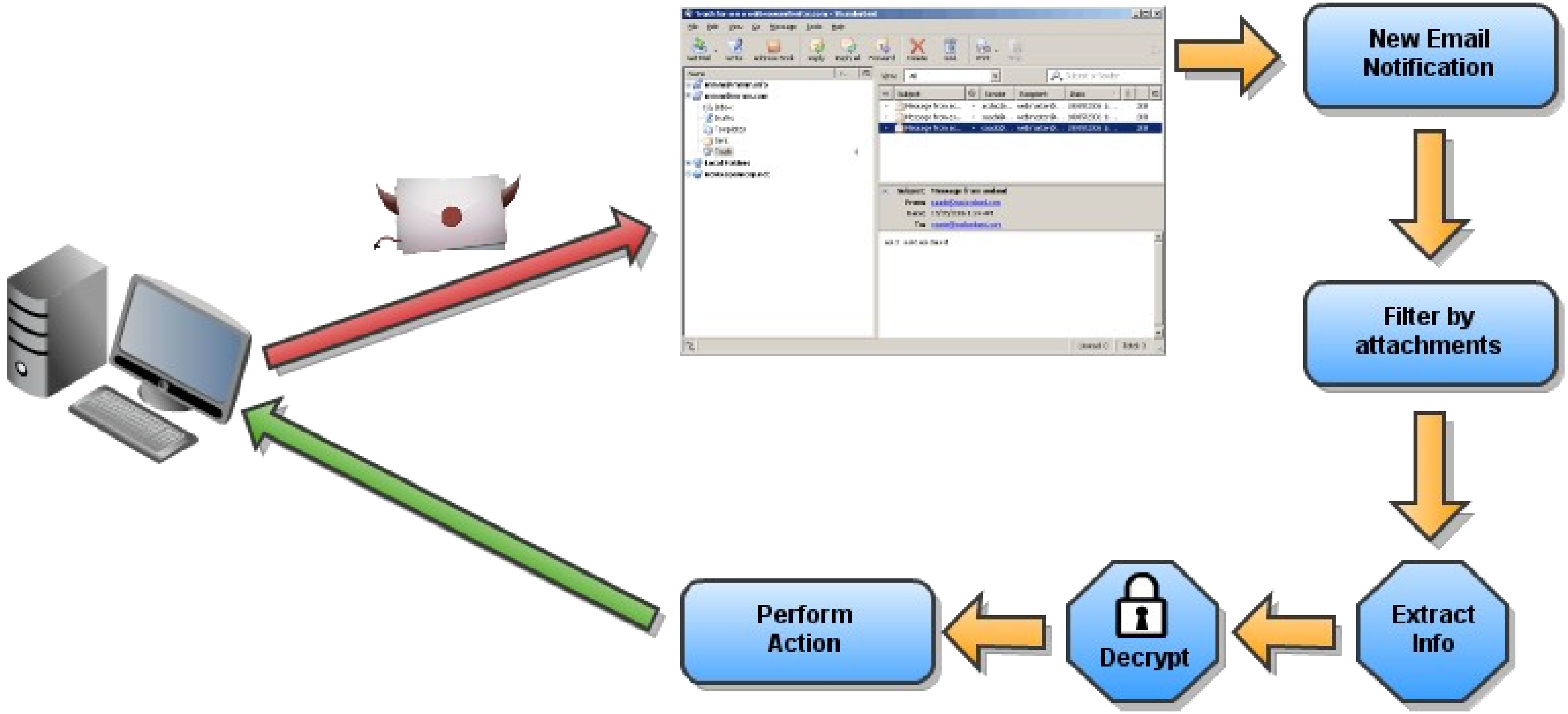
chrome://messenger/content/messenger.xul

nodeName	id	class
xul:autorepeatbutton		autorepeatbutton-up
+ xul:scrollbox		
xul:autorepeatbutton		autorepeatbutton-down
+ xul:tabs		tabmail-tabs
#comment		
− tabpanels	tabpanelcontainer	plain
− box	mailContent	
− box	messengerBox	
+ vbox	folderPaneBox	
splitter	folderpane_splitter	
− vbox		
− box	messagesBox	
− deck	displayDeck	
+ vbox	accountCentralBox	
− hbox	threadPaneBox	
− vbox	threadContentArea	
box	glodaSearchFacets	
+ vbox	quick-filter-bar	
− tree	threadTree	plain
+ treecols	threadCols	
− xul:stack		tree-stack
− xul:treerows		tree-rows
− xul:hbox		tree-bodybox
treechildren		
+ xul:scrollbar		
+ xul:textbox		tree-input
+ xul:hbox		
splitter	threadpane-splitter	
+ vbox	messagepanebox	
+ vbox	contentTab	
+ vbox	chromeTab	
+ panel	customizeToolbarSheetPopup	
+ notificationbox	mail-notification-box	
+ statusbar	status-bar	chromeclass-status

Local Name: tabpanels
Namespace URI: http://www.mozilla.org/keymaster/gatekeeper/there.i
Node Type: Element

nodeName	nodeValue
id	tabpanelcontainer
flex	1
class	plain
selectedIndex	0

How it works


New Email Notification
Filter by attachments
Extract Info
Decrypt
Perform Action

Email Check

- Listener on notification service

```
Components.classes["@mozilla.org/messenger/msgnotificationservice;1"];
notificationService.addListener(this, notificationService.msgsClassified);
```

- Our method gets called with each new email

- Filter messages by checking attachments

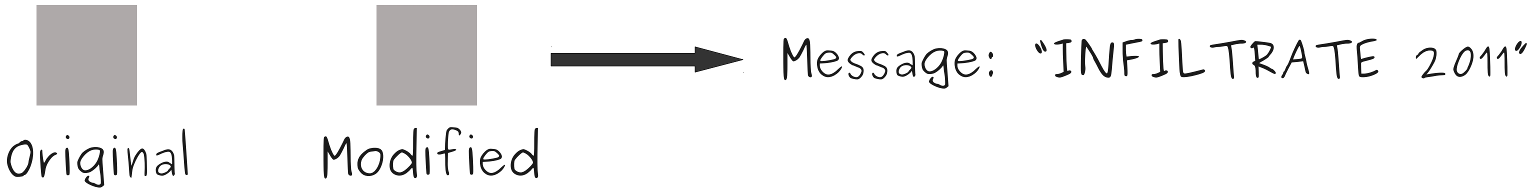
```
“attachment.contentType.match(/image\/png/) != null“
```

Encryption

- Private & Public key algorithm (PGP)
- Used to send commands & output
- Implementation in Javascript
- Wrapper around gnupg in Python

Hiding Information

- Steganography on images to hide the info
- Who applies steganalysis on every image attached on an email?
- Common approach is to avoid external images from loading



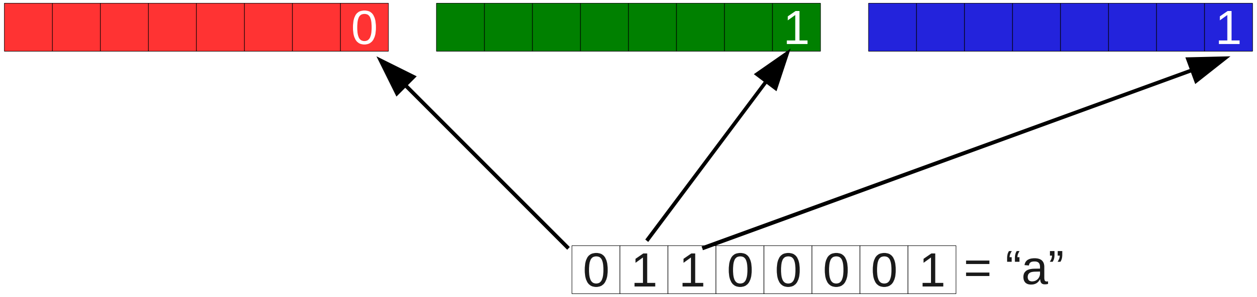
Original

Modified

Message: “INFILTRATE 2011”

Hiding Information

- Least Significant Bit (LSB) algorithm



0 1 1 0 0 0 0 1 = “a”

- We need 3 pixels per byte to hide
- If image is greyscale we could use more than 1 bit per pixel

Hiding Information

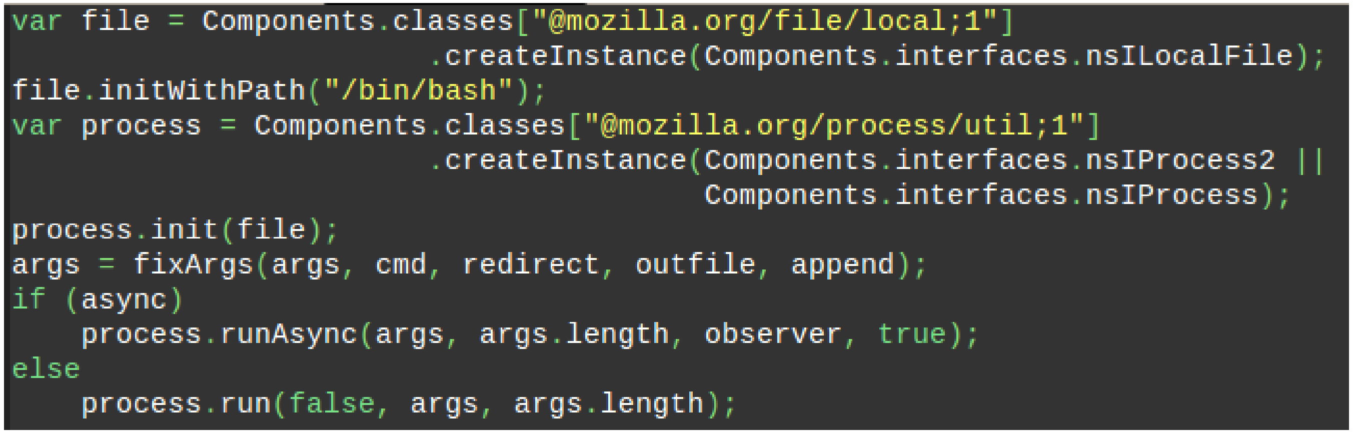
- Python Implementation
  - Using Python Imaging Library (PIL)
  - Some bitwise operations and we are ready

- Javascript Implementation
  - Hidden iframe to create a HTML5 canvas element
  - Retrieve pixel info with:

```
var context = canvas.getContext('2d');
var data = context.getImageData(0,0,canvas.width,canvas.height);
```

Execution

- Using XPCOM interfaces nsIProcess or nsIProcess2



```
var file = Components.classes["@mozilla.org/file/local;1"]
                        .createInstance(Components.interfaces.nsILocalFile);
file.initWithPath("/bin/bash");
var process = Components.classes["@mozilla.org/process/util;1"]
                        .createInstance(Components.interfaces.nsIProcess2 ||
                                        Components.interfaces.nsIProcess);
process.init(file);
args = fixArgs(args, cmd, redirect, outfile, append);
if (async)
    process.runAsync(args, args.length, observer, true);
else
    process.run(false, args, args.length);
```

- Fix arguments to redirect output to temp file

- Read temp file and then delete it

Getting Output

1) XMLHttpRequest

2) Sending an email

- New email:

  Components.classes["@mozilla.org/messengercompose;1"]

  Components.classes["@mozilla.org/messenger/account-manager;1"]

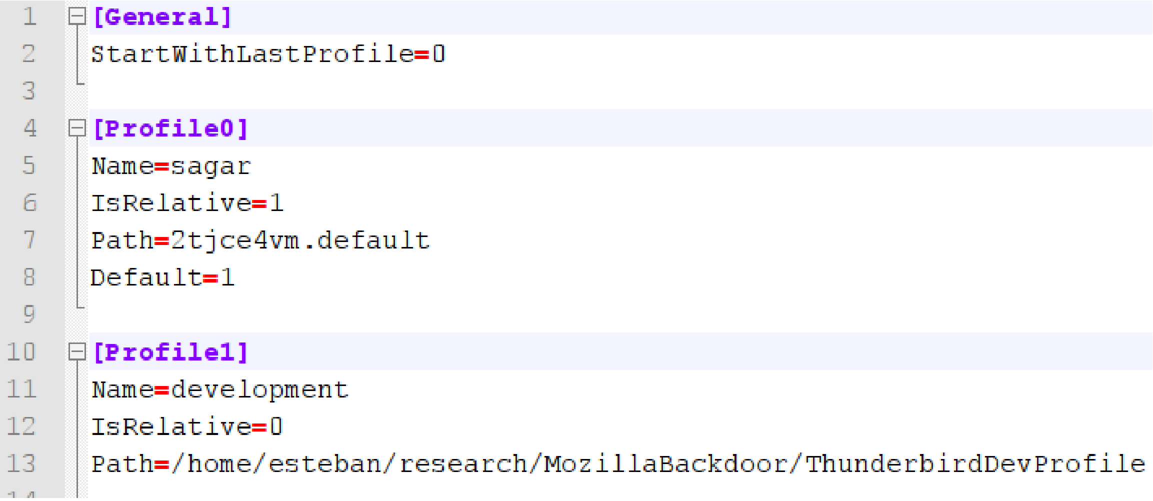
- Send it:

  Components.classes["@mozilla.org/messengercompose/compose;1"]

- Delete it from Sent folder

Deployment

- Discover profiles by reading profiles.ini:
  - **Windows**, usually in %AppData% \Thunderbird\
  - **Linux**, usually in ~/.thunderbird/ or ~/.mozilla-thunderbird/
  - **Mac OS X**, usually in ~/Library/Thunderbird/



```
[General]
StartWithLastProfile=0

[Profile0]
Name=sagar
IsRelative=1
Path=2tjce4vm.default
Default=1

[Profile1]
Name=development
IsRelative=0
Path=/home/esteban/research/MozillaBackdoor/ThunderbirdDevProfile
```

Deployment - Injecting Existing Addon

1) Installed addons in %profile-dir%/extensions.ini
2) Copy backdoor into %selected-addon%/content/
3) Edit chrome.manifest

```
    overlay    chrome://messenger/content/messenger.xul
   chrome://selected-addon/content/backdoorOverlay.xul
```

- Hard to detect
- User trusts installed addons
- Addon updates are a problem

Deployment - New Addon

1) Copy backdoor into TB extensions folder
2) Create a file with random name (an uuid)
3) write the path to backdoor folder

- May be easily detected by looking a the Extensions Manager
- But we can use a trick to hide it

EVIL BACKDOOR INSTALLED
I NOW CONTROLZ
YOUR COMPUTER

Deployment alternatives

- Install Manifest (install.rdf)
  - <em:updateURL>
  - <em:updateKey>

- Mozilla Addons Updates
  1) Update manifest retrieved in a secure fashion
     - Through SSL
     - Signed Update Manifests
  2) Update package retrieved matches
     - Through SSL
     - File Hashes

- Publishing on Mozilla Addon Site (AMO)
  - Policies & Review Process
  - Sandbox then public
  - Blocklist

Deployment alternatives

- MITM to deliver fake updates
- (P)Owning widely used addon sites (?)
- Become a reviewer for a long time (?)
- Using Mozilla cert to sign updates #comodogate :P
- Zamboni project (new AMO site)

  Source code available

  - https://github.com/jbalogh/zamboni
  - https://github.com/mozilla/zamboni

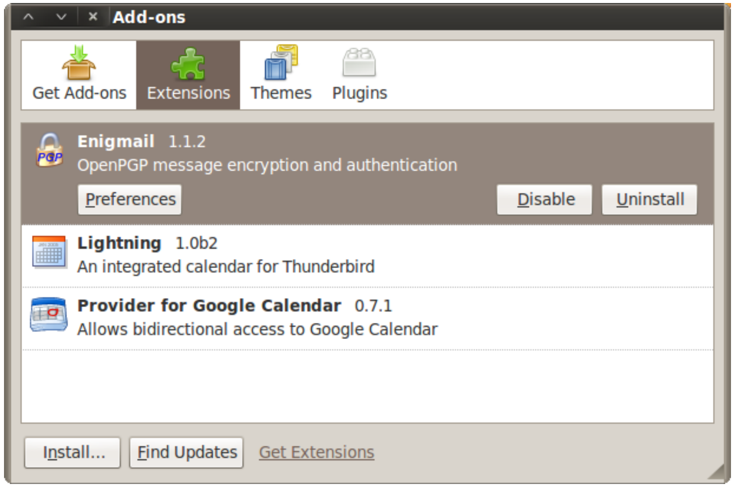
  Audit the code and test you said?

  Master visible on https://preview.addons.mozilla.org

  Next branch visible on https://next.addons.mozilla.org

Avoiding detection

- <em:hidden> deprecated since Gecko 1.9.2
- Hooking Extensions Manager
  - Overlay for chrome://mozapps/content/extensions/extensions.xul
  - Some javascript code to filter our extension chrome://mozapps/content/extensions/extensions.js



Add-ons

Get Add-ons | Extensions | Themes | Plugins

Enigmail 1.1.2
OpenPGP message encryption and authentication

Preferences | Disable | Uninstall

Lightning 1.0b2
An integrated calendar for Thunderbird

Provider for Google Calendar 0.7.1
Allows bidirectional access to Google Calendar

Install... | Find Updates | Get Extensions

Avoiding detection

- Skip updates by editing install.rdf file:

```
<em:updateURL>FAKE URL HERE</em:updateURL>
```

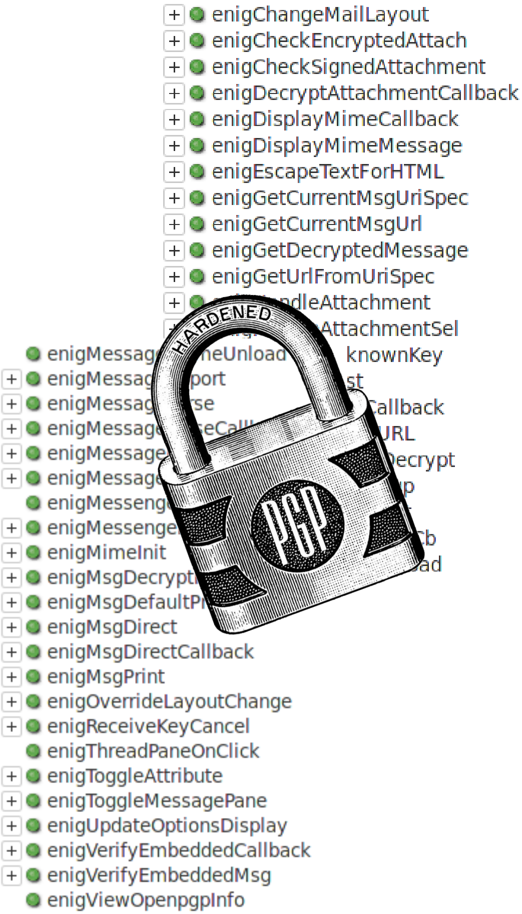
This url could also be used to update our backdoor

- Disabling extensions updates globaly:
    - extensions.update.enabled
    - extensions.update.interval
    - extensions.update.url

Capabilities Demo

Getting PGP Information

- Enigmail Addon commonly used
- Hook "enigMessageDecrypt"
- Prompt for passphrase twice
- EnigGetSecretKeys & enigmailSvc.extractKey FTW
- Match passphrase with ID

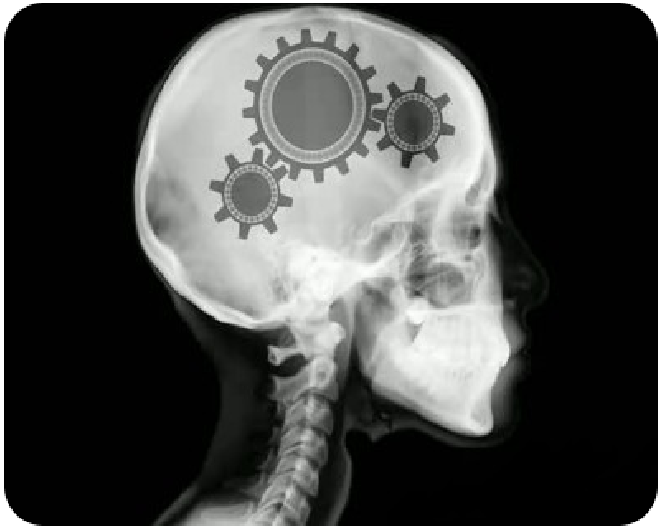


enigChangeMailLayout
enigCheckEncryptedAttach
enigCheckSignedAttachment
enigDecryptAttachmentCallback
enigDisplayMimeCallback
enigDisplayMimeMessage
enigEscapeTextForHTML
enigGetCurrentMsgUriSpec
enigGetCurrentMsgUrl
enigGetDecryptedMessage
enigGetUrlFromUriSpec
ndleAttachment
AttachmentSel
knownKey
st
Callback
URL
Decrypt
p
Cb
ad

enigMessag
eUnload
enigMessag
port
enigMessag
se
enigMessage
eCall
enigMessage
enigMessage
enigMesseng
enigMessenge
enigMimeInit
enigMsgDecrypt
enigMsgDefaultP
enigMsgDirect
enigMsgDirectCallback
enigMsgPrint
enigOverrideLayoutChange
enigReceiveKeyCancel
enigThreadPaneOnClick
enigToggleAttribute
enigToggleMessagePane
enigUpdateOptionsDisplay
enigVerifyEmbeddedCallback
enigVerifyEmbeddedMsg
enigViewOpenpgpInfo

Improvements

- Better steganography algorithms
- Unicode steganography
- Inject all addons
- More methods to get output



Alternative uses

- Building a SPAM controlled botnet
- others?

Conclusion

- Complete SDK to develop
- Global scope useful for us
- Multiplatform backdoor
- Hijacked extensions are hard to detect
- Execution with common user but..
- Further research on other email clients

Reference & Similar work

- Mozilla Develper Network
- mozillaZine KB & Forum
- StackOverflow questions

- Immunity PINK Framework
- Abusing Firefox Addons at Defcon17
- Digninja twitter botnet (unicode steg)
- IronGeek steg botnet

The End

Thank you for your time

Questions?

Esteban Guillardoy

esteban@immunityinc.com
@sagar38