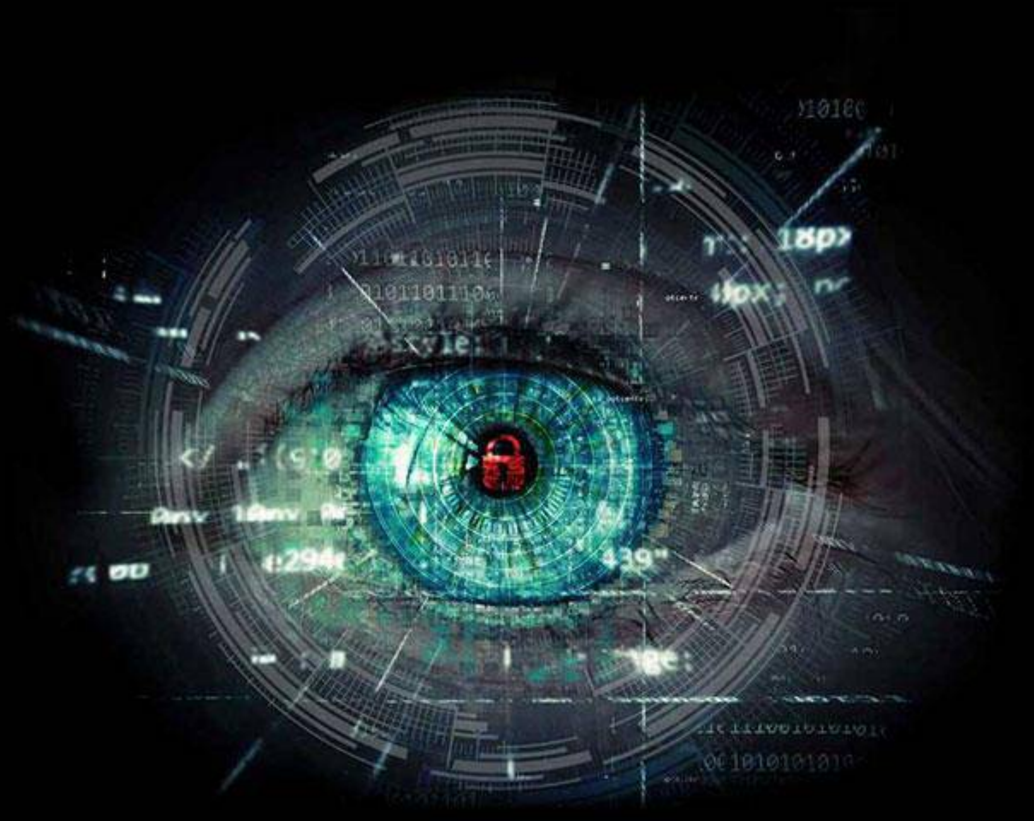
HELLODIGI.IR

Web Shell Detection & Prevention (English)

Abolfazl razipour



Contents

START

www.hellodigi.ir

Abolfazl Razipour

Hi, I am a senior network security professional with more than 8 years of experience in network security. During this time, I have worked with developing and implementing security solutions for large and small networks, identifying and fixing network vulnerabilities, analyzing security attacks and managing security events in various organizations.

https://hellodigi.ir/profile/razipoor

razipoorabolfazl@yahoo.com

“Decide where you want to be and don’t stop until you get there”

Introduction

First, we discussed the use of WebShell detection methods on Linux and Windows operating systems, and then discussed the use of log analysis tools such as Splunk and ELK Stack to review logs. Also we've talked about using IDS/IPS tools like Snort and Suricata and WAFs like Mod Security to prevent attacks and detect webshell. Finally we checked the detection method in wireshark tool

I hope this introduction will help you to better understand the topics raised and you can use this information to increase the security of your systems.

Webshell

“

A webshell is a type of script that runs on a web server and allows an attacker to execute commands on the server through a web browser or web-related tool. This script can be written in various programming languages, such as PHP, ASP, JSP, Perl, Python, etc. Web shells can allow an attacker to view, edit, or delete files, steal data, perform espionage operations, steal system resources, or even use the server as part of a botnet.

”

Simple webshell example

This is a simple web shell for PHP: using this web shell, commands can be sent via the "cmd" parameter.

```
<?php
   if(isset($_REQUEST['cmd'])){
      $cmd = ($_REQUEST['cmd']);
      system($cmd);
   }
?>
```

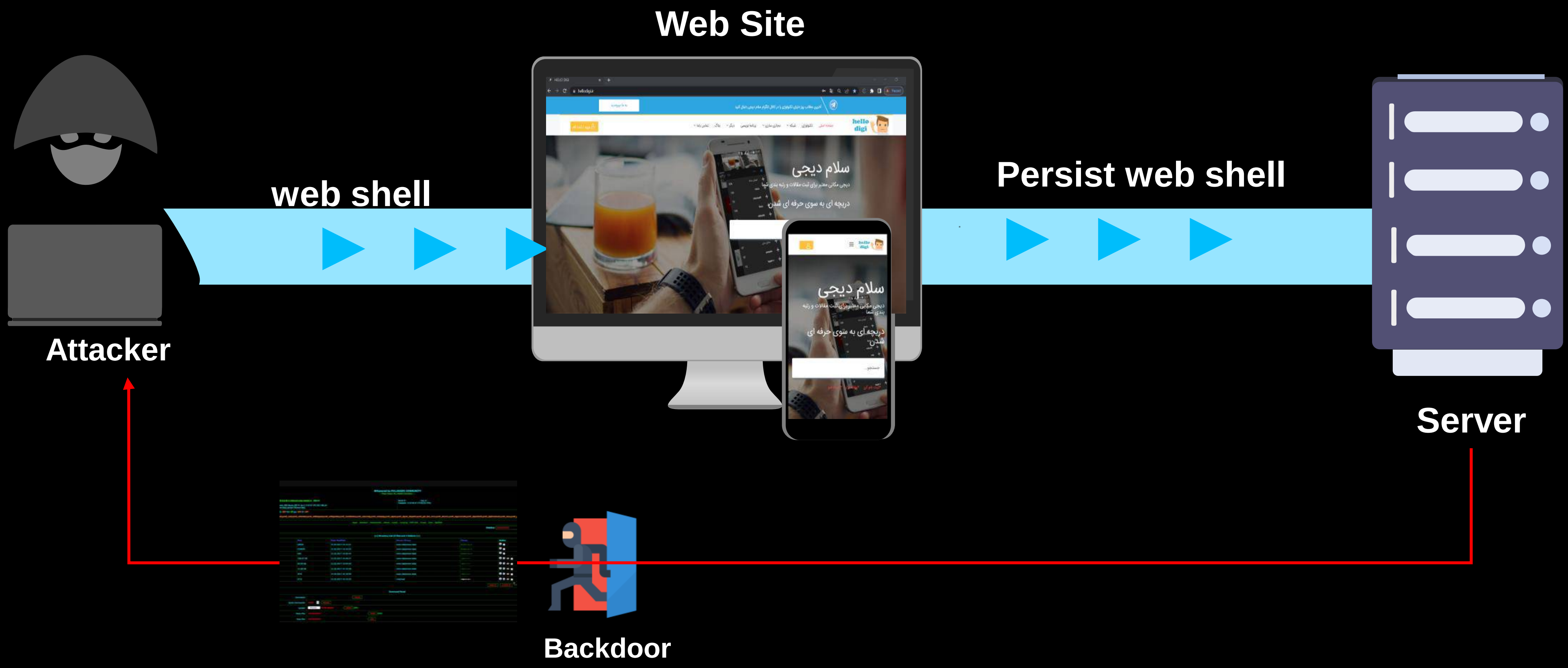
It is a simple web shell for ASP. Similar to the PHP web shell, using this web shell commands can be sent through the "cmd" form field.
In both examples, the web shell can execute commands through an HTTP request.

```
<%
If Request.Form("cmd")<>"" Then
   Dim cmd, rs
   Set cmd = Server.CreateObject("WScript.Shell")
   Set rs = cmd.Exec("cmd /c " & Request.Form("cmd"))
   Response.Write(rs.StdOut.ReadAll)
End If
%>
```

One of the things you should pay attention to is that using web shells for illegal purposes is prohibited and can cause legal problems. Also, the coding of web shells requires knowledge and experience and must be done very carefully so as not to threaten the security of the system. Below are two simple examples of web shells in PHP and ASP languages:

Abolfazl razipour

Below is a simple example of a Web Shell intrusion process:



Abolfazl razipour

Abolfazl razipour

The most famous webshell used

30%

C99 WebShell

25%

China Chopper WebShell

20%

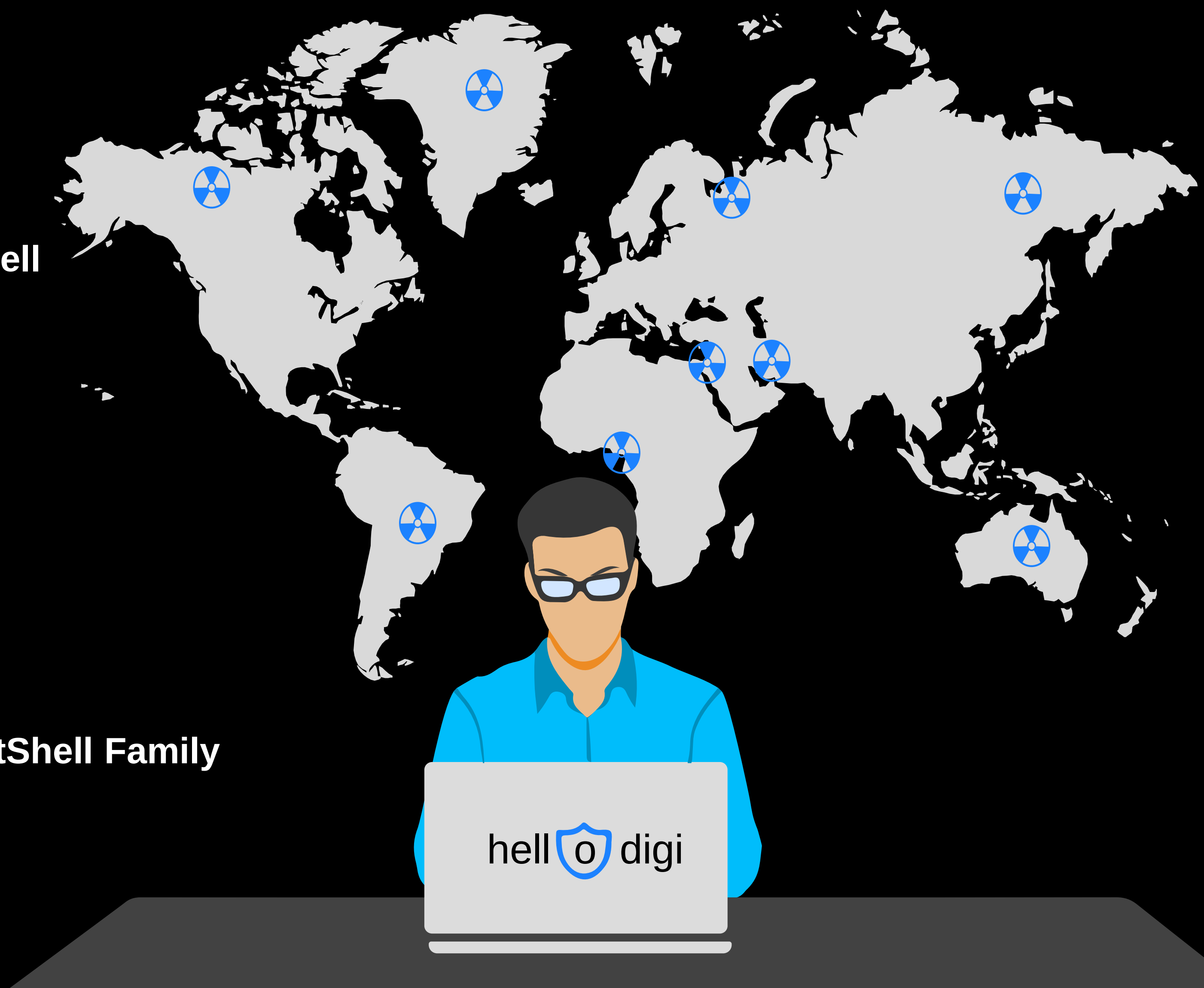
JFolder WebShell

15%

Tux WebShell

10%

B374K WebShell / RootShell Family



Apache web server log

“

Apache HTTP Server, commonly referred to simply as Apache, is an open source web server developed by the Apache Software Foundation.
Apache was one of the first web servers that were provided for web services, and with its help, web sites can be hosted on the Internet.

It stores its logs in the following paths:

For Debian-based systems such as Ubuntu:

```
/var/log/apache2/access.log
/var/log/apache2/error.log
```

For RedHat based systems as CentOS:

```
/var/log/httpd/access_log
/var/log/httpd/error_log
```

”

Apache web server log

Apache log sample

The default structure of an Apache log includes the following fields:

```
127.0.0.1 - razipour [10/Oct/2000:13:55:36 -0700] "GET /apache.gif HTTP/1.0" 200 2326 "http://hellodigi.ir" "Mozilla/4.08 [en] (Win98; I ;Nav)"
```

127.0.0.1 :This field indicates the IP address of the client or user who sent the request.

razipour: **This field indicates the username of the user connected to the server via HTTP authentication or FTP authentication.**

[10/Oct/2000:13:55:36 -0700]: This field shows the time and date of the request.

/iisstart.htm: This field shows the requested path or URL.

"GET /apache_pb.gif HTTP/1.0“:This field shows the command sent to the server. In this example, "GET" is the HTTP method, "/apache_pb.gif" is the path to the requested file, and "HTTP/1.0" is the HTTP protocol version used.

200:This field displays the HTTP status code that indicates the result of the request. In this example, 200 means "successful" or "OK"

2326 :This field displays the HTTP status code that indicates the result of the request. In this example, 200 means "successful" or "OK"..

:"http://www.hellodigi.ir" This field indicates the web address of the page from which the user has linked to the desired file. This field may be "-" which means no information is available

"Mozilla/4.08 [en] (Win98; I ;Nav) :This field is related to browser and system information

Apache was one of the first web servers that were provided for web services, and with its help, web sites can be hosted on the Internet.

Abolfazl razipour

The Webshell log on the Apache web server

Apache log example

Access to files with suspicious names

```
192.168.1.102 - - [12/May/2023:15:45:32 +0000] "GET /c99.php HTTP/1.1" 200 4523 "http://hellodigi.ir/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/53.3"
```

Use of questionable functions in request parameters

```
192.168.1.105 - - [12/May/2023:16:21:15 +0000] "GET /index.php?eval=base64_decode('c29tZSBjb2RlIGhlcmU=') HTTP/1.1" 200 1287 "http://hellodigi.ir/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/ (KHTML, like Gecko) Chrome"
```

Access to files with suspicious extensions

```
192.168.1.108 - - [12/May/2023:17:54:21 +0000] "GET /uploads/r57.jsp HTTP/1.1" 200 2516 "http://hellodigi.ir/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3"
```

Execute system commands in a PHP file

```
192.168.1.108 - - [12/May/2023:17:54:21 +0000] "GET /uploads/r57.jsp HTTP/1.1" 200 2516 "http://hellodigi.ir/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3"
```

Using the eval function to run the encrypted code

```
192.168.1.108 - - [12/May/2023:17:54:21 +0000] "GET /uploads/r57.jsp HTTP/1.1" 200 2516 "http://hellodigi.ir/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3"
```

These example logs may indicate suspicious activity related to webshells. However, to confirm the reality of these cases, more research and a more detailed investigation is needed, which we will do together

Abolfazl razipour

IIS web server log

“ Internet Information Services (IIS) is a web server owned by Microsoft and designed for use in Windows operating systems. IIS supports HTTP, HTTP/2, HTTPS, FTP, FTPS, SMTP, and NNTP.
IIS is part of the Windows operating system and is installed with it by default
IIS (Internet Information Services) web server logs, provided by Microsoft, are a record of all requests and responses processed by the web server. These logs provide important information that can be used for performance analysis, troubleshooting, security, and web server optimization. Microsoft's IIS server stores its logs in the following path by default:%SystemDrive%\inetpub\logs\LogFiles ”

IIS web server log

Sample IIS log

The default structure of an IIS log includes the following fields:

```
2023-02-25 00:00:00 192.168.5.55 GET /iisstart.htm - 80 - 192.18.15. 5 Mozilla/4+(compatible;Windows++Server) - 200 0 0 187
```

2023-02-25 00:00:00: **This field shows the time and date of the request.**

192.168.5.55: **This field indicates the IP address of the client or user who sent the request.**

.**GET: This field indicates the HTTP method used. In this example, GET is used**

/iisstart.htm: This field shows the requested path or URL.

:- This field is usually used to indicate the service port or username. In this example, both are empty.

80: This field indicates the server port.

192.18.15.5 :This field indicates the IP address of the server.

Mozilla/4+(compatible;Windows++Server) :**This field indicates the client's browser specifications.**

:– **This field is usually used to indicate the referring URL. In this example, the referrer URL does not exist**.

0 :The code below shows the status of IIS.

0:This code field shows the following Win32 status.

187 : This field indicates the number of bytes sent to the client in response. This value does not include HTTP .headers. In this example, 187 bytes are sent

IIS (Internet Information Services) web server logs, provided by Microsoft, are a record of all requests and responses processed by the web server.

Abolfazl razipour

The webshell log on the IIS web server

Sample IIS log

Access to files with suspicious names

```
2023-05-08 19:35:46 192.168.1.101 GET /cmd.jsp - 80 - 198.51.100.2
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/58.0.3029.110+Safari/537.36 200 0 0 15
```

Access to files with suspicious extensions

```
2023-05-10 10:16:45 192.168.0.2 POST /uploads/shell.asp - 443 - 10.0.0.1 Mozilla/5.0 - 200 0 0 2345
```

Execute system commands

```
2023-05-10 10:17:30 192.168.0.2 POST /uploads/shell.asp cmd=dir 443 - 10.0.0.1 Mozilla/5.0 - 200 0 0 3456
```

These example logs may indicate suspicious activity related to webshells. However, to confirm the reality of these cases, more research and a more detailed investigation is needed, which we will do together

Abolfazl razipour

Webshell detection in Linux 02

“There are four key steps to detecting webshells on Linux:

1. Check files: Server files should be checked with cut-grep-awk commands to find suspicious or unusual files that may be webshells.
2. Check logs: Web server logs can indicate web shell attacks, so they should be checked and analyzed.
3. Use of specialized tools: various diagnostic tools such as IDS/IPS... And user activity monitoring tools can be useful in detecting webshells.
4. Network traffic analysis**: Using network traffic analysis tools such as tshark, suspicious traffic can be identified and investigated.”

Webshell detection in Linux

detect webshells on Linux systems, you can use various available tools and commands. Here are some methods and commands you can use to find webshells on Linux:

Using the find command to find files with suspicious names:

```
find /var/www -type f \( -iname "*.php" -o -iname "*.jsp" -o -iname "*.asp" -o -iname "*.aspx" \) -exec grep -l -E 'c99|r57|wso' {} \;
```

Using the grep command to search for suspicious functions in files:

```
grep -r -E --include='*.php' 'eval|shell_exec|system|passthru|exec|popen' /var/www
```

Using the Clam AV tool to scan files:

```
sudo clamscan -r -i /var/www
```

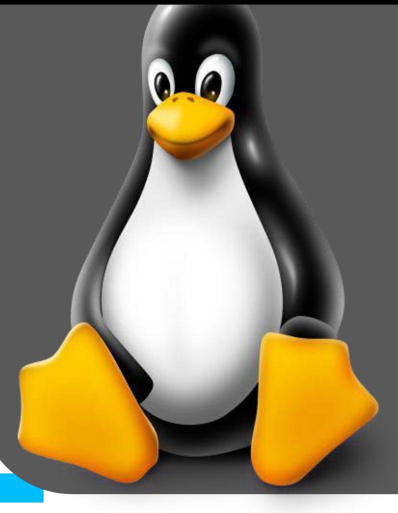
Using the awk command to check the number of suspicious functions in the files:

```
find /var/www -type f -iname "*.php" -exec awk '/eval|shell_exec|system|passthru|exec|popen/ { count++ } END { if(count > 0) { print FILENAME ": " count } }' {} \;
```

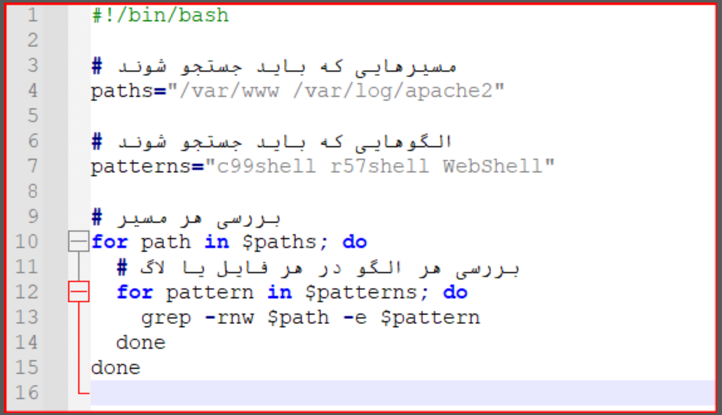
Using the find command to find very small files:

This command looks for PHP files smaller than 2000 bytes in the /var/www directory. Many web shells are small in size to load quickly.

```
find /var/www -type f -size -2000c -iname "*.php“
```



Simple sample script to find suspicious webshell activity



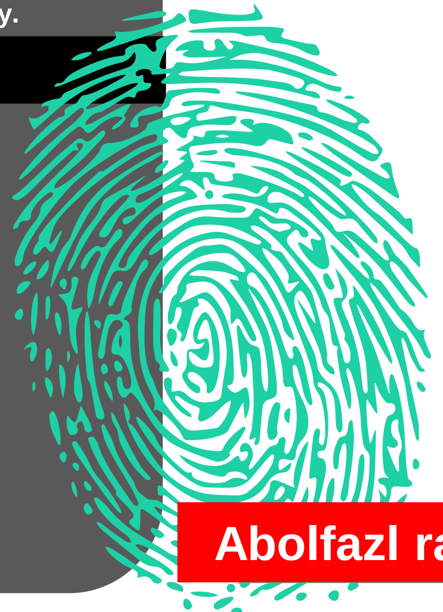
```
#!/bin/bash

# مسیرهایی که باید جستجو شوند
paths="/var/www /var/log/apache2"

# الگوهایی که باید جستجو شوند
patterns="c99shell r57shell WebShell"

# بررسی هر مسیر
for path in $paths; do
  # بررسی هر الگو در هر فایل یا لاگ
  for pattern in $patterns; do
    grep -rnw $path -e $pattern
  done
done

```



Abolfazl razipour

Webshell detection in Windows 03

“

There are several key steps to detect webshells in Windows:

1. Check files: Check system files for signs of webshells, such as files with suspicious extensions or files with unusual PHP, ASP, JSP, or PERL code.

2. Check the logs: The logs of the IIS web server or any other web server used can contain information about web shell attacks. Therefore, they must be examined and analyzed.

3. Using specialized tools: Tools like PowerShell to check files and logs, or specialized tools like IDS/IPS... and forensic tools can help detect webshells.

4. Network traffic analysis: Using network traffic analysis tools such as Wireshark, suspicious traffic can be identified and investigated.

”

Webshell detection in Windows

Suspicious requests in IIS logs

```
Get-Content C:\inetpub\logs\LogFiles\W3SVC1\u_ex180410.log | Select-String 'cmd.asp’
```

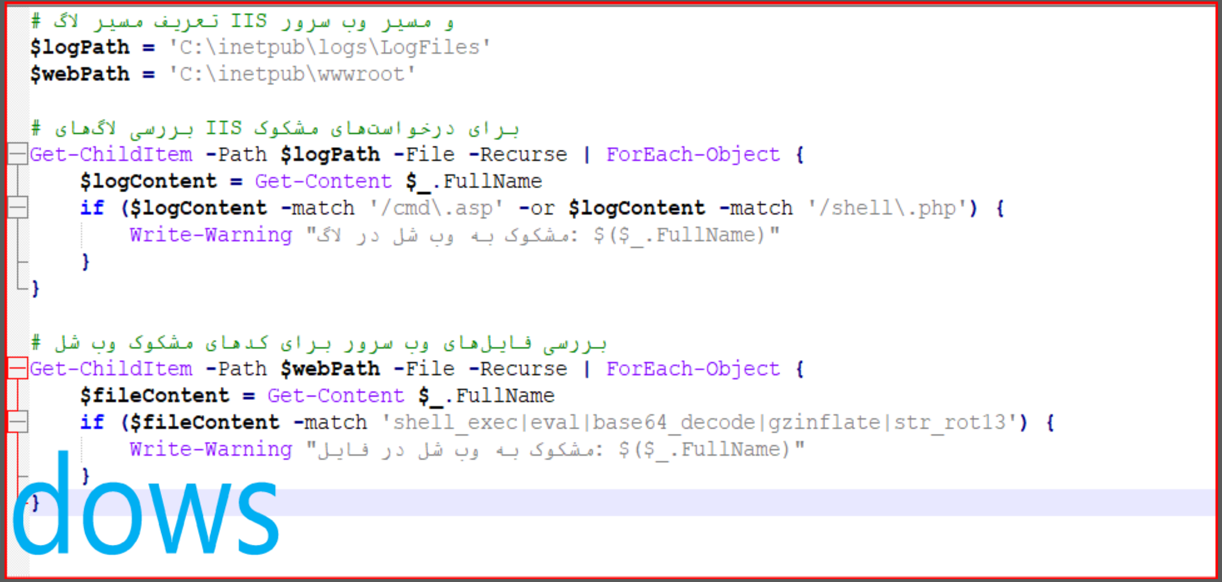
Scan files and directories: Using PowerShell commands like Get-ChildItem, you can scan files and directories where webshells may be hiding. In particular, you should pay attention to directories related to web servers and web applications.

```
Get-ChildItem -Path C:\inetpub\wwwroot\ -File -Recurse | Where-Object { $_.Extension -eq '.asp' -or $_.Extension -eq '.aspx' -or $_.Extension -eq '.php’ }
```

File Content Analysis: Using PowerShell commands like Get-Content and Select-String, you can check the content of files to make sure there is any suspicious code.

```
Get-ChildItem -Path C:\inetpub\wwwroot\ -File -Recurse | Where-Object { $_.Extension -eq '.asp' -or $_.Extension -eq '.aspx' -or $_.Extension -eq '.php' } | Get-Content | Select-String -Pattern 'shell_exec|eval|base64_decode|gzinflate|str_rot13’
```

Simple sample script to find suspicious webshell activity

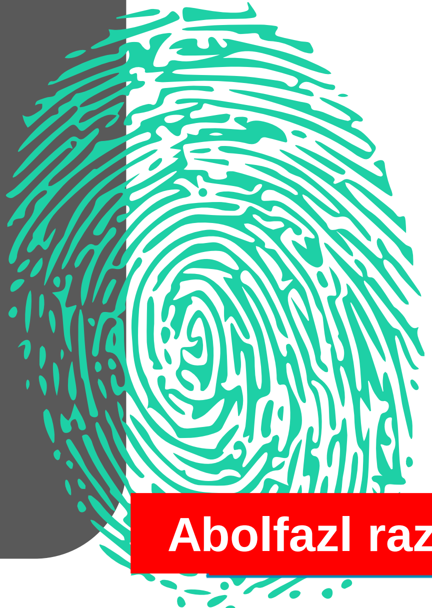


```
# تعریف مسیر لاگ IIS و مسیر وب سرور
$logPath = 'C:\inetpub\logs\LogFiles'
$webPath = 'C:\inetpub\wwwroot'

# بررسی لاگ‌های IIS برای درخواست‌های مشکوک
Get-ChildItem -Path $logPath -File -Recurse | ForEach-Object {
    $logContent = Get-Content $_.FullName
    if ($logContent -match '/cmd\.asp' -or $logContent -match '/shell\.php') {
        Write-Warning "مشکوک به وب شل در لاگ: $($_.FullName)"
    }
}

# بررسی فایل‌های وب سرور برای کدهای مشکوک وب شل
Get-ChildItem -Path $webPath -File -Recurse | ForEach-Object {
    $fileContent = Get-Content $_.FullName
    if ($fileContent -match 'shell_exec|eval|base64_decode|gzinflate|str_rot13') {
        Write-Warning "مشکوک به وب شل در فایل: $($_.FullName)"
    }
}
```

To detect webshells on Windows systems, you can use available tools and various commands. Here are some methods and commands you can use to find webshells in Windows:



Abolfazl razipour

Detect in Splunk Webshell 04

“

To detect webshells using Splunk, the following key steps can be followed:

1. Data collection: To detect webshell, you first need to send web server logs to Splunk. These logs may contain data from Apache, IIS, or any other web server.

2. Data analysis: Using Splunk's analytical capabilities, you can review the collected data. You can use SPL (Splunk Processing Language) queries to find suspicious patterns in the data.

3. Use rules and alarms: You can set up rules and alarms in Splunk to alert you when it detects suspicious patterns or webshell-specific symbols.

4. Using Splunk ES or UBA For more advanced detection, you can use Splunk security products such as Splunk Enterprise Security (ES) or Splunk User Behavior Analytics (UBA). These tools recognize more complex patterns and increase the possibility of detecting webshells.

”

Webshell detection in Splunk

One of the queries you can use in Splunk to identify webshells and suspicious activity in the Apache sensor is as follows:

01

```
index=your_webserver_index sourcetype=access_combined
| eval suspicious_extensions = "php|asp|aspx|pl|cgi"
| rex field=uri_path ".*\.(?P<file_extension>(?i)php|asp|aspx|pl|cgi)"
| search file_extension!="" http_status>=400
| stats count by src_ip, uri_path, http_status, file_extension
| sort - count
```

Adjust this query according to your needs. It works like this:

index=your_webserver_index sourcetype=access_combined: This section searches for web server logs of the type "access_combined" returns at the specified index. Replace "your_webserver_index" with your appropriate index name.
eval suspicious_extensions: This section introduces a variable to store suspicious extensions.
rex field=uri_path: This field creates a regex vector to extract the extension of the requested files.
search file_extension!="" http_status>=400: This field only selects records with suspicious extensions and HTTP status codes higher than 400.stats count by src_ip, uri_path, http_status, file_extension: It calculates requests based on source IP address, URI path, HTTP status code, and file extension.
sort - count: This section sorts the results by count in descending order.
.With this query, you can identify requests with suspicious extensions and HTTP status codes higher than 400.

To detect web shells, you can use the Splunk tool and query because to create an alert or create a dashboard:

Abolfazl razipour

Webshell detection in Splunk

Here are some more queries in Splunk that you can use to identify suspicious activity and various attacks:

Repeated failed login attempts:

```
index=your_webserver_index sourcetype=access_combined
| search http_status=401
| stats count by src_ip
| where count > THRESHOLD
| sort - count
```

Multiple requests to non-existent files (404):

```
index=your_webserver_index sourcetype=access_combined
| search http_status=404
| stats count by uri_path
| where count > THRESHOLD
| sort - count
```

Find suspicious activity by browser:

```
index=your_webserver_index sourcetype=access_combined
| rex field=user_agent "(?i)(?P<suspicious_user_agent>curl|wget|nmap|python|perl|nikto)"
| search suspicious_user_agent!=""
| stats count by src_ip, uri_path, http_status, suspicious_user_agent
| sort - count
```

To detect web shells, you can use the Splunk tool and query because to create an alert or create a dashboard:

Abolfazl razipour

Webshell detection in Splunk

Here are some more queries in Splunk that you can use to identify suspicious activity and various attacks:

Access to files with suspicious extensions

```
index=your_webserver_index sourcetype=access_combined
| rex field=uri_path "\.(?P<suspicious_extension>php[345]?|jsp|jspx|asp|aspx|cgi|pl|sh)$"
| stats count by src_ip, uri_path, http_status, suspicious_extension
| sort - count
```

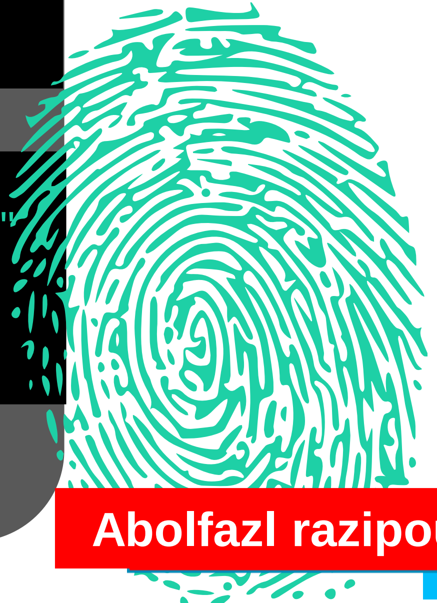
Identifying suspicious activities using specific keywords:

```
index=your_webserver_index sourcetype=access_combined
| rex field=uri_query"(?i)(?P<suspicious_keywords>(c99|c100|r57|wso|shell_exec|passthru|eval|assert|base64_decode|str_rot13|gzinflate))(\W|$)"
| search suspicious_keywords!=""
| stats count by src_ip, uri_path, http_status, suspicious_keywords
| sort - count
```

Detection of access to files with suspicious names

```
index=your_webserver_index sourcetype=access_combined
| rex field=uri_path "(?i)(?P<suspicious_filenames>(c99|r57|wso)\.(php[345]?|jsp|jspx|asp|aspx|cgi|pl|sh))$"
| search suspicious_filenames!=""
| stats count by src_ip, uri_path, http_status, suspicious_filenames
| sort - count
```

To detect web shells, you can use the Splunk tool and query because to create an alert or create a dashboard:



Abolfazl razipour

Webshell detection in Splunk

04

Here are some more queries in Splunk that you can use to identify suspicious activity and various attacks:

Detect Directory Traversal Attempts:

```
index=your_webserver_index sourcetype=access_combined
| rex field=uri_path "(?i)(?P<dir_traversal>\.\./|\.\.\\)"
| search dir_traversal!=""
| stats count by src_ip, uri_path, http_status
| sort - count
```

Identifying Brute Force Attempts:

```
index=your_webserver_index sourcetype=access_combined
| eval time=strftime(_time, "%Y-%m-%d %H:%M:%S")
| transaction src_ip maxspan=10m maxpause=2s
| where eventcount > THRESHOLD
| table time, src_ip, uri_path, http_status, eventcount
| sort - eventcount
```

Find spam requests:

```
index=your_webserver_index sourcetype=access_combined
| rex field=referer "(?i)(?P<spam_referrer>http:\/\/|https:\/\/)(?P<spam_domain>[^\/]+)"
| search spam_domain!=your_domain
| stats count by src_ip, uri_path, http_status, spam_domain
| sort - count
```

To detect web shells, you can use the Splunk tool and query because to create an alert or create a dashboard:

Abolfazl razipour

Webshell detection in Splunk

Here are some more queries in Splunk that you can use to identify suspicious activity and various attacks:

HTTP requests with unusual paths:

```
index=iis_logs cs_uri_stem=*cmd* OR cs_uri_stem=*shell* OR cs_uri_stem=*upload* OR cs_uri_stem=*exec*
OR cs_uri_stem=*admin*
| table _time, c_ip, cs_uri_stem, cs(User_Agent), sc_status
```

Searching for high volume POST requests:

```
index=iis_logs method=POST
| stats sum(sc_bytes) as total_bytes by cs_uri_stem
| where total_bytes > SOME_THRESHOLD
```

Searching for POST requests to anonymous files:

```
index=iis_logs method=POST cs_uri_stem=*.php
```

To detect web shells, you can use the Splunk tool and query because to create an alert or create a dashboard:

Abolfazl razipour

2.تشخیص فعالیت‌های مشکوک با استفاده از کلمات کلیدی مشخص:

Webshell detection in Splunk

Here are some more queries in Splunk that you can use to identify suspicious activity and various attacks:

Detecting unusual requests using structural analysis:

```
index=iis_logs
| eval length=len(cs_uri_query)
| stats avg(length) as avg stdev(length) as stdev by cs_uri_stem
| eval upper_bound=(avg + (3*stdev))
| where length > upper_bound
```

Detection of access to files with suspicious extensions:

```
index=iis_logs cs_uri_stem=*.*php* OR cs_uri_stem=*.*asp* OR cs_uri_stem=*.*aspx* OR cs_uri_stem=*.*jsp*
```

Identifying suspicious activities using specific keywords:

```
index=iis_logs cs_uri_query=*cmd.exe* OR cs_uri_query=*bash* OR cs_uri_query=*sh* OR
cs_uri_query=*python*
```

To detect web shells, you can use the Splunk tool and query because to create an alert or create a dashboard:

Abolfazl razipour

2.تشخیص فعالیت‌های مشکوک با استفاده از کلمات کلیدی مشخص:

Webshell detection in Splunk

Here are some more queries in Splunk that you can use to identify suspicious activity and various attacks:

06

Detection of access to files with suspicious names:

```
index=iis_logs cs_uri_stem=*shell* OR cs_uri_stem=*hack* OR cs_uri_stem=*exploit*
```

Multiple requests to non-existent files (404):

```
index=iis_logs sc_status=404 | stats count by cs_uri_stem | where count > 10
```

Find suspicious activity by browser:

```
index=iis_logs cs(User-Agent)=*python* OR cs(User-Agent)=*perl* OR cs(User-Agent)=*curl* OR cs(User-Agent)=*wget*
```

To detect web shells, you can use the Splunk tool and query because to create an alert or create a dashboard:

Abolfazl razipour

ELK detection in Webshell 05

“

To use ELK (Elasticsearch, Logstash, Kibana) to detect Webshell attacks, you must first send log information from your web server (whether from Apache, IIS, or another web server) to Elasticsearch. This can be done using Logstash or Beats.
After sending data to Elasticsearch, you can analyze this data using Kibana. To find signs of a Webshell attack, you can use the following methods:
1. Checking requests to files with suspicious extensions (eg .php, .asp, .aspx, .jsp and ...).
2. Check queries that have suspicious keywords such as cmd.exe, bash, sh, python, etc.
3. Check requests that go to files with suspicious names such as shell, hack, exploit, etc.
4. Investigate a large number of 404 (not found) requests, which could indicate an attempt to find vulnerable files.
5. Checking User-Agents that have suspicious names such as python, perl, curl, wget, etc.

”

Webshell detection in ELK

01

Here are some more queries in ELK that you can use to identify suspicious activity and various attacks:

Identifying suspicious activities using specific keywords and patterns:

```
GET filebeat-*/_search
{
  "query": {
    "regexp": {
      "request":
".*((cmd|passthru|eval|exec|assert|create_function|include|require)(_once)?|system|popen|show_source|phpinfo|shell_exec|base64_decode|gzinflate|chmod|mkdir|fopen|fclose|readfile)\\(.*\\).*"
    }
  }
}
```

Check for attempts to manipulate files (such as changing access levels, creating new files, opening and closing files):

```
GET filebeat-*/_search
{
  "query": {
    "regexp": {
      "request": ".*((chmod|mkdir|fopen|fclose)\\(.*\\)).*"
    }
  }
}
```

To detect web shells, you can use the ELK tool and query because to create an alert or create a dashboard:

Abolfazl razipour

Webshell detection in ELK

Here are some more queries in ELK that you can use to identify suspicious activity and various attacks:

Review of strange commands used in Webshell attacks:

```
GET filebeat-*/_search
{
  "query": {
    "regexp": {
      "request": ".*((wget|curl)\\s.*\\s-o\\s.*|php\\s.*\\s-r\\s.*).*"
    }
  }
}
```

Check the commands used to run processes in the background:

```
GET filebeat-*/_search
{
  "query": {
    "regexp": {
      "request": ".*(&\\s.*|;\\s.*).*"
    }
  }
}
```

To detect web shells, you can use the ELK tool and query because to create an alert or create a dashboard:

Abolfazl razipour

Snort detection in Webshell 06

“

Snort is an open source IDS intrusion detection system that can be used to detect web shell attacks. This tool can help detect malicious activities by analyzing network traffic and detecting suspicious patterns. You can use different rules to detect web shell attacks with Snort. These rules are designed depending on the specific patterns used in web shell attacks. Below is an example of these rules:

”

Webshell detection in Snort

Detection of access to files with suspicious names:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC access to potential webshell"; flow:to_server,established;
content:".php"; http_uri; nocase; pcre:"/\/(shell|upload|admin|c99|r57|wso)\.php/Ui"; classtype:web-application-attack; sid:1000002; rev:1;)
```

Detect suspicious command line commands in HTTP request parameters:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC potential webshell command execution"; flow:to_server,established;
content:"cmd="; http_uri; nocase; pcre:"/cmd=(id|uname|ls|cat|whoami|wget|curl|nc|netcat|ping|ifconfig|ipconfig)/Ui"; classtype:web-application-attack;
sid:1000003; rev:1;)
```

Detection of uploaded files with suspicious extensions:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC potential webshell file upload"; flow:to_server,established;
content:"Content-Disposition: form-data;"; http_header; content:".php"; http_client_body; nocase; classtype:web-application-attack; sid:1000004; rev:1;)
```

You can use different rules to detect web shell attacks with Snort. These rules depend on the specific patterns used in webshell attacks

Abolfazl razipour

Suricata detection in Webshell 07

“

Suricata is an open source intrusion detection system that can be used to analyze network traffic and detect attacks. Using Suricata rules, web shell attacks can be detected.
Below is an example of these rules:

”

Webshell detection in Suricata

Detection of access to files with suspicious names:

```
alert http $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC access to potential webshell"; flow:to_server,established; content:".php";
http_uri; nocase; pcre:"/\/(shell|upload|admin|c99|r57|wso)\.php/Ui"; classtype:web-application-attack; sid:1000002; rev:1;)
```

Detect suspicious command line commands in HTTP request parameters:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC potential webshell command execution"; flow:to_server,established;
content:"cmd="; http_uri; nocase; pcre:"/cmd=(id|uname|ls|cat|whoami|wget|curl|nc|netcat|ping|ifconfig|ipconfig)/Ui"; classtype:web-application-attack;
sid:1000003; rev:1;)
```

Detection of uploaded files with suspicious extensions:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC potential webshell file upload"; flow:to_server,established;
content:"Content-Disposition: form-data;"; http_header; content:".php"; http_client_body; nocase; classtype:web-application-attack; sid:1000004; rev:1;)
```

To detect web shell attacks with Suricata, you can use different rules. These rules depend on the specific patterns used in webshell attacks

Abolfazl razipour

Wireshark detection in Webshell 08

“

Wireshark is an advanced network traffic analysis tool. This tool can be used to detect suspicious activities such as web shell attacks. However, it should be noted that Wireshark is a network traffic analysis tool and not a complete IDS intrusion detection system. To detect web shell activity using Wireshark, you can look for specific patterns in HTTP or HTTPS traffic.

”

Webshell detection in Wireshark

Filter suspicious files:

```
http.request.uri contains ".php"
```

Filter suspicious files:

```
http.request.uri contains "c99.php"
```

To detect suspicious activity based on the browser, we can use the http.user_agent field in Wireshark. The following filter filters out HTTP requests whose User-Agent matches a specified value. For example, if we want to see all requests whose User-Agent is "Mozilla/5.0", we can use the following filter:

```
http.user_agent contains "curl" || http.user_agent contains "wget"
```

WIRESHARK

To detect web shell activity using Wireshark, you can look for specific patterns in HTTP or HTTPS traffic.

Abolfazl razipour

Modsecurity in Webshell detection

“

ModSecurity is an open source web firewall module for Apache, IIS, and Nginx that acts as a web application firewall (WAF) and can help detect and prevent webshell attacks. For this purpose, ModSecurity rules can be used.

”

Webshell detection in Modsecurity

For example, specifically for webshell detection, one of the rules we can use in ModSecurity is as follows:

```
SecRule REQUEST_FILENAME "@endsWith .php" \
   "id:1000000,phase:1,t:none,log,deny,msg:'PHP file requested'"
SecRule ARGS_POST|ARGS_GET "@rx (eval\(|base64_decode|gzinflate|str_rot13|convert_uudecode)\s*\(.*\)" \
   "id:1000001,phase:2,t:none,log,deny,msg:'Potential web shell activity'"
```

The first rule checks if the requested file name ends with .php. If so, it logs an event and rejects the request.

The second rule checks if any of the POST or GET parameters match patterns that can indicate webshell activity, including the eval, base64_decode, gzinflate, str_rot13, and convert_uudecode functions. If so, it logs an event and rejects the request.

modsecurity
Open Source Web Application Firewall

To detect web shell attacks with modsecurity, you can use different rules. These rules depend on the specific patterns used in webshell attacks

Abolfazl razipour

THANK YOU

Abolfazl razipour