

Study Guide

Identifying Web Attacks Through Logs

Created By: Igor Vieira, Ravi Raj

Module 1: Introduction and Review

Lesson 1.1: Course Introduction

This is the introductory video. The most important thing here is to understand the objective of the course. The course objectives are:

- Understanding a Web Server Log and its information
 - The student will learn how to read a Web server log and identify important information in the log.
- Identify Suspicious activity and Web Application Attacks
 - With the knowledge of the Web server logs, the student will be able to identify suspicious logs and Web application attacks based on log analysis.

Lesson 1.2: Web Applications Architecture Review – Part 1

Skills Learned From This Lesson: **Internet Components, Web Application, Web Servers, User Agents**

- Web application can either be static or dynamic
- Web application uses Client-Server model
- WWW – World Wide Web - A specific portion of the Internet as is known as “Web”
- HTML – Default Language to build Web Pages and Web Applications
- HTTP – Application protocol used to transmit messages over WWW
- Sometimes Web applications are composed of many components.
 - Web server, Database server, Application server
- User agents are software that sends HTTP messages and process the received messages.
 - Browsers are the most common user agents
 - ncat, telnet, wget are examples of other user agents
- Web servers are responsible to receive and process HTTP messages from the clients
 - Apache, Nginx and Microsoft IIS are the most used software

Lesson 1.3: Web Applications Architecture Review – Part 2

Skills Learned From This Lesson: **Internet Components, Web Application, Web Servers, User Agent, HTTP Methods, HTTP Status, TCP/IP**

- HTTP methods – commands that clients to server to perform a action
 - GET – Request an information

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- POST – Send some information as payload to be processed
-
- HTTP status codes – Used by the Web server to specify the result of the request
 - 5 categories from 1XX to 5XX.
 - 1XX – Informational
 - 2XX – Successful
 - 200 – Ok
 - 3XX – Redirection
 - 4XX – Client Error
 - 404 – Not found
 - 5XX – Server Error
- HTTP is a higher layer protocol and it relies on the lower layers like TCP and IP to work.
 - Well known TCP port for HTTP is 80 or 8080 and for HTTPS is 443 or 8443
 - TCP uses a process called 3 way handshake to establish a connection
 - The connection is established by the Operating System like Linux or Windows
 - After the 3 way handshake, the Web server and HTTP join the connection and is possible to see Web server logs
 - Web server logging starts only once the TCP 3 way handshake is established
- The laboratory will be a virtual lab and OWASP DFWA will be used as vulnerable Web application

Lesson 1.4: Web Server Logs Review - Part 1

Skills Learned From This Lesson: **Log Basics, Log Analysis, Server Logs**

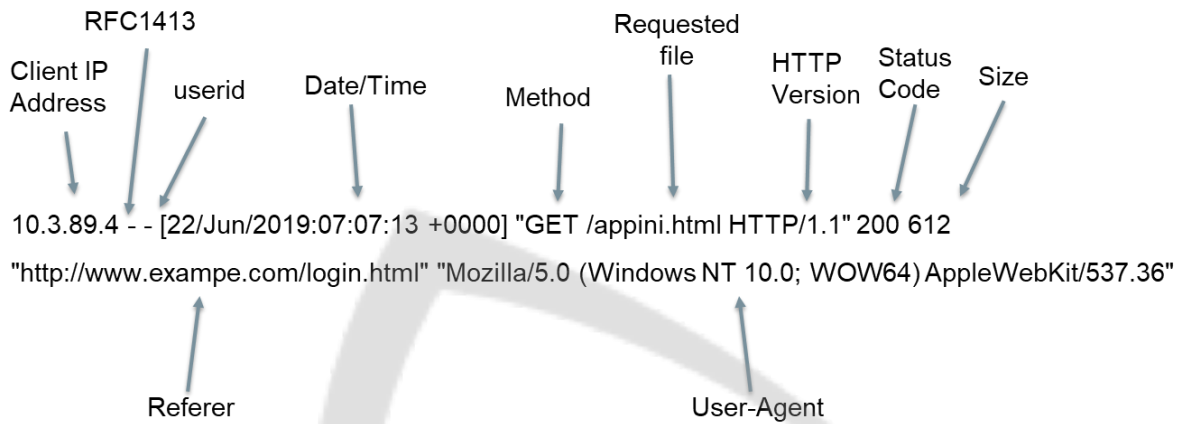
- Logs are a full written record of an event
- Logs are important and helps in many ways:
 - Troubleshooting
 - Know your application
 - Investigations
- Logs can be stored either locally or remotely
- Some questions that should be answered by the log:
 - Who did the action?
 - When the action was performed?
 - What action was performed?
- Log important fields

Brought to you by:

CYBRARY | FOR BUSINESS

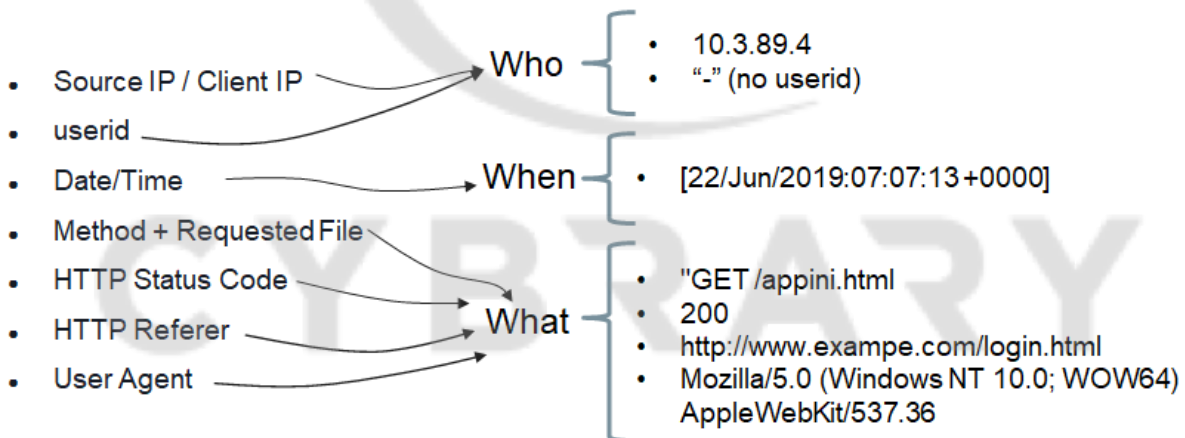
Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY



*Hyphen "-" means information is not available

- Client IP Address, RFC1413, UserID, Date/Time, Method, Requested File, HTTP Version, Status Code, Size, Referer, User Agent



Lesson 1.5: Web Server Logs Review - Part 2

Skills Learned From This Lesson: Web Servers, Log Analysis, Serve Logs

- Apache and Nginx logs are similar and are configured in a file:
 - Apache: httpd.conf

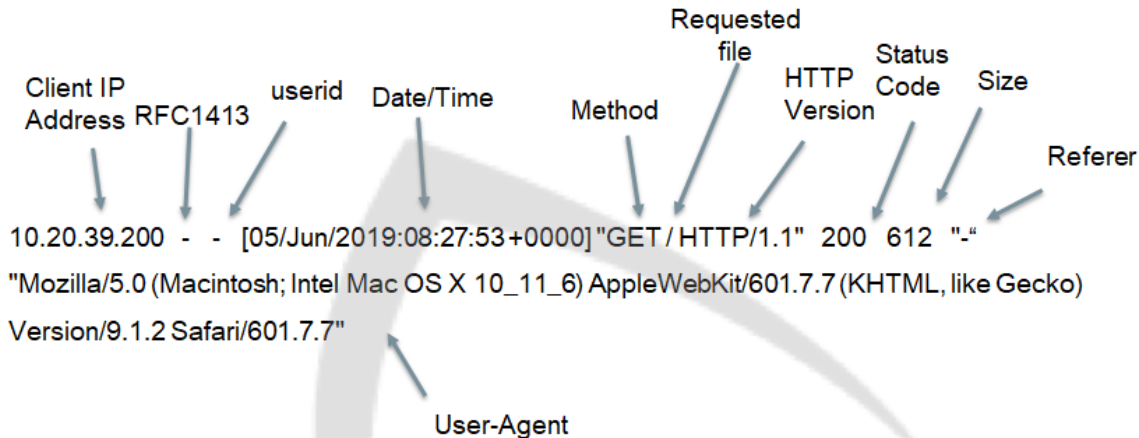
Brought to you by:

CYBRARY | FOR BUSINESS

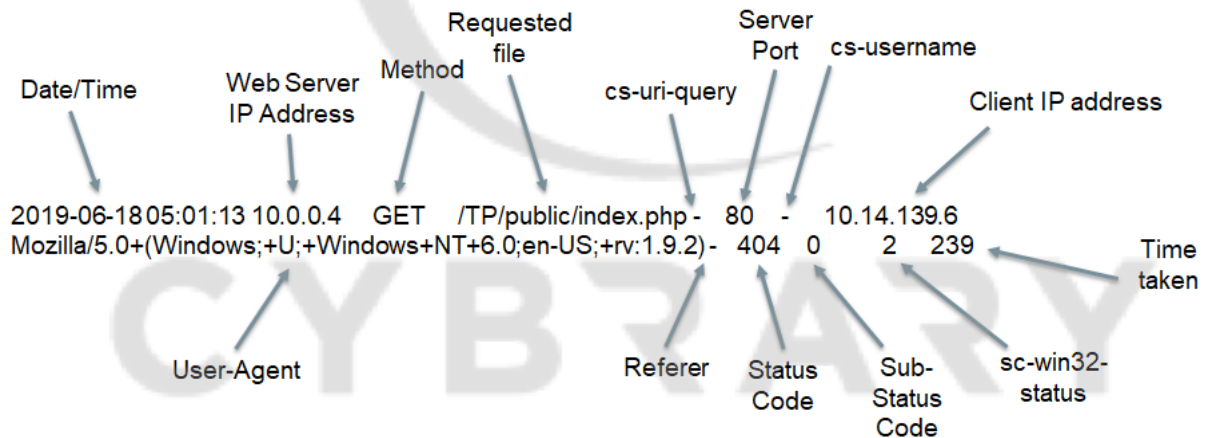
Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Nginx: nginx.conf
- Default configuration includes logging



- Microsoft IIS logs are different, but contains the key fields.
 - Default configuration includes logging



- Error logs are another source of information
 - They contains some key information
 - Can help in identifying some attacks
 - It is important to ask for the error logs too
 - Location:
 - Apache - /var/log/apache/error.log
 - Nginx - /var/log/nginx/error.log
 - IIS – %SystemDrive%\inetpub\logs\LogFiles*.log

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

Lesson 1.6: Tips and Common Issues - Part 1

Skills Learned From This Lesson: *Log Basics, Log Analysis, Log Analysis Mistakes, Log Analysis Best Practices, Crafted HTTP Requests*

NOC is worried about availability

- SOC is worried about security
- Security Incidents can happen if the component is working
- Logs = Client Request + Web Server Response
- Logs are not 100% trustable, because it can be faked
- HTTP is basic text and it is easy to generate text
- It is important to know your application
- Know your client/user behavior
- Correlate log fields

Log Field	Can be crafted in a request?	Explanation/Example
Source IP	No	Because of the TCP 3-way handshake the IP address can't be faked
Date/Time	No, it uses the current time on the server	Client does not send this field to the Server.
UserID	Yes	<code>curl -u john:pass http://example.com/</code>
Method + Requested File	Yes	It is possible to force some methods and request files.
HTTP Status Code	No	Client does not send this field to the Server. This is a response to the client request.
User Agent	Yes	<code>curl -A "nameoftheagent"</code>

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

Lesson 1.7: Tips and Common Issues - Part 2

Skills Learned From This Lesson: **Log Basics, Log Analysis, Log Analysis Mistakes, Log Analysis Best Practices**

- Know your application
- Always check for all the logs
- Busy servers generate more logs
- The logs could be wrong
- The logs could be incomplete
- Be careful not to interpret wrong information from logs
- Always remember to document the investigation

Module 2: Identifying Web Attacks with Logs

Lesson 2.1: Web Application Attacks Review

Skills Learned From This Lesson: **URL Components, OWASP Top 10, Web application architecture**

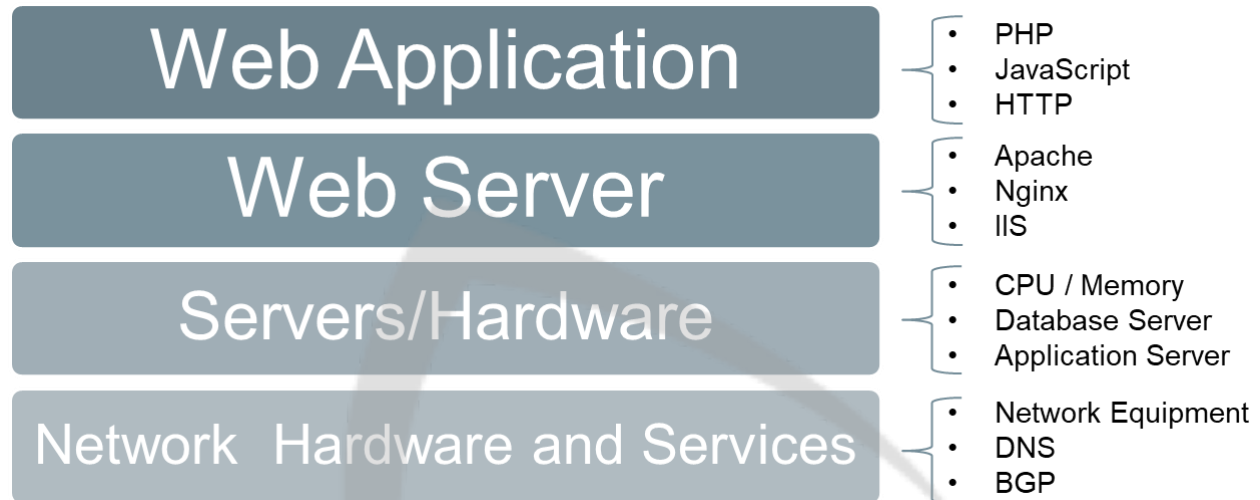
- **Learning Objectives**
 - Understand the difference between infrastructure and application attacks: There are various points which can be attacked.
 - Introduce the OWASP Top 10 Project: Critical points to attack a Web application. The version used in this course was released in 2017, while this course was created and some of the critical attack vectors are discussed.
 - Review common Web application attacks: Common weakness in the Web applications that are exploited.
 - Understand URL components: There are various components in a URL. We need to understand the components that can be leveraged for an attack.
- **Web Application vs Infrastructure Attacks:**

Brought to you by:

CYBRARY | FOR BUSINESS

*Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.*

CYBRARY



- **Web application** is the top layer, Scripting languages like Javascript or PHP and protocols like HTTP are used here.
- **Web Servers** comprises the next layer and consists of the Web servers like Apache, Nginx and IIS which hosts a Web application.
- **Servers/Hardware** forms the next layer. It is the physical machine with a Windows or LINUX based OS which would be hosting the Web application. A virtual machine might also be used to host Web applications. Database Server and Application Server too can be added to this layer.
- **Network Hardware and Services** is the last layer. It makes the communication possible between the servers (hosting Web application) and the client.

Each component of the architecture discussed above can be attacked resulting in a compromise of the Web application. All the bottom layers comprise the infrastructure. We will focus on Web application layer in the course.

- **Web Application Architecture**

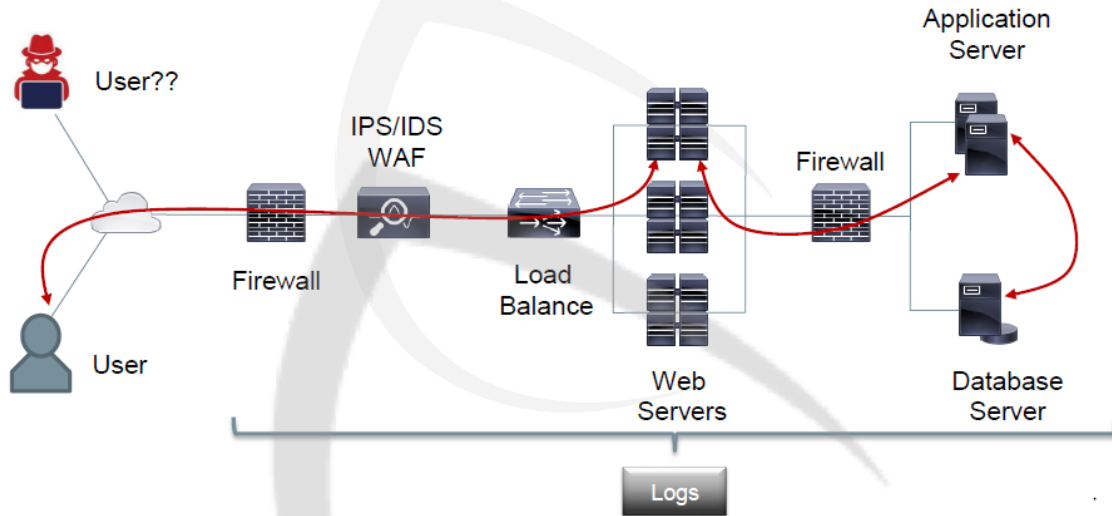
Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

❖ Multi-Layer/Balanced



- Other designs are also possible but they should be similar to the above architecture.
- To access a Web page user will send the request to the Web server and the Web server will access the other components. This entire infrastructure comprises different types of logs which could help to identify an attack.
- To distinguish between a malicious user and a legitimate user, we should have logs and able to understand them.
- **Web Application Attacks**
 - **Client-side Attacks:** The end user is targeted in these attacks.
 - **Server-side Attacks:** The server hosting the Web application is being targeted in these attacks. The aim is to compromise the Web application or use it to compromise the users accessing it.
- **Vulnerability:** Weakness in a Web application, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source. An attacker exploits a vulnerability to compromise a Web application. All the vulnerabilities comprise the attack surface.
- **Risk:** Risk in Cyber security is the chance of data exposure or loss occurring due to illegitimate access from an attacker.
- **Target:** For us, in the course, it will be Web Servers and Web Applications.
- **Attack:** An action that attempts to exploit vulnerability or cause an impact to a Web application.

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- **What are the most common Web application attacks?**



- **What are the most common Web attacks:** What are the weak points that an attacker might exploit?
- **Attacks exploit vulnerabilities:** An attacker will try to exploit a vulnerability in the Web application or infrastructure to compromise it.
- **Most common vulnerabilities are the most common attacks**
- **What are the most common vulnerabilities:** Understand the most exploited vulnerabilities in Web application.
- **OWASP Top 10:** Open Web Application Security Project catalogues the top 10 vulnerabilities in the Web applications. First version was launched in 2003 and the newest version is launched in 2017 which will be discussed in this course.

- **Comparison between OWASP Top 10 – 2013 and OWASP Top 10 – 2017**

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	⊗	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	⊗	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- In this course, we will discuss,
 - A1 - Injection
 - A2 - Broken Authentication
 - A3 - Cross-Site-Scripting
 - A5 - Security Misconfiguration
 - A9 - Using components with known Vulnerabilities.
- **URL - Uniform Resource Locator**

It is a type of Universal Resource Identifier. User agent requests resources from Web servers using URL. Each Web Application has 1 resource locator. This enables a Web server to host multiple Web applications. Also known as Web address and has multiple parts:

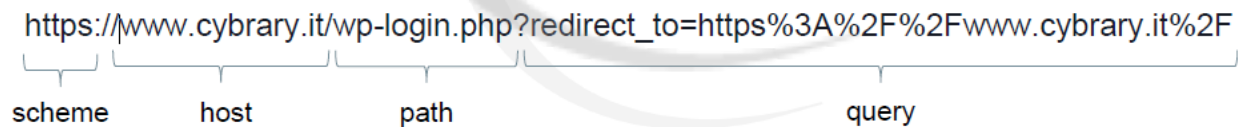
URI = scheme://host:port/path?query

scheme: Identifies the protocol

host: Domain being addressed, it may be followed by the Port number to direct or send the client request.

path: Identifies the resource the user wants to access.

query: Used to query or send information to the Web server.



- It is important to understand that most of the attacks are performed using the path or the query component.
- **URL Encode**
 - URLs can be sent over the Internet using only the ASCII character set, to maintain consistency.
 - URL encoding converts characters into a format that can be transmitted over the Internet.
 - URL encoding replaces unsafe ASCII characters with a "%" followed by 2 hexadecimal digits.
 - E.g. URLs cannot contain spaces. Normally a space is replaced by a %20 characters.

User Input in browser of /login.php?username=my username translates to URL of /login.php?username=my%20username

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Encoding is also used to transfer different write system.
- It is common to see attacks using encoded URLs. Although a % in the URL doesn't mean always a malicious request. The below Cybrary link contains % but isn't malicious.
https://www.cybrary.it/wp-login.php?redirect_to=https%3A%2F%2Fwww.cybrary.it%2F
- The below diagram shows a typical SQL attack. It can be identified by the multiple SQL words in the encoded message like SELECT, UNION, schema etc.

```
10.1.0.10 - - [06/Jun/2019:06:01:23 -0400] "GET /?Pjzq=3481%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E%27%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20..%2F..%2F..%2Fetc%2Fpasswd%27%29%23 HTTP/1.1" 200 4785 "-" "sqlmap/1.3.4#stable (http://sqlmap.org)"
```



```
10.1.0.10 - - [06/Jun/2019:06:01:23 -0400] "GET /?Pjzq=3481 AND 1=1 UNION ALL SELECT 1,NULL,'<script>alert("XSS")</script>',table_name FROM information_schema.tables WHERE 2>1-/**/; EXEC xp_cmdshell('cat ../..../etc/passwd')# HTTP/1.1" 200 4785 "-" "sqlmap/1.3.4#stable (http://sqlmap.org)"
```

- A common behavior in today's Web page is typically below where various components of a Web page might be requested. In below request various components of a Web page are being requested.



Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- 10.1.0.10 - - [22/Jun/2019:03:18:48 +0000] "GET /login.php" HTTP/1.1" 200 571 "-"
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/63.0.3239.108 Safari/537.36"
- 10.1.0.10 - - [22/Jun/2019:03:18:48 +0000] "GET /welcome.jpg" HTTP/1.1" 200 501 "-"
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/63.0.3239.108 Safari/537.36"
- 10.1.0.10 - - [22/Jun/2019:03:18:48 +0000] "GET /logo.gif" HTTP/1.1" 200 201 "-"
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/63.0.3239.108 Safari/537.36"
- **Post-Assessment Question**
 - Considering basic Web infrastructure, only Web servers are susceptible to attacks. This statement is:
True
False
Answer: The statement is false as various components of Web application might be targeted in the attack.
 - Which of these vulnerabilities are present in OWASP Top 10 Project 2017?
i.DNS Poisoning
ii.Injection
iii.Syn-Flood
iv.Security Misconfiguration
v.DDOS
Answer: ii.Injection and iv.Security Misconfiguration. Check OWASP Top 10 for more information.
 - Web requests with % in it are always malicious.
This information is:
True
False
Answer: The statement is false as "%" can be used to transfer a different writing system or unsupported characters.

Lesson 2.2: Vulnerability Scans

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

Skills Learned From This Lesson: **Vulnerability Scanners, Log Analysis, Web application attacks**

- **Learning Objectives**
 - Review vulnerability scans and the security tools used to conduct vulnerability scans: Know about various tools which are used to conduct a vulnerability scan
 - Identify vulnerability scans through log analysis: We will try to identify from logs if a vulnerability scan is being conducted
- **Vulnerability:** Weakness in information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source. It is referred in OWASP TOP 10 via A9: Using Components with Known Vulnerabilities. E.g our Web application may be vulnerable if we are using older version of PHP in it.
- **Understanding the vulnerabilities** the next question that arises is, how to know that our Web application has vulnerabilities. We can review the application source code but that will be a very time consuming process. Another way is to use vulnerability scanners to identify vulnerabilities.
- **Vulnerability scanners**
 - Launches some known attacks to check application response: We can use a vulnerability scanner to identify a vulnerability in the Web application. Based on the response from the Web application, we can identify if there exists a vulnerability in the Web application.
 - Used by attackers and non-attackers to find weaknesses on their targets: It can be used by attackers and defenders alike to identify the vulnerabilities. An attacker would use the scanner to exploit the vulnerabilities while a defender would use the scanner to identify the vulnerabilities and fix them.
 - Can be used to test your security tools like IDS, IPS and WAF: It can be used to test the defensive capabilities of the tool deployed in our Web application setup or infrastructure.
 - Vulnerability scan may not be categorized as an attack in some organizations, rather only as a pre-attack method to find more information about the target.
 - Open source or paid software options available for vulnerability scanners:

Open Source	Paid
Nikto	Burp
Zed	Accunetix
W3af	AppScan
SqlMap	Netsparker

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Some of the scanners like sqlmap are only for testing a specific vulnerability . It is used only for testing SQL Injection.
- The easiest way to identify a vulnerability scanner is to check the user agent in the log generated.
- From the below log, it is clear that the user agent sending requests is the vulnerability scanner, sqlmap.

```
10.1.0.10 - - [06/Jun/2019:06:01:23 -0400] "GET /?Pjq=3481%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E%27%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%r28%27cat%20.%2F..%2F..%2Fetc%2Fpasswd%27%29%23 HTTP/1.1" 200 4785 "-" "sqlmap/1.3.4#stable (http://sqlmap.org)"
```

- User agents are easy to detect, but can be faked or crafted and thus, they are not always trustable to detect a vulnerability scan against our Web application, many scanners provide options to change the user agent name.
- Sometimes weird user agents can be seen. For example: programming languages (python-requests/2.18.4).
- **Nikto Example:**

```
10.1.0.10 - - [11/Jul/2019:05:51:16 -0400] "GET /dvwa/common/about HTTP/1.1" 404 215 "-" "Mozilla/5.0 (Nikto/2.1.6) (Evasions:None) (Test:007190)"
10.1.0.10 - - [11/Jul/2019:05:51:16 -0400] "GET /dvwa/master.xml HTTP/1.1" 404 213 "-" "Mozilla/5.0 (Nikto/2.1.6) (Evasions:None) (Test:007191)"
```

- From the agent used it is clear that Nikto is being used.
- Also if we check the logs, it will be clear that there are a large number of requests generated from the scanner

```
+ 7914 requests: 0 error(s) and 34 item(s) reported on remote host
+ End Time:      2019-07-11 05:51:17 (GMT-4) (38 seconds)
-----
+ 1 host(s) tested
```

- We would expect a large number of requests generated, as a scanner would try to scan for all the vulnerabilities, but a large number of requests should show return error code. This nature is demonstrated in the example below where most of the requests are returning the error code. First command output shows the number of requests with Nikto

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

as agent and the second command output shows the number of requests with Nikto as agent and returning the error code. Most of the requests return the error code.

```
root@owaspbwa:~# cat /var/log/apache2/access.log | grep "Nikto/2.1.6" | wc -l
7852
root@owaspbwa:~# cat /var/log/apache2/access.log | grep "Nikto/2.1.6" | grep " 404 " | wc -l
7533
```

- The important things to note are the user agent, Number of requests generated and the number of requests returning the error code.
- Analyzing from the log below, the log generated is from a Nikto scanner. The logs at both side of the arrow are from Nikto scanner. Thus a request can easily be forged to reflect a different user agent.

```
10.1.0.10 - - [30/Jul/2019:04:29:48 -0400] "GET /shell?cat%20/etc/passwd HTTP/1.1" 404 203 "-" "Mozilla/5.0 (Nikto/2.1.6) (Evasions:None) (Test:007084)"
10.1.0.10 - - [30/Jul/2019:04:29:48 -0400] "GET /sitecore/shell/webservice/service.asmx HTTP/1.1" 404 236 "-" "Mozilla/5.0 (Nikto/2.1.6) (Evasions:None) (Test:007132)"
10.1.0.10 - - [30/Jul/2019:04:29:49 -0400] "GET /shell?cat+/etc/hosts HTTP/1.1" 404 203 "-" "Mozilla/5.0 (Nikto/2.1.6) (Evasions:None) (Test:007235)"
10.1.0.10 - - [30/Jul/2019:05:19:21 -0400] "GET /cgi-bin/rwwwshell.pl HTTP/1.1" 404 218 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:19:24 -0400] "GET /cgi-bin/dasp/fm shell.asp HTTP/1.1" 404 223 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:19:25 -0400] "GET /cgi-bin/jailshell HTTP/1.1" 404 215 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:19:25 -0400] "GET /cgi-bin/noshell HTTP/1.1" 404 213 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:08 -0400] "GET /App Config/sitecore/shell/sitecore.version.xml HTTP/1.1" 404 244 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:08 -0400] "GET /sitecore/shell/WebService/Service.asmx HTTP/1.1" 404 236 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:11 -0400] "GET /shell?cat%20/etc/passwd HTTP/1.1" 404 203 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:11 -0400] "GET /sitecore/shell/webservice/service.asmx HTTP/1.1" 404 236 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:12 -0400] "GET /shell?cat+/etc/hosts HTTP/1.1" 404 203 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
```

- **Identifying a Vulnerability scan:**
 - Check the user agent – can be vulnerability scanner software or something else weird
 - If there are many requests in a short amount of time then it points to a vulnerability scan.
 - If there are lots of error returned for the requests or weird requests like PHP request on a page that doesn't have PHP. better understanding of the application helps in recognizing the attack.

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Requests directed to admin or configuration pages are also a signature, as they are critical points to be exploited.
- Commands or OS words in requests: ping, cat, shell, admin, config, and others are suspicious too.
- **Post-Assessment Question**
 - Consider this scenario: The NOC team asked you to check the behavior of a Web server because they found an increase in the number of weird requests and 404 errors. The NOC team sent a print of the Web server CPU usage and says it looks normal.

```
top - 04:38:11 up 1 day, 14:32, 3 users, load average: 0.03, 0.11, 0.10
Tasks: 113 total, 1 running, 112 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1026100k total, 649476k used, 376624k free, 173424k buffers
Swap: 397304k total, 105828k used, 291476k free, 170652k cached

  PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
  1 root  20  0 2800 1028 676 S  0.0  0.1  0:00.42 init
  2 root  20  0  0  0  0 S  0.0  0.0  0:00.00 kthreadd
  3 root  RT  0  0  0  0 S  0.0  0.0  0:00.00 migration/0
  4 root  20  0  0  0  0 S  0.0  0.0  0:00.75 ksoftirqd/0
  5 root  RT  0  0  0  0 S  0.0  0.0  0:00.00 watchdog/0
  6 root  20  0  0  0  0 S  0.0  0.0  0:00.07 events/0
  7 root  20  0  0  0  0 S  0.0  0.0  0:00.00 cpuset
  8 root  20  0  0  0  0 S  0.0  0.0  0:00.00 khelper
  9 root  20  0  0  0  0 S  0.0  0.0  0:00.00 netns
 10 root  20  0  0  0  0 S  0.0  0.0  0:00.00 async/mgr
 11 root  20  0  0  0  0 S  0.0  0.0  0:00.00 pm
 12 root  20  0  0  0  0 S  0.0  0.0  0:00.13 sync_supers
 13 root  20  0  0  0  0 S  0.0  0.0  0:00.17 bdi-default
 14 root  20  0  0  0  0 S  0.0  0.0  0:00.00 kintegrityd/0
 15 root  20  0  0  0  0 S  0.0  0.0  0:02.00 kblockd/0
 16 root  20  0  0  0  0 S  0.0  0.0  0:00.00 kacpid
 17 root  20  0  0  0  0 S  0.0  0.0  0:00.00 kacpi_notify
 18 root  20  0  0  0  0 S  0.0  0.0  0:00.00 kacpi_hotplug
 19 root  20  0  0  0  0 S  0.0  0.0  0:00.00 ata/0
 20 root  20  0  0  0  0 S  0.0  0.0  0:00.00 ata aux
```

You ask for more information about the Web server and the Web server logs. They say it is Apache server, and not WordPress site, despite the WordPress requests in the logs.

-
-

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

```
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/uploads/dump.sql HTTP/1.1" 404 234
 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/plugins/pods/sql/dump.sql HTTP/1.1"
 404 243 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/plugins/simplemap/dump.sql HTTP/1.1"
 404 244 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/plugins/simplemap/classes/error_log
 HTTP/1.1" 404 253 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/plugins/emailbuddy/db.sql HTTP/1.1"
 404 243 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/uploads/ HTTP/1.1" 404 226 "-" "Moz
 illa/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/uploads/ HTTP/1.1" 404 226 "-" "Moz
 illa/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/plugins/wpmu-dev-post-votes/db.sql
 HTTP/1.1" 404 252 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/plugins/wpvotes/db.sql HTTP/1.1" 40
 4 240 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-content/plugins/post-voting/db.sql HTTP/1.1"
 404 244 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-admin/wp-login.php HTTP/1.1" 404 228 "-" "M
 ozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
10.1.0.10 - - [30/Jul/2019:05:20:07 -0400] "GET /wordpress-login.php HTTP/1.1" 404 219 "-" "Mozilla/5.
 0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox/39.0"
```

- Analyzing from the above logs, it is clear that the same IP is making multiple requests at the same time and 404 error logs are returned for most of the requests.
- Based on the scenario presented, as a SOC Analyst, answer the questions:
 - 1) Which IP address is causing the trouble?
 - a) 10.2.0.101
 - b) Can't say, need more logs.
 - c) 10.1.0.10Answer is c
 - 2) What behavior did you identify in the log?
 - a) SYN Scan
 - b) SQL Injection
 - c) Vulnerability Scan
 - d) It is not an AttackThe answer is c.

From SOC perspective, based on the above logs we can say the below:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

When Who What

On July 30th, around 5:20 the IP address 10.1.0.10 did a vulnerability scan trying to find a vulnerable WordPress web application on our infrastructure.

The best action would be to block the IP as it is performing a Vulnerability scan.

Lesson 2.3: Brute Force Attack

Skills Learned From This Lesson: **Log analysis, Brute force attack, GET and POST requests**

- A brute force attack is when someone tries to get access to our Web page. Achieved through trying multiple logins.
- Brute force attacks uses dictionary or leaked information. Dictionary is a list of common used passwords. Attacker might use the leaked information for cracking a password and gain illegitimate access. So, it is important to change password immediately for the account that has been informed about leaked password and also to be safe.
- For Web applications, HTTP forms are commonly attacked targets. HTTP form may use GET or POST requests.
- Brute Force attacks are part of the OWASP Top10 A2: Broken Authentication
- We will use tools THC-Hydra and Burp Community Edition for performing attacks on our DVWA lab.
- Analyzing from logs: Below are logs from a brute force attack being attempted. We will try to identify the key indicators which will point to a brute force attempt on the Web application.

Brought to you by:

CYBRARY | FOR BUSINESS

*Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.*

Brute Force Attacks

- 10.1.0.10 -- [16/Jul/2019:05:42:23 -0400] "GET /dvwa/vulnerabilities/brute/?username=admin&password=1234567&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
- 10.1.0.10 -- [16/Jul/2019:05:42:24 -0400] "GET /dvwa/vulnerabilities/brute/?username=admin&password=1234567&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/?username=admin&password=1234567&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
- 10.1.0.10 -- [16/Jul/2019:05:42:28 -0400] "GET /dvwa/vulnerabilities/brute/?username=admin&password=abc123&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
- 10.1.0.10 -- [16/Jul/2019:05:42:29 -0400] "GET /dvwa/vulnerabilities/brute/?username=Pablo&password=abc123&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
- 10.1.0.10 -- [16/Jul/2019:05:42:31 -0400] "GET /dvwa/vulnerabilities/brute/?username=admin&password=nicole&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

- Multiple username and password combinations are being attempted. Also, it is important to notice that these requests are being generated from same IP and within very short span which isn't possible for a human user.

- Another log analysis:

```
10.1.0.10 -- [14/Jul/2019:13:06:30 -0400] "GET /dvwa/vulnerabilities/brute/?username=Pablo&password=111111&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
10.1.0.10 -- [14/Jul/2019:13:06:31 -0400] "GET /dvwa/vulnerabilities/brute/?username=Pablo&password=iloveu&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
10.1.0.10 -- [14/Jul/2019:13:06:33 -0400] "GET /dvwa/vulnerabilities/brute/?username=Pablo&password=000000&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
10.1.0.10 -- [14/Jul/2019:13:06:34 -0400] "GET /dvwa/vulnerabilities/brute/?username=Pablo&password=michelle&Login=Login HTTP/1.1" 200 1401 "http://10.2.0.101/dvwa/vulnerabilities/brute/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- From the above log we can see that someone is trying to login using username, Pablo. Behavior is common to a brute force attempt. Multiple requests are generating from the same IP within a short span of time.
- Analyzing the Brute force attempt from POST requests:
 - 10.1.0.10 - - [14/Jul/2019:09:40:40 -0400] "POST /dvwa/login.php HTTP/1.0" 302 - "-" "Mozilla/5.0 (Hydra Proxy)"
 - 10.1.0.10 - - [14/Jul/2019:09:40:40 -0400] "POST /dvwa/login.php HTTP/1.0" 302 - "-" "Mozilla/5.0 (Hydra Proxy)"
 - 10.1.0.10 - - [14/Jul/2019:09:40:40 -0400] "POST /dvwa/login.php HTTP/1.0" 302 - "-" "Mozilla/5.0 (Hydra Proxy)"
 - 10.1.0.10 - - [14/Jul/2019:09:40:41 -0400] "POST /dvwa/login.php HTTP/1.0" 302 - "-" "Mozilla/5.0 (Hydra Proxy)"
 - 10.1.0.10 - - [14/Jul/2019:09:40:41 -0400] "POST /dvwa/login.php HTTP/1.0" 302 - "-" "Mozilla/5.0 (Hydra Proxy)"
 - 10.1.0.10 - - [14/Jul/2019:09:40:41 -0400] "POST /dvwa/login.php HTTP/1.0" 302 - "-" "Mozilla/5.0 (Hydra Proxy)"
 - 10.1.0.10 - - [14/Jul/2019:09:40:41 -0400] "POST /dvwa/login.php HTTP/1.0" 302 - "-" "Mozilla/5.0 (Hydra Proxy)"
 - 10.1.0.10 - - [14/Jul/2019:09:40:41 -0400] "POST /dvwa/login.php HTTP/1.0" 302 - "-" "Mozilla/5.0 (Hydra Proxy)"
- In a POST request, the username and the passwords won't be visible as they are contained in the payload.
- Hydra Proxy is being used as it is clear from the User Agent field. Hydra is a well known tool used to perform the Brute force attacks. Another important point to note is the large number of requests originating from the same IP and the attempts on the login page.
- Analyzing the Brute force attacks from the below logs:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- 10.1.0.10 - - [14/Jul/2019:12:49:06 -0400] "POST /dvwa/login.php HTTP/1.1" 302 20 "http://10.2.0.101/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
 - 10.1.0.10 - - [14/Jul/2019:12:49:14 -0400] "POST /dvwa/login.php HTTP/1.1" 302 20 "http://10.2.0.101/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
 - 10.1.0.10 - - [14/Jul/2019:12:49:18 -0400] "POST /dvwa/login.php HTTP/1.1" 302 20 "http://10.2.0.101/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
 - 10.1.0.10 - - [14/Jul/2019:49:22: -0400] "POST /dvwa/login.php HTTP/1.1" 302 20 "http://10.2.0.101/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
 - 10.1.0.10 - - [14/Jul/2019:12:49:25 -0400] "POST /dvwa/login.php HTTP/1.1" 302 20 "http://10.2.0.101/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
- Here we have POST but the user agent looks normal. It can be a genuine attempt from a user trying to login. User types a username and a password, then the page refreshes. But, it is highly unlikely for someone to type the username and password in very short span, which points to a brute force attempt.
- Indicators for Brute force attempt:
 - Many requests to login pages in a short span of time: Same IP trying multiple login attempts in a very short span which might not be possible for a human user.
 - GET = Different users or passwords: Different combinations being tried to brute force.
 - POST = Many requests in a short amount of time or check other logs. Check if the attack or attempt is on a login page and the requests are sent within short intervals.
 - User Agent: Check if the user agent is a tool used to perform Brute force attacks.
 - **Post-Assessment Question**

"You can always identify a Brute Force attack by analyzing only the user agent"
Is this statement true or false?
Answer: It is false as a user agent can be faked easily.
 - **Post-Assessment Question**

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

Analyze the Web log below and identify the source IP of the attacker, type of attack, and what is the attacker is trying to do.

Analyze the web log below and identify the IP source of the attacker, type of attack, and what the attacker is trying to do.

```
10.1.0.10 - - [16/Jul/2019:05:43:58 -0400]
"GET /example.com/?username=administrator&password=iloveu&Login=Login HTTP/1.1" 200 1401
 "-" Mozilla/5.0 (X11; Linux x86_64; rv:60.0)
10.1.0.10 - - [16/Jul/2019:05:44:08 -0400]
"GET /example.com/?username=administrator&password=michelle&Login=Login HTTP/1.1" 200 1401
 "-" Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
10.1.0.10 - - [16/Jul/2019:05:44:19 -0400]
"GET /example.com/?username=administrator&password=tigger&Login=Login HTTP/1.1" 200 1401
 "-" Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
10.1.0.10 - - [16/Jul/2019:05:44:30 -0400]
"GET /example.com/?username=administrator&password=sunshine&Login=Login HTTP/1.1" 200 1401
 "-" Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

Answer:

```
Source IP: 10.1.0.10
Attack: Brute Force
Attacker is trying to obtain the administrator password
```

- Common Indicators for a Brute Force Attack:
 - User Agent: Check if the user agent being used is the one commonly used for Brute force attack
 - Many requests: Multiple requests are being sent in a short span of time.
 - Login Web Pages: The page where this unusual behaviour is observed is a login page.
 - Administrator username: The Administrator username is being attempted for cracking.

Lesson 2.4 - SQL Injection

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

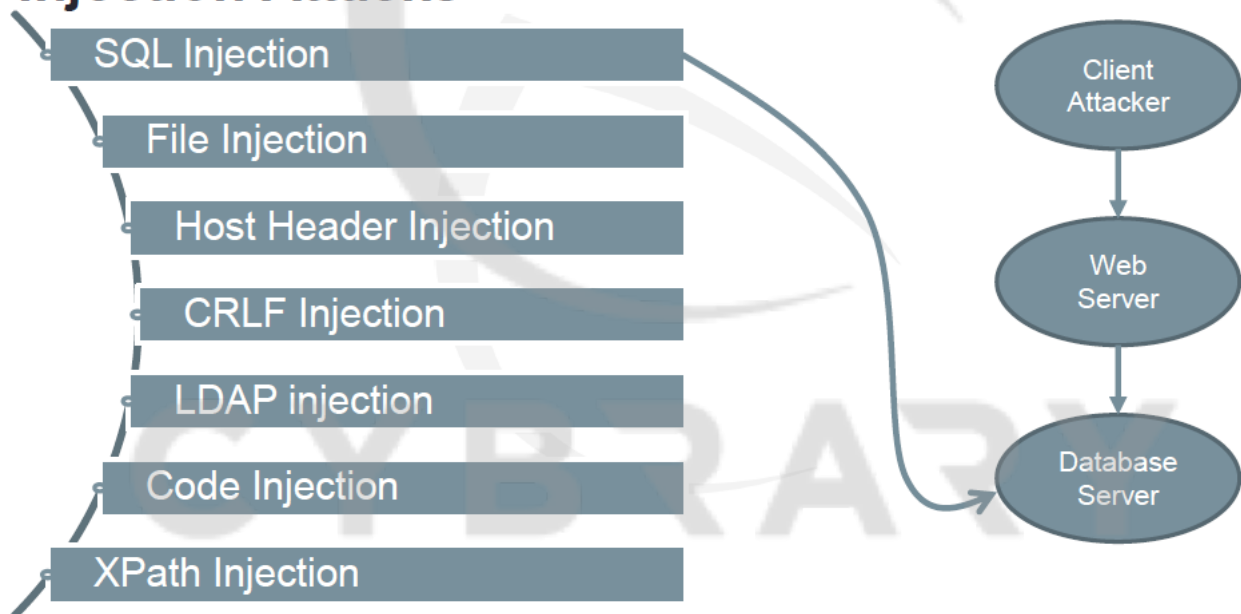
Skills Learned From This Lesson: **Log analysis, Injection attack, SQL injection attack**

- **Injection Attacks:**

“Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or a query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.” It is part of A1:2017-Injection attack in OWASP 2017. It continues to occupy the top spot as the most critical vulnerability from OWASP Top 10-2013 to the latest version of OWASP Top 10 in 2017. Web server doesn't care about the malicious requests and may execute a malicious request.

- **Various Injection attacks:**

Injection Attacks



SQL injection attack directly impacts the Database server. Database can execute some malicious commands impacting the integrity of the database server and ultimately the Web application. SQL injection is considered a critical vulnerability.

- Important point to note about the SQL injection attack:
 - Uses Structured Query Language(SQL), else, database might not process it.

Brought to you by:

CYBRARY | FOR BUSINESS

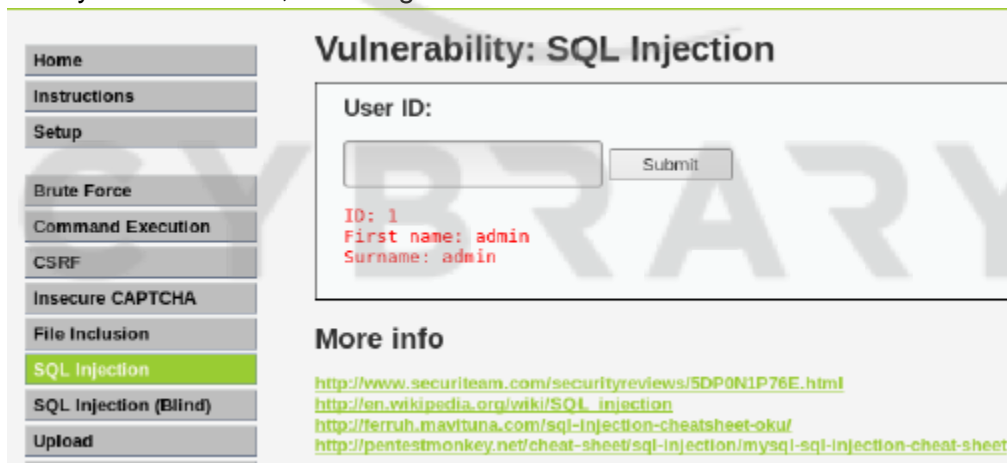
Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Caused by lack of proper user input validation, like, allowing of special characters in the form.
- It has been around since 1998 and is common in legacy applications.
- It is a Server-Side Attack and has multiple types like Blind, Classic, Union-Based and Error Based attacks.
- Example of SQL injection on DVWA.



If we try the number 1 , we can get the admin user information.



Following request is made to the Web server:

<http://10.2.0.101/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#>

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

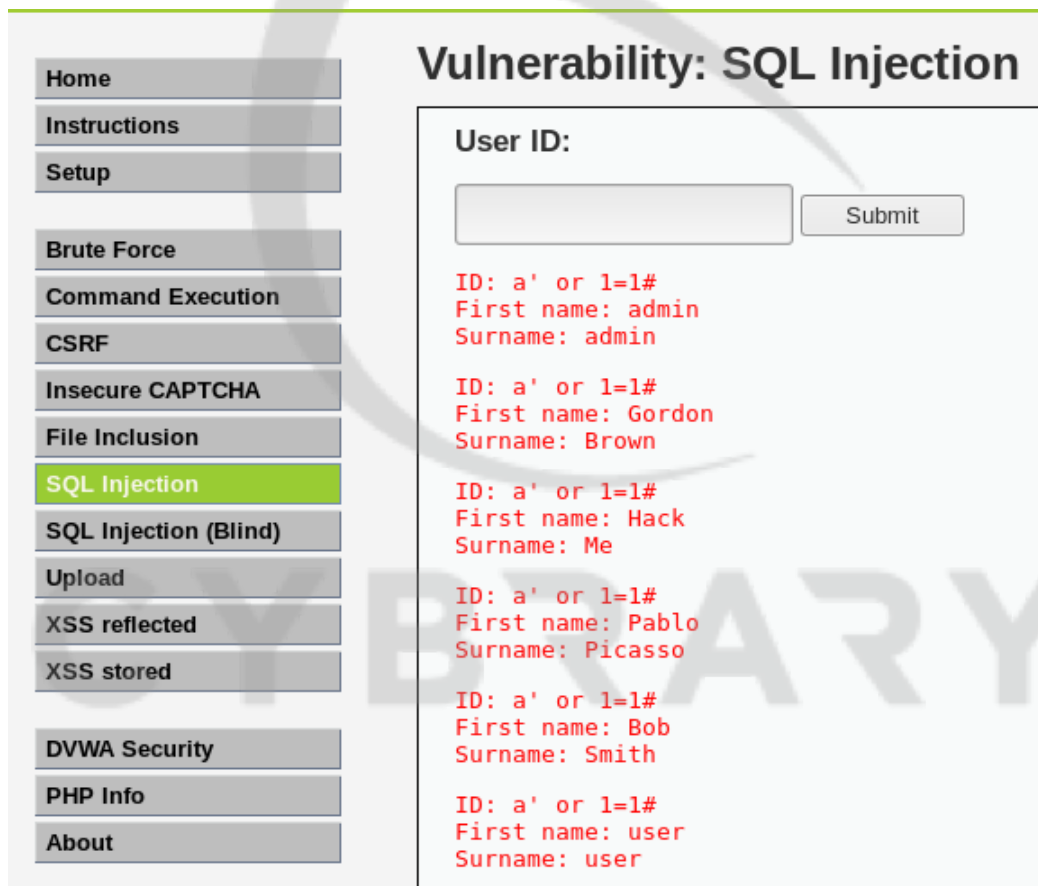
CYBRARY

- What happens if we send the request, a' or 1=1# in the UserID field.



User ID:

The query being passed in the UserID field is always true and outputs all the users details:



Home
Instructions
Setup
Brute Force
Command Execution
CSRF
Insecure CAPTCHA
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About

Vulnerability: SQL Injection

User ID:

```
ID: a' or 1=1#  
First name: admin  
Surname: admin  
  
ID: a' or 1=1#  
First name: Gordon  
Surname: Brown  
  
ID: a' or 1=1#  
First name: Hack  
Surname: Me  
  
ID: a' or 1=1#  
First name: Pablo  
Surname: Picasso  
  
ID: a' or 1=1#  
First name: Bob  
Surname: Smith  
  
ID: a' or 1=1#  
First name: user  
Surname: user
```

- **SQL injection Logs Examples**

- Logs from the Web server when UserID =1 is passed in the field:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

```
10.1.0.10 - - [02/Jul/2019:05:04:00 -0400] "GET
/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit HTTP/1.1" 200 1369
"http://10.2.0.101/dvwa/vulnerabilities/sqli/" "Mozilla/5.0 (X11; Linux x86_64;
rv:60.0) Gecko/20100101 Firefox/60.0"
```

- Logs from the server when a' or 1=1# is passed:

```
10.1.0.10 - - [02/Jul/2019:05:05:32 -0400] "GET
/dvwa/vulnerabilities/sqli/?id=a%27+or+1%3D1&Submit=Submit HTTP/1.1" 200
139 "http://10.2.0.101/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" "Mozilla/5.0
(X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```



```
/dvwa/vulnerabilities/sqli/?id=a'+or+1=1&Submit=Submit HTTP/1.1
```

- **Analyzing logs from sqlmap tool:**

```
10.1.0.10 - - [08/Sep/2019:10:45:15 -0400] "GET /dvwa/vulnerabilities/sqli/?id=%2
7%20AND%207890%3D8539%20AND%20%27aGaG%27%3D%27aGaG HTTP/1.1" 302 20 "-"
"sqlmap/1.3.6#stable (http://sqlmap.org)"
```

```
10.1.0.10 - - [08/Sep/2019:10:45:15 -0400] "GET /dvwa/vulnerabilities/sqli/?id=%2
0AND%208046%3D4769--%20yAOU HTTP/1.1" 302 20 "-" "sqlmap/1.3.6#stable (http://sqlmap.org)"
```

```
10.1.0.10 - - [08/Sep/2019:10:45:15 -0400] "GET /dvwa/vulnerabilities/sqli/?id=%2
8SELECT%20%28CASE%20WHEN%20%286377%3D3397%29%20THEN%20%27%27%20E
LSE%20%28SELECT%203397%20UNION%20SELECT%207834%29%20END%29%29
HTTP/1.1" 302 20 "-" "sqlmap/1.3.6#stable (http://sqlmap.org)"
```

It is important to note the common SQL words being used in the request. In the above logs, status code 302 is being returned which means redirection and the commands aren't successfully executed.

Below are the 3 lines of code decoded from the above example:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

10.1.0.10 - - [08/Sep/2019:10:45:15-0400] "GET/dvwa/vulnerabilities/sqli/?id=' AND 7890=8539 AND 'aGaG'='aGaG HTTP/1.1" 302 20 "-" "sqlmap/1.3.6#stable(http://sqlmap.org)"

10.1.0.10 - - [08/Sep/2019:10:45:15-0400] "GET/dvwa/vulnerabilities/sqli/?id= AND 8046=4769- yAOU HTTP/1.1" 302 20 "-" "sqlmap/1.3.6#stable(http://sqlmap.org)"

10.1.0.10 - - [08/Sep/2019:10:45:15-0400] "GET/dvwa/vulnerabilities/sqli/?id=(SELECT (CASE WHEN (6377=3397) THEN " ELSE (SELECT 3397 UNION SELECT 7834) END)) HTTP/1.1" 302 20 "-" "sqlmap/1.3.6#stable(http://sqlmap.org)"

We can consult our database admins to understand if a command is a malicious.

- **SQL injection Logs Example**

10.1.0.10 - - [08/Sep/2019:10:45:16 -0400] "GET /dvwa/vulnerabilities/sqli/?id=%20AND%202934%3D%28SELECT%20UPPER%28XMLType%28CHR%2860%29%7C%7CCHR%2858%29%7C%7CCHR%28113%29%7C%7CCHR%28122%29%7C%7CCHR%28120%29%7C%7CCHR%28107%29%7C%7CCHR%28113%29%7C%7C%28SELECT%20%28CASE%20WHEN%20%282934%3D2934%29%20THEN%201%20ELSE%200%20END%29%20FROM%20DUAL%29%7C%7CCHR%28113%29%7C%7CCHR%2898%29%7C%7CCHR%28118%29%7C%7CCHR%28107%29%7C%7CCHR%28113%29%7C%7CCHR%2862%29%29%29%20FROM%20DUAL%29 HTTP/1.1" 302 20 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

10.1.0.10 - - [06/Jun/2019:06:01:23 -0400] "GET /?Pjq=3481%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20.%2F..%2F.%2Fetc%2Fpasswd%27%29%23 HTTP/1.1" 200 4785 "-" " Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

In the first log's encoded request, we can notice the various SQL words underlined.

In the second log, it is important to note the UNIX commands underlined.

The decoded request for the above logs would look like the below:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

```
?id= AND 2934=(SELECT  
UPPER(XMLType(CHR(60)||CHR(58)||CHR(113)||CHR(122)||CHR(120)||CHR(107)||CHR(113)||  
(SELECT (CASE WHEN (2934=2934) THEN 1 ELSE 0 END) FROM  
DUAL)||CHR(113)||CHR(98)||CHR(118)||CHR(107)||CHR(113)||CHR(62))) FROM  
DUAL)HTTP/1.1
```

```
10.1.0.10--[06/Jun/2019:06:01:23-0400]"GET/?Pjzq=3481 AND 1=1 UNION ALL SELECT  
1,NULL, table_name FROM information_schema.tables WHERE 2>1--/**/; EXEC  
xp_cmdshell('cat ../../etc/passwd')# HTTP/1.1"2004785"- " Mozilla/5.0 (X11; Linux x86_64;  
rv:60.0) Gecko/20100101 Firefox/60.0 "
```

- Identifying the SQL injection of the request a' or 1=1# from the POST request:
10.1.0.10 --[03/Jul/2019:19:38:58 -0400] "POST /mutillidae/index.php?page=login.php
HTTP/1.1" 302 8955 "http://10.2.0.101/mutillidae/index.php?page=login.php" "Mozilla/5.0
(X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

The packet capture for the above request is below:

```
Hypertext Transfer Protocol  
POST /mutillidae/index.php?page=login.php HTTP/1.1\r\nHost: 10.2.0.101\r\nUser-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nReferer: http://10.2.0.101/mutillidae/index.php?page=login.php\r\nContent-Type: application/x-www-form-urlencoded\r\nContent-Length: 69\r\nCookie: showhints=1; acop=1\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\n\r\nHTML Form URL Encoded: application/x-www-form-urlencoded  
  Form item: "username" = "admin' or 1=1#"  
  Form item: "password" = ""  
  Form item: "login-php-submit-button" = "Login"  
  
File Data: 69 bytes  
  HTML Form URL Encoded: application/x-www-form-urlencoded  
    Form item: "username" = "admin' or 1=1#"  
    Form item: "password" = ""  
    Form item: "login-php-submit-button" = "Login"
```

It is important to note that the username field in the HTML form is similar to the SQL injections that we noticed via GET request.

- Identifying the SQL injection attacks from the logs:
 - Check for various SQL commands in the logs. Below are some commonly used SQL commands:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- FROM
- SELECT
- WHERE
- AND
- OR
- EXEC
- TABLE
- DROP
- UNION

- Look for the encoded request with the % symbol.
- Look for the User Agents like sqlmap.
- OS commands being used in requests might be suspicious.

- **Post-Assessment Question**

Analyze the Web application log below and identify the type of attack:

```
10.1.0.10 - - [08/Sep/2019:10:45:15 -0400] "GET /login/?id=%20AND%20%28SELECT%208038%20FROM%28SELECT%20COUNT%28%2A%29%2C29%29x%20FROM%20INFORMATION_SCHEMA.PLUGINS%20GROUP%20BY%20x%29a%29 HTTP/1.1" 302 20 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

- i. DDOS
- ii. SQL injection
- iii. SYN Flood
- iv. Remote File Inclusion
- v. Local File Inclusion

It is easy to notice the SQL words like AND, SELECT, COUNT, FROM in the request. The decoded request would look like below:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

10.1.0.10--[08/Sep/2019:10:45:15-0400]" GET /login/?id= AND (SELECT 8038 FROM(SELECT COUNT(*) FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) HTTP/1.1" 302 20 "-" "Mozilla/5.0(X11; Linuxx86_64;rv:60.0) Gecko/20100101 Firefox/60.0"

- **Post-Assessment Question**

"Web Server logs will always show all the information about the user actions". Is this statement True or False?

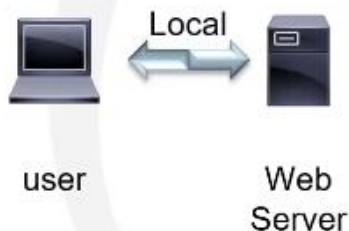
Answer: The information is false as in case of a post or if the Web server configuration for logs are wrong we may not get all the needed information.

Lesson 2.5:File Inclusion

Skills Learned From This Lesson: **Log analysis, Local File inclusion, Remote File inclusion**

- It is a type of Injection attack. It is caused by wrong user input validation.
- Allows user to submit input into files or upload files to the server.
- Usually, has the directory traversal characters with the ../ symbol
- 2 types of File Inclusion:
 - Local File Inclusion (LFI): Access and executes Local Files/Commands.
 - Remote File Inclusion (RFI): Access and executes Remote Files/Commands
- It is a Server-Side Attack.

- **Local File Inclusion**



Local File Inclusion accesses local files and sends requests on the same Web server.

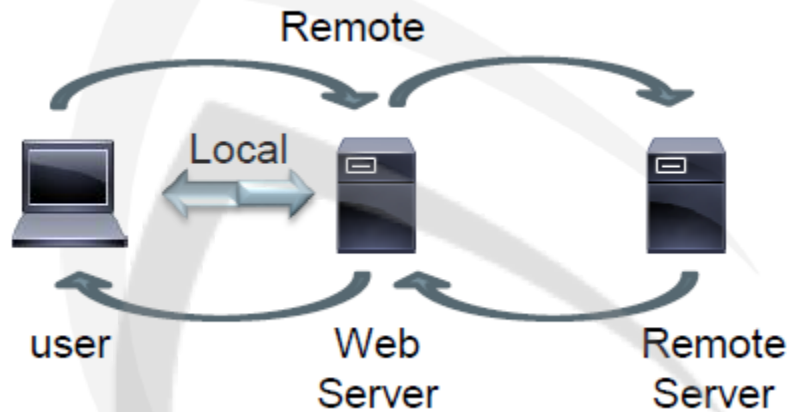
Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Remote File Inclusion



Remote File Inclusion accesses remote files on remote servers using the local server. We need 2 different servers to perform this attack.

- Local File Inclusion

The screenshot shows the DVWA interface. The top navigation bar includes 'Home', 'Instructions', 'Setup', 'Brute Force', 'Command Execution', 'CSRF', 'Insecure CAPTCHA', and 'File Inclusion' (which is highlighted in green). The main content area is titled 'Vulnerability: File Inclusion' and contains a text box with the instruction: 'To include a file edit the ?page=index.php in the URL to determine which file is included.' Below this, there is a 'More info' section with two links: http://en.wikipedia.org/wiki/Remote_File_Inclusion and http://www.owasp.org/index.php/Top_10_2007-A3.

When the request is sent:

<http://10.2.0.101/dvwa/vulnerabilities/fi?page=../../../../etc/passwd>

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

As a result of the above request, the password file will be read and displayed in the Web page as below:



- Below is a normal request.

```
10.1.0.10 - - [09/Jul/2019:08:49:44 -0400] "GET /dvwa/vulnerabilities/fi/?page=include.php HTTP/1.1" 200 1250 "http://10.2.0.101/dvwa/index.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

- Below is request to access the passwd file. The Web server's response is "error code 404" which means the Web server didn't answer.

```
10.1.0.10 - - [09/Jul/2019:08:51:27 -0400] "GET /etc/passwd HTTP/1.1" 404 181 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

- In the below request, the Web server responds with a "code 200" which means that the passwd file is returned successfully

```
10.1.0.10 - - [09/Jul/2019:08:55:30 -0400] "GET /dvwa/vulnerabilities/fi/?page=../../../../../../etc/passwd HTTP/1.1" 200 1656 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Below is encoded form of the above request and returns “code 200” with response as success

```
10.1.0.10 - - [09/Jul/2019:08:58:02 -0400] "GET -  
/dvwa/vulnerabilities/fi/?page=..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"  
200 1656 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

- Files targeted in the inclusion attacks for various OS:

OS File Locations		
Linux	Windows	OS X/Mac OS
<ul style="list-style-type: none">- /etc/issue- /proc/version- /etc/profile- /etc/passwd- /etc/passwd- /etc/shadow- /root/.bash_history- /var/log/dmmessage- /var/mail/root- /var/spool/cron/crontabs/root	<ul style="list-style-type: none">- %SYSTEMROOT%repairsystem- %SYSTEMROOT%repairSAM- %SYSTEMROOT%repairSAM- %WINDIR%win.ini- %SYSTEMDRIVE%boot.ini- %WINDIR%Panthersysprep.inf- %WINDIR%system32configAppEvent.Evt	<ul style="list-style-type: none">- /etc/fstab- /etc/master.passwd- /etc/resolv.conf- /etc/sudoers- /etc/sysctl.conf

- Identifying LFI from the logs:
 - Weird requests: Look out for the requests that you wouldn't expect on a Web page.
 - Access to OS Commands/Critical Files that you wouldn't expect.
 - New Files noticed in server
 - URL Encoded requests
 - Increase in upload of files from or to server.
 - Answers with huge size might be a content of critical file being reflected
 - Many symbols of “/” or “%2F” in the request indicating directory traversal

- **Remote File Inclusion**

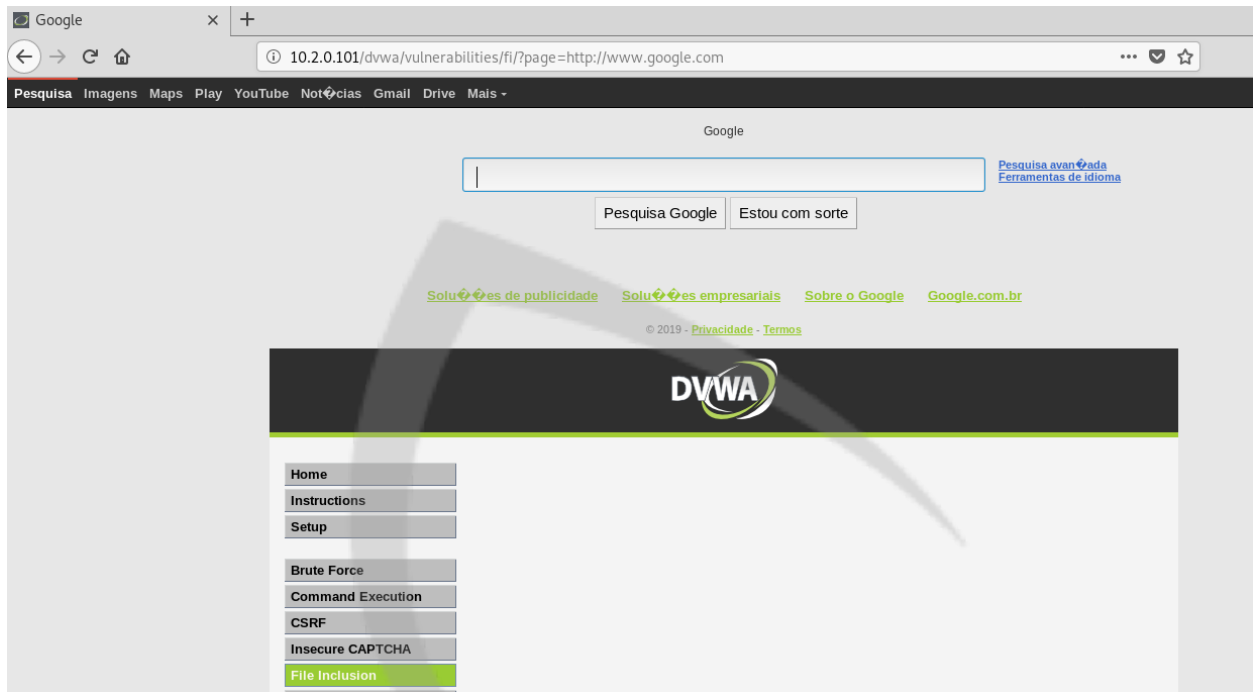
Files from a remote server are accessed from our local server. Google site is accessed from our lab:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY



As a result, Google search bar is loaded in our lab Web page. The remote server here is the Google server. Although we are accessing the remote server, we will get logs from our local server too as it is the target.

- **Identifying RFI attacks from logs**

"GET /dvwa/vulnerabilities/fi/?page=include.php HTTP/1.1" 200 1250 "http://10.2.0.101/dvwa/"

The above log is a genuine request. The last few fields of the original request is omitted.

"GET /dvwa/vulnerabilities/fi/?page=http://www.google.com HTTP/1.1" 200 16360 "-"

The above log is an example of RFI attack where user or attacker tried to access the Google servers via the Web application.

All the below requests are example of RFI. user or attacker tried to access one or more resources from the remote Google Web server.

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

"GET /images/branding/googlelogo/1x/googlelogo_white_background_color_272x92dp.png HTTP/1.1" 404 229
"http://10.2.0.101/dvwa/vulnerabilities/fi/?page=http://www.google.com"

"GET /textinputassistant/tia.png HTTP/1.1" 404 192 "http://10.2.0.101/dvwa/vulnerabilities/fi/?page=http://www.google.com"

"GET /client_204?&atyp=i&biw=1440&bih=683&ei=0cQdXYnyG9SV0Abf6Y-oCg HTTP/1.1" 404 182
"http://10.2.0.101/dvwa/vulnerabilities/fi/?page=http://www.google.com"

"GET /images/nav_logo229.png HTTP/1.1" 404 193 "http://10.2.0.101/dvwa/vulnerabilities/fi/?page=http://www.google.com"

"GET /xjs/_/js/k=xjs.hp.en.sE3cBMPylaA.O/m=sb_he,d/am=ABbUBA/d=1/rs=ACT90oHXMt3OvyOjimr_tHB1hyLdxQq99A HTTP/1.1" 404 254 "http://10.2.0.101/dvwa/vulnerabilities/fi/?page=http://www.google.com"

"GET /images/nav_logo229.png HTTP/1.1" 404 193 "http://10.2.0.101/dvwa/vulnerabilities/fi/?page=http://www.google.com"

- **Remote File Inclusion**

- Requests for other/unknown servers – outside traffic
- Increase in Web server download traffic
- Encoded requests
- User Agent

- **Post-Assessment Question**

Analyze the Web log below and choose the type of attack.

10.1.0.10 - - [04/Jul/2019:05:34:56 -0400] "GET /forum/?page=http://www.cybrary.it HTTP/1.1" 200 26768 "http://10.2.0.101/forum/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

- i. DDOS
- ii. SQL injection
- iii. SYN Flood
- iv. Remote File Inclusion
- v. Local File Inclusion

Answer: It is iv. Remote File Inclusion attack as we are trying to access the Cybrary Web page from our application.

- **Post-Assessment Question**

Analyze the Web log below and identify which Web page is vulnerable to the file inclusion attack.

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

10.1.0.10 - - [04/Jul/2019:05:34:56 -0400] "GET /forum/?page=http://www.cybrary.it HTTP/1.1" 200 26768 "http://10.2.0.101/forum/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

Answer: If we check the referrer we can identify the original Web page <http://10.2.0.101/forum/?page=>

Lesson 2.6: XSS – Cross-Site Scripting

Skills Learned From This Lesson: **Log analysis, Reflected XSS, Stored XSS**

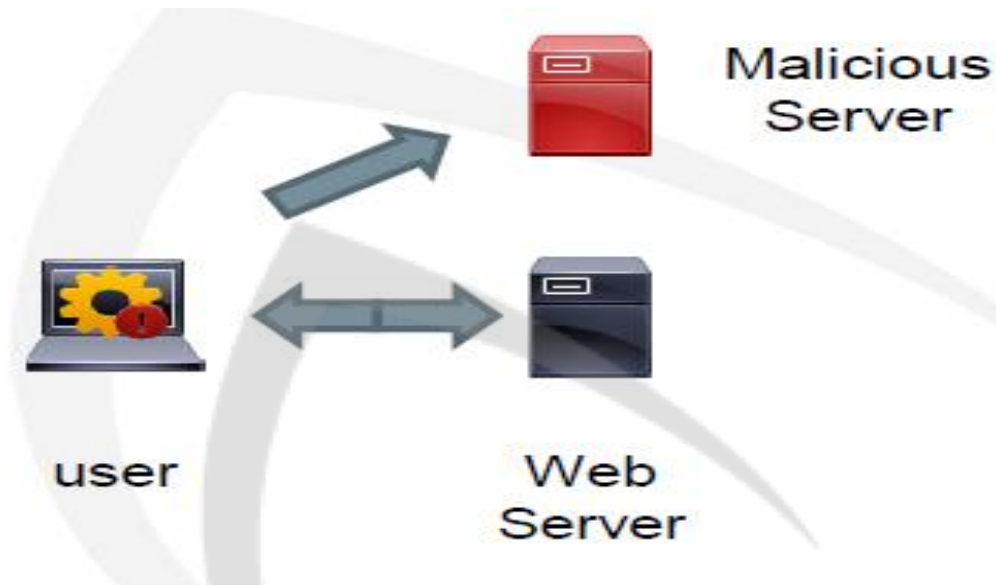
- It is a type of Injection Attack
- Client-Side Attack
- Pages with Forms and Forums are most common examples
- Uses JavaScript processed by user browser
- 2 Types – Stored and Reflected
- Caused by lack of user input validation
- A7 Topic in 2017 – OWASP Top 10
- **Process of XSS**
 - User accesses the Website
 - The Web server answers the user's request
 - The Web server answer is processed by the user's browser
 - If the answer contains a malicious code, it will be executed by the browser
 - The types (Client or Server) of XSS depends on the origin of the malicious code
 - Some actions common to XSS are: Redirection to other sites, Cryptomining, Credential theft, infecting user computer with malwares or backdoors.

Brought to you by:

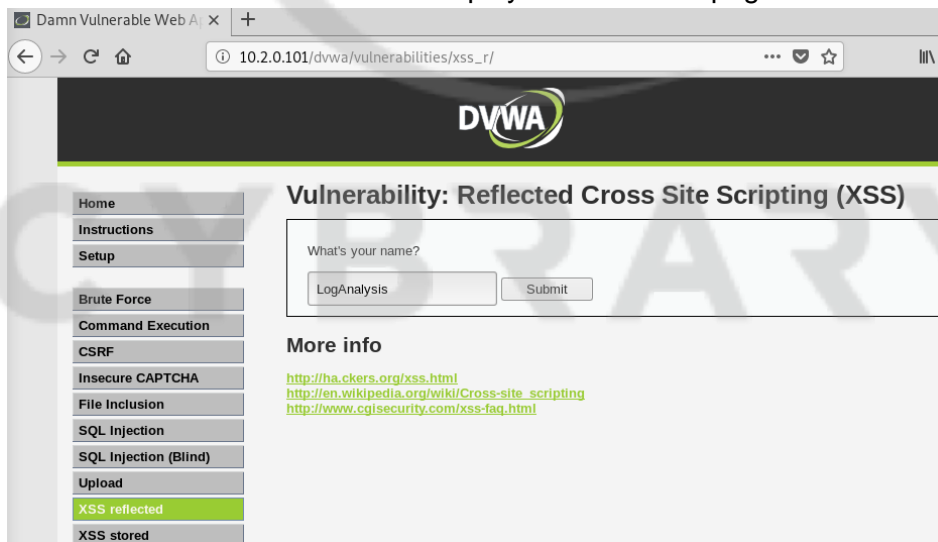
CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY



- **Reflected Cross-Site Scripting (XSS)**
Demonstrating Reflected XSS in lab
Whatever we enter in the box will be displayed in the Web page.



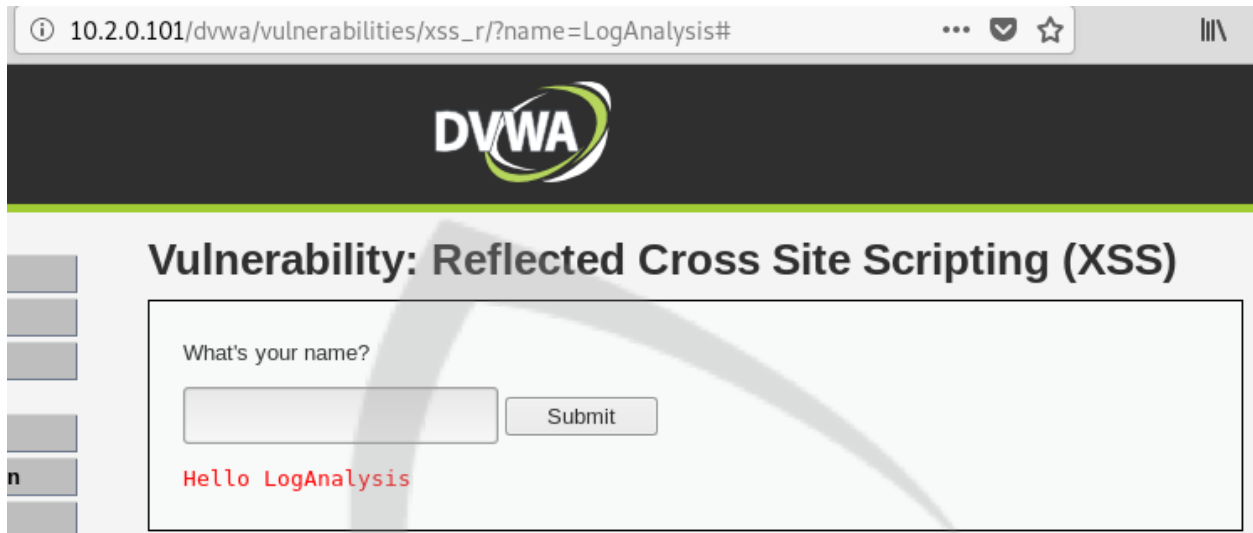
Following message is displayed in the Web page as a result, once we click submit in the box

Brought to you by:

CYBRARY | FOR BUSINESS

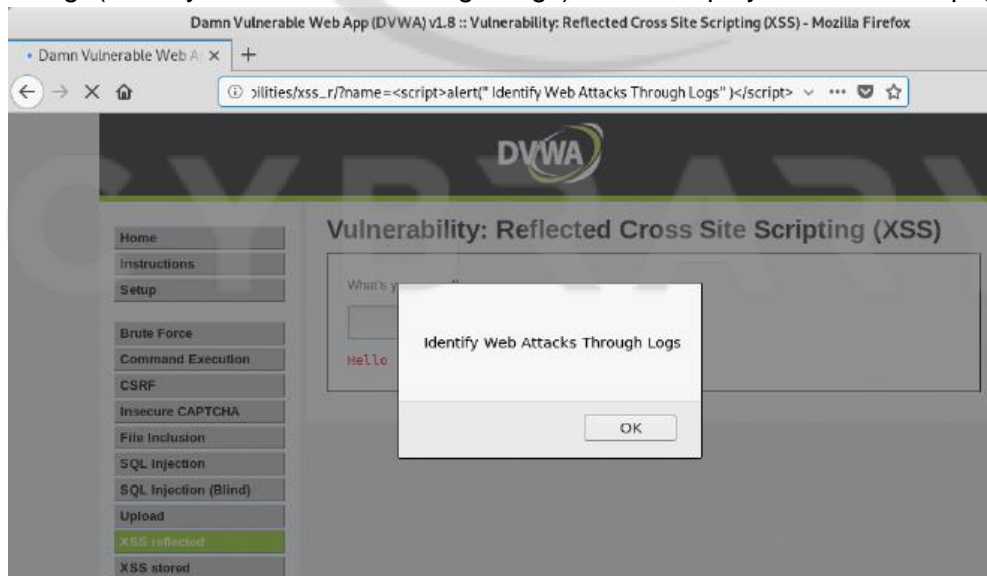
Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY



http://10.2.0.101/dvwa/vulnerabilities/xss_r/?name=LogAnalysis#

- Result of passing a javascript in the above textbox: `?name=<script>alert("Identify Web Attacks Through Logs")</script>` Message(Identify Web Attacks Through Logs) will be displayed in the Web page:



When Web browser processes the answer from the Web server, it executes the java script and a message is displayed.


Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Web logs from 1st attack where we have typed the text, "Log Analysis":
10.1.0.10 - - [10/Jul/2019:17:30:54 -0400] "GET /dwa/vulnerabilities/xss_r/?name=?name=LogAnalysis HTTP/1.1" 200 1340 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
- Web logs from 2nd attack where we have executed a javascript in the browser
10.1.0.10 - - [10/Jul/2019:17:35:13 -0400] "GET /dwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert(%22%20Identify%20Web%20Attacks%20Through%20Logs%22%20)%3C/script%3E HTTP/1.1" 200 1349 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

<script>alert("Identify Web Attacks Through Logs")</script>  %3Cscript%3Ealert(%22%20Identify%20Web%20Attacks%20Through%20Logs%22%20)%3C/script%3E

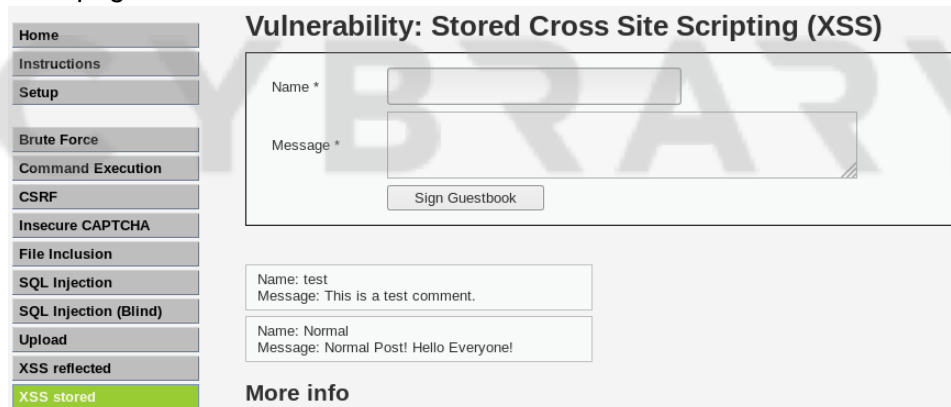
- **Identifying Reflected Cross-Site Scripting (XSS)**

- <script> and </script> in the request
- JavaScript code in the request
- Many encoded characters
- Unexpected user agents

- **Stored Cross-Site Scripting (XSS)**

It changes a Web page.

Name and message box are available, where we can type some data which will reflect in the Web page under normal behavior:



To perform the stored XSS, we will pass the following javascript:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

Vulnerability: Stored Cross Site Scripting (XSS)

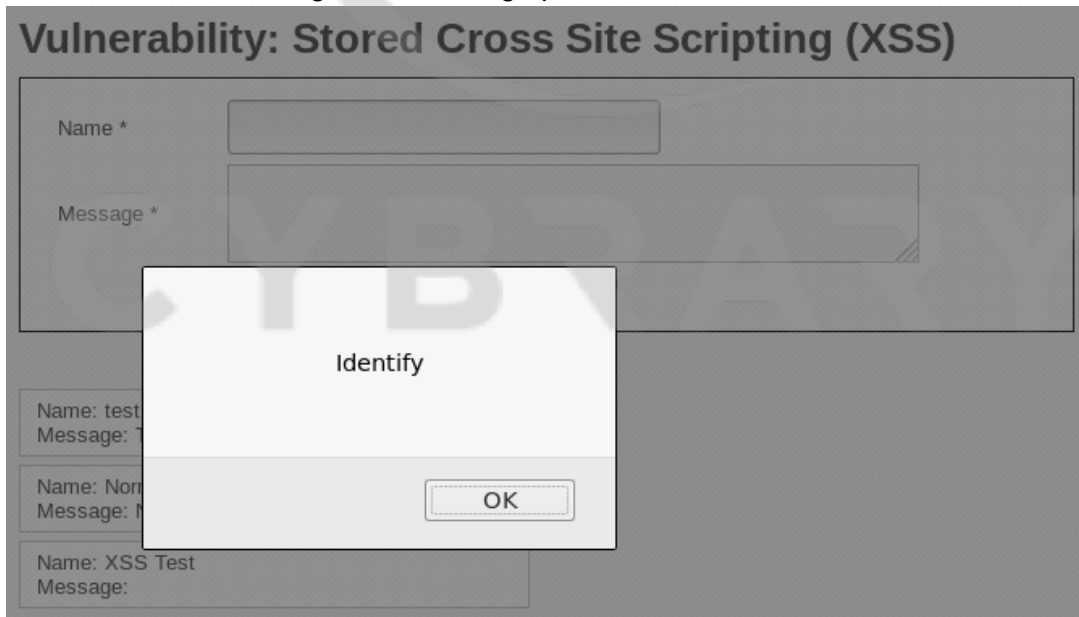
Name *

Message *

Name: test
Message: This is a test comment.

Name: Normal
Message: Normal Post! Hello Everyone!

As a result, whenever a Web page is opened, following alert will pop up and message board will show nothing in the message part



This behavior is observed because our message part contains a script and a script is executed in the Web browser but not reflected in the message.

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

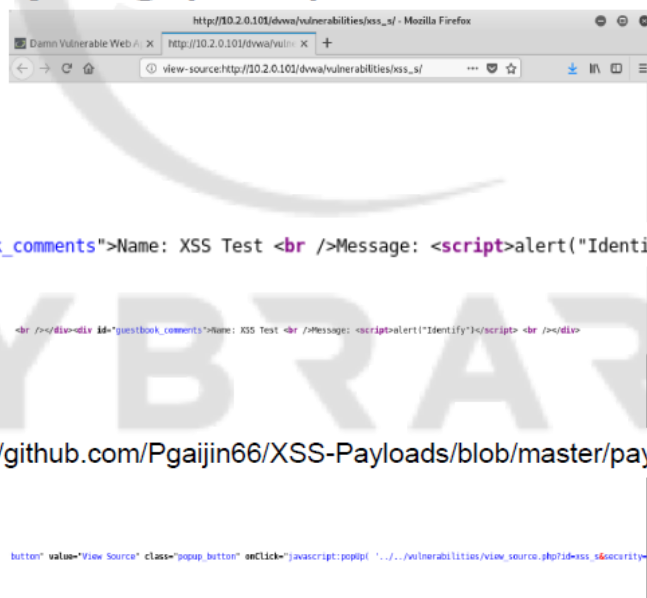
- Identifying the Stored XSS:

PHPIDS Log

```
Date/Time: 2019-07-10T18:45:19-04:00
Vulnerability: xss csrf id rfe lfi sqli
Request: /dvwa/vulnerabilities/xss_s/
Variable: REQUEST.mtxMessage=<script>alert("Identify")</script> POST.mtxMessage=
<script>alert("Identify")</script>
IP: 10.2.0.101
```

The IPS logs can help us in identifying the Stored XSS attack. If our IPS is in block mode, the attack will fail. The IPS analyzes the full packet and with that kind of analysis, we can see the malicious request. Another example of stored XSS is Cryptomining. Since the attacker can store the malicious code in the Web page and whenever a user loads the Web page, the malicious code will be processed. If the user's Web browser is asked for Cryptomining, the user's machine will run slow.

Another way to check the attack is to check the Web page code:



Multiple payloads are used in XSS. We could use similar ways to detect a stored XSS and reflected XSS. However, since reflected/stored(#choose) XSS uses the POST

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

request, it is best to get logs from other log sources too, like IPS. If possible we shall check the Web page code and look for malicious commands.

- **Post-Assessment Question**

“There is no difference between Stored XSS and Reflected XSS Attacks”. Is this statement True or False?

Answer: False as they differ in the way they are executed.

- **Post-Assessment Question**

Analyze the Web log below and identify which part of the log is malicious.

Analyze the web log below and identify which part of the log is malicious.

```
10.1.0.10 - - [10/Jul/2019:17:35:13 -0400] "GET
/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert(%22%20Identify%20Web%20Attacks%20T
hrough%20Logs%22%20)%3C/script%3E HTTP/1.1" 200 1349 "-" "Mozilla/5.0 (X11; Linux x86_64;
rv:60.0) Gecko/20100101 Firefox/60.0"
```

```
%3Cscript%3Ealert(%22%20Identify%20Web%20Attacks%20Through%20Logs%22%20)%3C/script%3E
```

Answer: The decoded part of malicious request is below:

```
%3Cscript%3Ealert(%22%20Identify%20Web%20Attacks%20Through%20Logs%22%20)%3C/script%3E
```

```
<script>alert(" Identify Web Attacks Through Logs" )</script>
```

Lesson 2.7: CSRF – Cross Site Request Forgery

Skills Learned From This Lesson: **Log analysis, CSRF, forging requests**

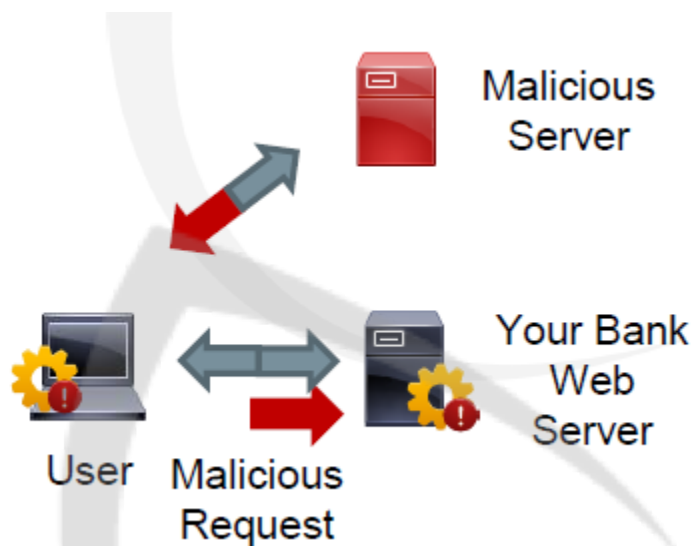
- Exploit the trust between Web server and user's browser
- User accesses malicious Web server
- Weird commands are executed by browser

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY



User accesses a bank Web server using its browser. The Banks Web server trusts the user browser. User visits a malicious server after that which compromises the user Web browser. It may infect the user browser to perform unsanctioned transactions like sending money to the attackers account. The malicious server is the forger here, which sends the request as coming from the trusted Web browser.

- **Identifying through logs**

Below are the logs from legitimate access:

```
10.1.0.10 -- [Jul/17/2019:05:37:01 -0400] "POST /dvwa/login.php HTTP/1.1" 302 20
"http://10.2.0.101/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0"
10.1.0.10 -- [Jul/17//2019:05:37:01 -0400] "GET /dvwa/index.php HTTP/1.1" 200 1660
"http://10.2.0.101/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0"
10.1.0.10 -- [Jul/17//2019:05:37:05 -0400] "GET /dvwa/vulnerabilities/csrf/ HTTP/1.1" 200 1376
"http://10.2.0.101/dvwa/index.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0"
```

- Logs from a password change requests would be like below:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

```
10.1.0.10 - - [Jul/17//2019:05:37:10 -0400] "GET
/dvwa/vulnerabilities/csrf/?password_new=asdf&password_conf=asdf&Change=Change HTTP/1.1" 200
1395 "http://10.2.0.101/dvwa/vulnerabilities/csrf/" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0)
Gecko/20100101 Firefox/60.0"
```

We can see the password change request from the above logs.

- Below logs are generated from the CSRF:

```
10.1.0.10 - - [Jul/17//2019:05:38:50 -0400] "GET
/dvwa/vulnerabilities/csrf/?password_new=test2&password_conf=test2&Change=Change HTTP/1.1" 200
1388 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

It is important to note the difference between the 2 logs generated from the password change request. Note that the 2 logs have different passwords being entered and the second log misses the referrer.

- **Identifying attack through logs**

- Referrer is the best way to identify: Checking Referrer is a good way to identify CSRF attack.
- Different behavior from the user: Changing password or trying to change the password in a very small period.
- Same action with a uncommon interval.

- **Post-Assessment Question**

“CSRF attack only happens if the user browser is compromised” Is this affirmation True or False?

Answer: False. Most of the times attack will happen because the user visited a malicious Website.

- **Post-Assessment Question**

Analyze the Web log below and identify the possible attack:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

```
10.1.0.30 - - [11/Jul/2019:05:37:05 -0400] "GET /example.com/ HTTP/1.1" 200 1376 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
10.1.0.30 - - [11/Jul/2019:05:37:10 -0400] "GET /example.com/changelogin.html HTTP/1.1" 200 1395 "http://example.com/ " "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
10.1.0.30 - - [11/Jul/2019:05:38:50 -0400] "POST example.com/changelogin.html HTTP/1.1" 200 1388 "http://example.com/changelogin.html HTTP/1.1" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
10.1.0.30 - - [11/Jul/2019:05:41:05 -0400] "POST example.com/changelogin.html HTTP/1.1" 200 1388 "http://littlecutdogs.com/pictures.html" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

Answer:

Possible CSRF Attack using POST requests

Source attack: <http://littlecutdogs.com/pictures.html>

Username and Password: Unknown

Lesson 2.8: Other Log Sources - Part 1

Skills Learned From This Lesson: **Log analysis, SYN Flood attacks, HTTP flood attack**

- **Other Log Sources**

- **Packets:** Analyzing packets could reveal some critical information and attack signatures.
- **Firewall:** Firewall logs could help in identifying the connections made to the Web server and if they are malicious.
- **Network bandwidth usage:** Spiked network bandwidth usage could point towards an attack.

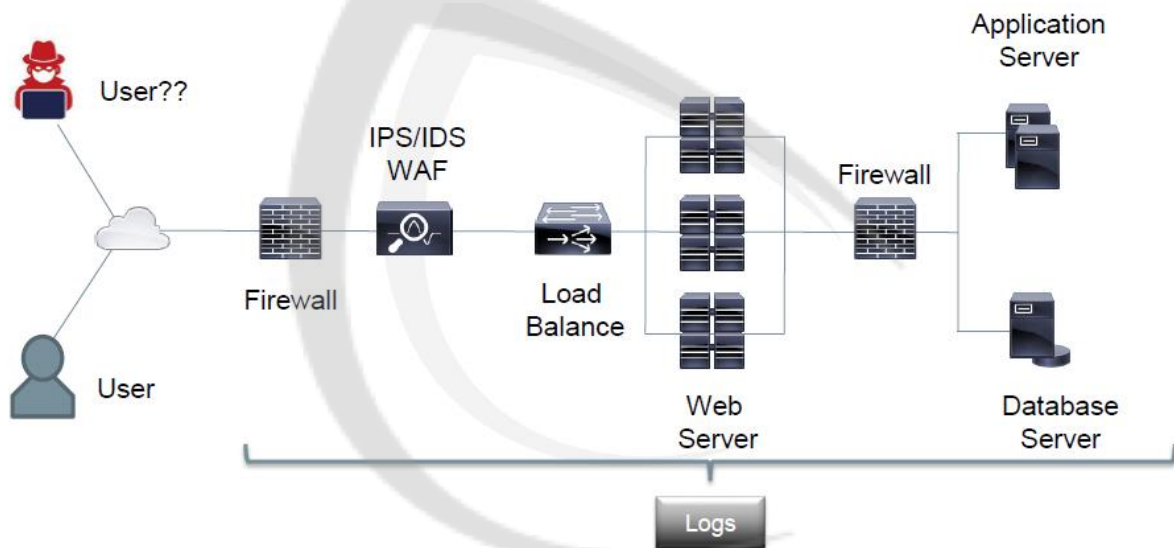
Brought to you by:

CYBRARY | FOR BUSINESS

*Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.*

CYBRARY

- **CPU and memory usage:** Spiked CPU and memory usage could point towards an attack.
 - **IPS/IDS:** IPS/IDS is used to analyze the packets at Network layer(L3). IPS/IDS helps us to detect attacks.
 - **WAF:** Web application Firewall (WAF) could inspect packets at the application layer(L7) and makes detecting the attacks easier.
- **Various sources that can help with logs**



- **Flooding Attacks:** Tries to consume all the resources available for hosting a Web server
- ### Network Attacks - SYN Flood
- Usually a one way connection to exhaust resources: A 3 way handshake won't be completed in this attack. Only the first SYN request will be sent to the server, trying to consume all the computing capacity on the networking device to render the genuine requests useless as they won't be processed due to the Web server being unavailable.
 - It's a way to cause a DOS/DDOS attack: Web server won't be available due to the resource crunch.
 - All equipment that handles the TCP Layer can be affected: The components associated with handling the networking will be impacted.

Application Attacks - HTTP Flood

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

- Usually a 3 way connection to exhaust resources without traffic
- It's a way to cause a DOS/DDOS attack
- The Web server will be the most affected as computing capacity will be in dearth for the genuine request.

- **Analyzing SYN flood attack from the logs**

Analyzing logs from the firewall. Analyze the below logs for SYN flood attack

- Jul 24 09:53:34 10.1.0.1 filterlog:
64,64832,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33496,80,120,S,1978812743:1978812863,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,55942,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33497,80,120,S,629909069:629909189,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,9656,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33498,80,120,S,51835925:51836045,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,52193,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33499,80,120,S,598699458:598699578,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,6071,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33500,80,120,S,1118867074:1118867194,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,17366,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33501,80,120,S,1293670631:1293670751,,64,,
- ...
- Jul 24 09:54:22 10.1.0.1 filterlog:
64,17366,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33501,80,120,S,1293671634:1293670821,,64,,

We could deduce the following from the above logs:

- Jul 24 09:53:34 10.1.0.1 filterlog:
64,64832,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33496,80,120,S,1978812743:1978812863,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,55942,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33497,80,120,S,629909069:629909189,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,9656,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33498,80,120,S,51835925:51836045,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,52193,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33499,80,120,S,598699458:598699578,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,6071,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33500,80,120,S,1118867074:1118867194,,64,,
- Jul 24 09:53:34 10.1.0.1 filterlog:
64,17366,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33501,80,120,S,1293670631:1293670751,,64,,
- ...
- Jul 24 09:54:22 10.1.0.1 filterlog:
64,17366,0,none,6,tcp,160,10.1.0.10,10.2.0.101,33501,80,120,S,1293671634:1293670821,,64,,

No Web Server Logs!!

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

Underlined in the above image, we can see the timestamp of the log. Firewall IP is 10.1.0.1 . A TCP connection is trying to be established. 10.1.0.10 is the attacker IP while 10.2.0.101 is the Web server IP. Port 80 means HTTP connection and S stands for the SYN flag. Depending on the firewall and the Web server capacity, a lot of connections can be handled. Thus depending on the environment, an attacker will need to generate a lot of connections. It is important to note that there won't be any Web server logs as a TCP 3-way handshake isn't performed yet and generally, the Web server logs are only logged after a TCP 3-way handshake.

Running the netstat command on the Web server we can deduce the following:

- An IP 10.1.0.10 is trying to connect to our Web server over port 80 using different ports.
- Only SYN_RECV connections are logged in the Web server and connection is waiting to be completed.

```
tcp      0      0 10.2.0.101:80      10.1.0.10:997      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:977      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:963      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:981      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:755      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:992      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:1004     SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:767      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:980      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:979      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:1003     SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:780      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:770      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:995      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:761      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:1001     SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:788      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:970      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:792      SYN_RECV
tcp      0      0 10.2.0.101:80      10.1.0.10:773      SYN_RECV
```

- **HTTP FLOOD Attack**

Firewall logs from the HTTP flood attack:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

```
Jul 24 22:02:39 10.1.0.1 filterlog:
64,14912,0,DF,6,tcp,60,10.1.0.10,10.2.0.101,57858,80,0,S,1295268142,,29200,,mss,sackOK;TS;nop;wsc
ale
Jul 24 22:02:39 10.1.0.1 filterlog:
64,56391,0,DF,6,tcp,60,10.1.0.10,10.2.0.101,57860,80,0,S,4204715817,,29200,,mss;sackOK;TS;nop;wsc
ale
Jul 24 22:02:39 10.1.0.1 filterlog:
64,26818,0,DF,6,tcp,60,10.1.0.10,10.2.0.101,57862,80,0,S,3139464109,,29200,,mss;sackOK;TS;nop;wsc
ale
Jul 24 22:02:39 10.1.0.1 filterlog:
64,42937,0,DF,6,tcp,60,10.1.0.10,10.2.0.101,57864,80,0,S,1703912662,,29200,,mss;sackOK;TS;nop;wsc
ale
Jul 24 22:02:39 10.1.0.1 filterlog:
64,31430,0,DF,6,tcp,60,10.1.0.10,10.2.0.101,57866,80,0,S,2567403945,,29200,,mss;sackOK;TS;nop;wsc
ale
```

Here we have, S=SYN but also the sackOK which means that the HTTP 3 way handshake was completed.

3 way HandShake SYN <-> SYN/ACK <-> ACK

Since we have established a connection via a 3 way handshake, let's check the server logs:

- 10.1.0.10 - - [24/Jul/2019:18:02:31 -0400] "GET /?1280 HTTP/1.1" 400 226 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36"
- 10.1.0.10 - - [24/Jul/2019:18:02:31 -0400] "GET /?1721 HTTP/1.1" 400 226 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36"
- 10.1.0.10 - - [24/Jul/2019:18:02:31 -0400] "GET /?330 HTTP/1.1" 400 226 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36"
- 10.1.0.10 - - [24/Jul/2019:18:02:31 -0400] "GET /?1066 HTTP/1.1" 400 226 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36"
- 10.1.0.10 - - [24/Jul/2019:18:02:31 -0400] "GET /?1765 HTTP/1.1" 400 226 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36"

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

The point to notice in these kinds of requests is, the large number of bad requests being sent, usually attacker doesn't go well with error being thrown in response. Depending on the firewall and the Web server capacity a lot of connections can be handled. Thus, depending on the environment, an attacker will need to generate a lot of connections. Always the response needs to be 400 bad response. It can be 5XX or a success of 2XX too.

Using netstat command from HTTP flood attack:

We can see that the connections are established here unlike to SYN attack. There are multiple connections from the same machine to Web server over different ports.

```
tcp      188      0 10.2.0.101:80      10.1.0.10:50214    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50102    ESTABLISHED
tcp      188      0 10.2.0.101:80      10.1.0.10:50216    ESTABLISHED
tcp      187      0 10.2.0.101:80      10.1.0.10:50226    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50030    ESTABLISHED
tcp      189      0 10.2.0.101:80      10.1.0.10:50202    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50036    ESTABLISHED
tcp      188      0 10.2.0.101:80      10.1.0.10:50190    ESTABLISHED
tcp      188      0 10.2.0.101:80      10.1.0.10:50298    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50108    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50094    ESTABLISHED
tcp      189      0 10.2.0.101:80      10.1.0.10:50292    ESTABLISHED
tcp      188      0 10.2.0.101:80      10.1.0.10:50346    ESTABLISHED
tcp      187      0 10.2.0.101:80      10.1.0.10:50366    ESTABLISHED
tcp      189      0 10.2.0.101:80      10.1.0.10:50254    ESTABLISHED
tcp      189      0 10.2.0.101:80      10.1.0.10:50342    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50048    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50110    ESTABLISHED
tcp      189      0 10.2.0.101:80      10.1.0.10:50370    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50066    ESTABLISHED
tcp      187      0 10.2.0.101:80      10.1.0.10:50180    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50044    ESTABLISHED
tcp      189      0 10.2.0.101:80      10.1.0.10:50350    ESTABLISHED
tcp      0        0 10.2.0.101:80      10.1.0.10:50042    ESTABLISHED
```

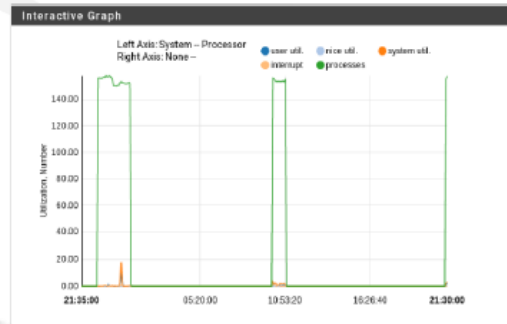
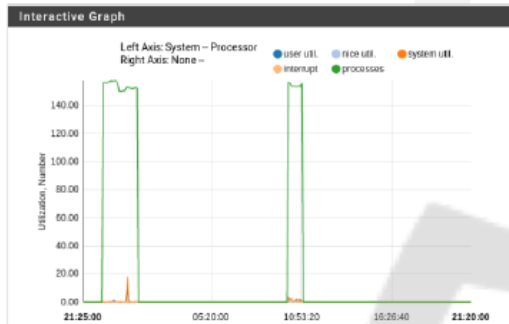
CPU Usage from the HTTP Flood attack:

Brought to you by:

CYBRARY | FOR BUSINESS

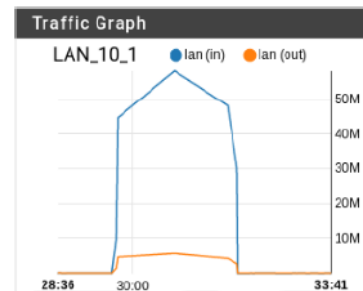
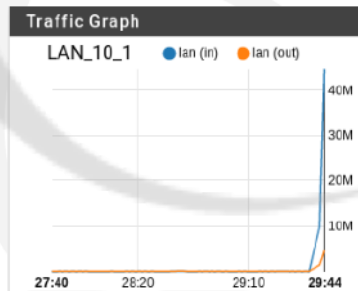
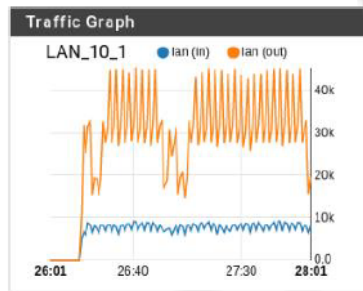
Develop your team with the *fastest growing catalog* in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY



#could we get a high resolution pic or zoom in?
We can see the high CPU usage during the HTTP flood attack.

Network Bandwidth usage from the HTTP flood attack:



We can see the network usage going from KBs to MBs.

- **Identifying Flood Attacks:**
 - Many equal requests:
 - Small period of time
 - High CPU/ Bandwidth usage
 - Half connected TCP connections
 - Uncommon or random requests
 - Web application slow/not working
 - User Agent

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

Lesson 2.9: Other Log Sources - Part 2

Skills Learned From This Lesson: **Log analysis, IDS, IPS**

- We are going to discuss about logs from below sources here:
 - IDS/IPS – Intrusion Detection/Prevention System
 - WAF – Web Application Firewall

- **IPS/IDS**

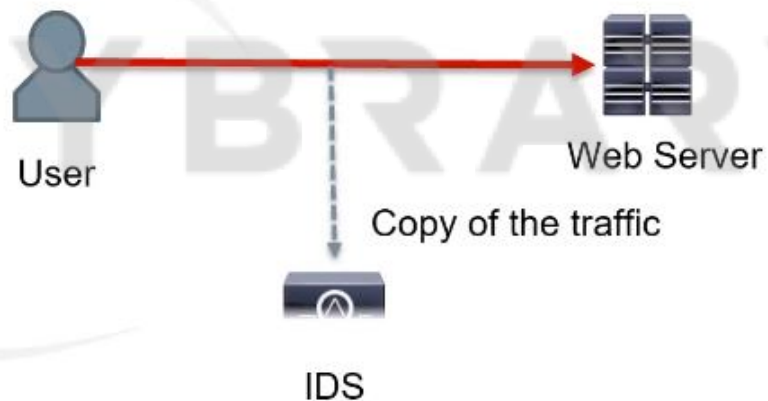
An IDS only detects an attack while IPS prevents the attack.

Tool: Snort is an Open Source IPS/IDS tool.

IPS/IDS positioning in any environment:



Prevention = Active Block



Detection = Alerts

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

Since IPS blocks and IDS doesn't block, it is obvious to think why do we have to use an IDS after all. IPS is a component and can't always be deployed in an environment as it may result in delay in communication. So it is better to have atleast an IDS in place to detect the attacks and alert the other devices for an attack.

Analyzing the IPS/IDS logs from Snort:

- 07/24-09:18:26.473642 **[**]** [1:2009714:8] ET WEB_SERVER Script tag in URI, Possible Cross Site Scripting Attempt **[**]** [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.86.10:1044 -> 69.175.78.90:80
- 11/26-19:08:42.964440 **[**]** [1:2015023:2] ET WEB_SERVER IIS 8.3 Filename With Wildcard (Possible File/Dir Bruteforce) **[**]** [Classification: Detection of a Network Scan] [Priority: 3] {TCP} 192.168.53.10:1235 -> 165.254.146.16:80
- 02/21-15:27:38.722223 **[**]** [1:10000807:1] COMMUNITY WEB-PHP CommonSense search.php SQL injection attempt **[**]** [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.21.10:1323 -> 173.239.42.213:80
- 11/09-21:53:52.487616 **[**]** [1:2016228:4] ET CURRENT_EVENTS Metasploit CVE-2013-0422 Jar **[**]** [Classification: Attempted User Privilege Gain] [Priority: 1] {TCP} 66.197.254.101:80 -> 192.168.49.10:1070
- 10/03-00:19:27.619598 **[**]** [1:2002945:10] ET POLICY Java Url Lib User Agent Web Crawl **[**]** [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.117.10:1071 -> 85.17.136.147:80

We can easily identify some attacks from the logs (underlined in blue colour) like XSS, Brute force attack, SQL injection, vulnerabilities, malicious user agent. Some well known fields like date and time, source IP-destination IP (underlined in red colour) also can be identified in the above image.

- **WAF – Web Application Firewall**
IPS and IDS analyzes all network traffic whereas WAF analyzes Web Traffic only. Like IPS WAF can be deployed before the application to take preventive actions or it can be deployed like IDS just to detect attacks.
Tool: Mod Security is Open Source example of WAF.
- Analyzing the WAF logs:
We have below WAF log to identify an attack:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

From the log, we can identify various useful information like, ModSecurity: Warning., where a file injection is being attempted. Other information like the timestamp of the client IP can be noted too.

```
[Sat Jul 14 13:40:10 2019] [error] [client 10.1.0.10] ModSecurity: Warning. Pattern match "(?:<!\\w)(?:\\.(?:ht(?:access|passwd|group)|www_?acl)|global\\.|asa|httpd\\.conf|boot\\.ini)\\|b\\|\\|Vetc\\|\\|)" at ARGS:page. [file "/etc/apache2/modsecurity-crs/base_rules/modsecurity_crs_40_generic_attacks.conf"] [line "205"] [id "950005"] [rev "3"] [msg "Remote File Access Attempt"] [data "Matched Data: /etc/ found within ARGS:page: ../../../../etc/passwd"] [severity "CRITICAL"] [ver "OWASP_CRS/2.2.9"] [maturity "9"] [accuracy "9"] [tag "OWASP_CRS/WEB_ATTACK/FILE_INJECTION"] [tag "WASCTC/WASC-33"] [tag "OWASP_TOP_10/A4"] [tag "PCI/6.5.4"] [hostname "10.2.0.101"] [uri "/dvwa/vulnerabilities/fi/"] [unique_id "XfUsin8AAQEAAgcUOwAAAAG"]
```

The Web server log for the above WAF log is below:

```
10.1.0.10 - - [14/Jul/2019:13:40:10 -0500] "GET /dvwa/vulnerabilities/fi/?page=../../../../etc/passwd HTTP/1.1" 302 20 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

The two logs can be connected as below with a conclusion of file injection attack from the WAF:

```
[Sat Jul 14 13:40:10 2019] [error] [client 10.1.0.10] ModSecurity: Warning. Pattern match "(?:<!\\w)(?:\\.(?:ht(?:access|passwd|group)|www_?acl)|global\\.|asa|httpd\\.conf|boot\\.ini)\\|b\\|\\|Vetc\\|\\|)" at ARGS:page. [file "/etc/apache2/modsecurity-crs/base_rules/modsecurity_crs_40_generic_attacks.conf"] [line "205"] [id "950005"] [rev "3"] [msg "Remote File Access Attempt"] [data "Matched Data: /etc/ found within ARGS:page: ../../../../etc/passwd"] [severity "CRITICAL"] [ver "OWASP_CRS/2.2.9"] [maturity "9"] [accuracy "9"] [tag "OWASP_CRS/WEB_ATTACK/FILE_INJECTION"] [tag "WASCTC/WASC-33"] [tag "OWASP_TOP_10/A4"] [tag "PCI/6.5.4"] [hostname "10.2.0.101"] [uri "/dvwa/vulnerabilities/fi/"] [unique_id "XfUsin8AAQEAAgcUOwAAAAG"]
```

```
10.1.0.10 - - [14/Jul/2019:13:40:10 -0500] "GET /dvwa/vulnerabilities/fi/?page=../../../../etc/passwd HTTP/1.1" 302 20 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

Analyzing WAF logs for a POST request

We have below log from a Web server:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

10.1.0.10 - - [14/Jul/2019:13:40:50 -0500] "POST /dvwa/login.php HTTP/1.1" 302 20
"http://10.2.0.101/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"

Equivalent WAF log detecting SQL injection is below:

```
[Sat Jul 14 13:40:50 2019] [error] [client 10.1.0.10] ModSecurity: Warning. Pattern match "(?:([\s\|\\"\\\xc2\x44\xe2\x80\x99\xe2\x80\x98\\\(\)\*\?)]|[\d\w]+)([\s\|\\"\\\xc2\x44\xe2\x80\x99\xe2\x80\x98\\\(\)\*\?])(?:|=|<=>|>|<|>|<|>|[\^]|\s+not ...) at ARGS:username. [file "/etc/apache2/modsecurity-crs/base_rules/modsecurity_crs_41_sql_injection_attacks.conf"] [line "77"] [id "950901"] [rev "2"] [msg "SQL Injection Attack: SQL Tautology Detected."] [data "Matched Data: 1=1 found within ARGS:username: admin' or 1=1#"] [severity "CRITICAL"] [ver "OWASP_CRS/2.2.9"] [maturity "9"] [accuracy "8"] [tag "OWASP_CRS/WEB_ATTACK/SQL_INJECTION"] [tag "WASCTC/WASC-19"] [tag "OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/CIE1"] [tag "PCI/6.5.2"] [hostname "10.2.0.101"] [uri "/dvwa/login.php"] [unique_id "XfUssn8AAQEAAAgeYI8AAAA"]
```

- **IPS/IDS and WAF Considerations**

- Helpful in protecting and identifying the attacks
- Working with attack signatures
- Usually has some built-in signatures
- Need adjustments, generally called as IPS/IDS tuning, after deployment
- Can be bypassed during attacks
- Can cause some availability issues like False Positives

- **IPv6 Considerations**

- IP is a lower layer protocol
- HTTP doesn't care if IP will use IPv4 or IPv6
- The only difference in the logs will be in the IP fields
- IPv4 is 32 bit and IPv6 is 128 bits

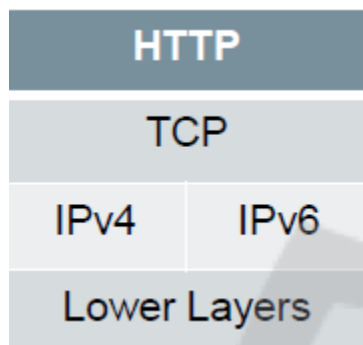
Positioning of HTTP and IP in layered stack would look like below:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY



Clearly, it doesn't matter which IP version is being used and we need to analyze in the same way for IPv6 as we have done so far for IPv4. The only difference will be in the IP field which will be a 128 bit address in case of IPv6 usage as visible in the log below:

```
2604:a880:400:d0::1 - - [13/Jul/2019:13:40:50 -0500] "GET /dvwa/login.php HTTP/1.1" 302 20 "-"  
"Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
```

- **Post-Assessment Question**

Analyze the log below to identify the key fields and the possible attack:

```
09/26-03:02:52.642180 [**] [1:2010517:6] ET WEB_SERVER Possible HTTP 404 XSS Attempt (Local  
Source) [**] [Classification: Web Application Attack] [Priority: 1] {TCP}  
66.147.244.197:80 -> 192.168.62.10:1138
```

```
[Sat Jul 14 13:40:50 2019] [error] [client 10.1.0.10] ModSecurity: Warning. Pattern match "(?:\b(?:\b|  
(?:\d{1,10})|[\w\W]*[^\d]{1,10})|[\w\W]*[^\d]{1,10})" ?[=<>]+(?:\s+x?or\s+.{1,20})+|\b-  
!<>=)]|\b(?:x?or)\b|\b|\b+(\d{1,10})|'[^=]{1,10}'|\b(?:x?or)\b|\b|\b+(\d{1,10})|'[^=]{1,10}'|\b|\b*?=[<>])  
" at ARGS:username. [file "/etc/apache2/modsecurity-  
crs/base_rules/modsecurity_crs_41_sql_injection_attacks.conf"] [line "133"] [id "959071"] [rev "2"] [msg  
"SQL Injection Attack"] [data "Matched Data: ' or 1= found within ARGS:username: admin' or 1=1#"]  
[severity "CRITICAL"] [ver "OWASP_CRS/2.2.9"] [maturity "9"] [accuracy "8"] [tag  
"OWASP_CRS/WEB_ATTACK/SQL_INJECTION"] [tag "WASCTC/WASC-19"] [tag  
"OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/CIE1"] [tag "PCI/6.5.2"] [hostname "10.2.0.101"] [uri  
"/dvwa/login.php"] [unique_id "XfUssn8AAQEAAAgeYI8AAAAI"]
```

The first log is XSS attack and second log is SQL injection attack as shown below:

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.

CYBRARY

09/26-03:02:52.642180 **[**]** [1:2010517:6] ET WEB_SERVER Possible HTTP 404 XSS Attempt (Local Source) **[**]** [Classification: Web Application Attack] [Priority: 1] {TCP} 66.147.244.197:80 -> 192.168.62.10:1138

```
[Sat Jul 14 13:40:50 2019] [error] [client 10.1.0.10] ModSecurity: Warning. Pattern match "(?:\b(?:\d{1,10}|[^\d]*[^\s]*\d{1,10}|[^\d]*[^\s]*[^\d]*[^\s]*\d{1,10})?[\<>]+(?:'\s+x?or\s+.{1,20}[+!<>=])\b(?:x?or)\b\s+(\d{1,10}|'[\^=]{1,10}')\b(?:x?or)\b\s+(\d{1,10}|'[\^=]{1,10}')\s*?[\<>]" at ARGS:username. [file "/etc/apache2/modsecurity-crs/base_rules/modsecurity_crs_41_sql_injection_attacks.conf"] [line "133"] [id "959071"] [rev "2"] [msg "SQL Injection Attack"] [data "Matched Data: ' or 1= found within ARGS:username: admin' or 1=#"] [severity "CRITICAL"] [ver "OWASP_CRS/2.2.9"] [maturity "9"] [accuracy "8"] [tag "OWASP_CRS/WEB_ATTACK/SQL_INJECTION"] [tag "WASCTC/WASC-19"] [tag "OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/CIE1"] [tag "PCI/6.5.2"] [hostname "10.2.0.101"] [uri "/dwa/login.php"] [unique_id "XfUssn8AAQEAAAgeYI8AAAAI"]
```

Supplementary Supplementary Material: Log analysis Practice

Most of the logs used during this course are available to download. There are 2 log files:

- Lab_logs.zip: Logs from the virtual lab. Inside there are two files:
 - LAB_accessLogs.txt
Contains the webserver logs from the course
 - LAB_errorLogs.txt
Contains the webserver error logs from the course
- Internet_logs.zip: Logs from webserver located on the Internet. Also there are two files inside the .zip:
 - Internet_accessLogs.txt
Contains the webserver logs from the course
 - Internet_errorLogs.txt
Contains the webserver error logs from the course

This webserver contains only one static web page and the web server is Nginx.

To use them just extract the .zip and open the file in a text editor like notepad. I like to use notepad++ (<https://notepad-plus-plus.org/downloads/>).

Brought to you by:

CYBRARY | FOR BUSINESS

Develop your team with the **fastest growing catalog** in the cybersecurity industry. Enterprise-grade workforce development management, advanced training features and detailed skill gap and competency analytics.