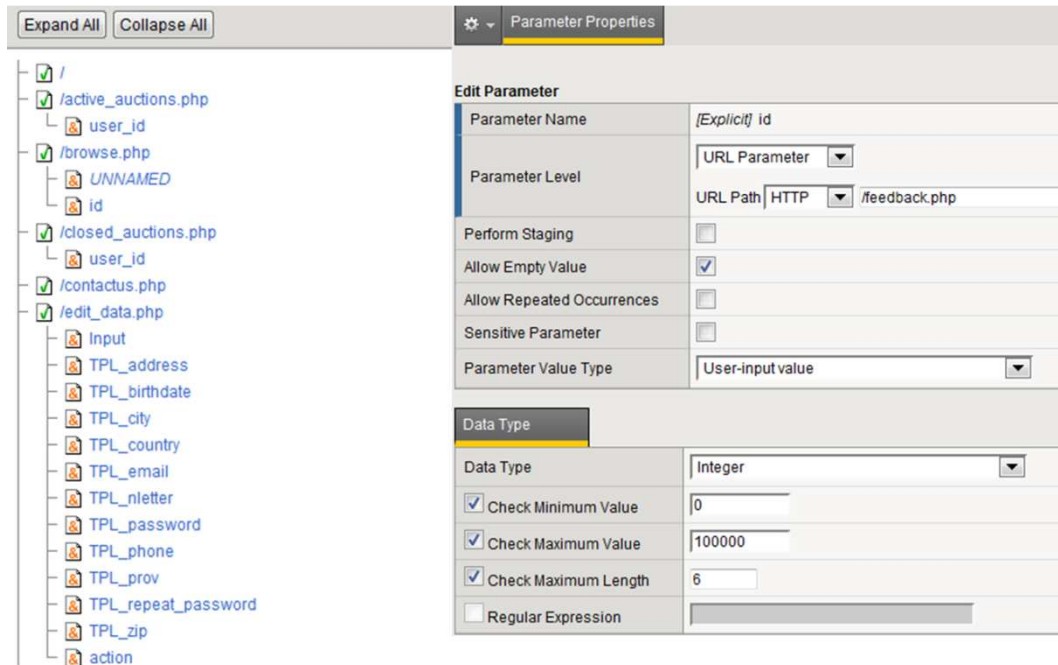


Analiza skuteczności zabezpieczeń przed atakami na aplikacje Web

Opracował: dr inż. Mariusz Stawowski
F5 Certified Product Consultant, ASM
Email: mariusz.stawowski@clico.pl

Zabezpieczenia sieciowe Firewall i Intrusion Prevention System (IPS) są podstawą do tworzenia architektury bezpieczeństwa sieci (m.in. stref bezpieczeństwa) oraz przeciwdziałania wielu groźnym atakom sieciowym (m.in. exploit na serwisy sieciowe i system operacyjny, propagacja robaków sieciowych, itd.). Światowe statystyki incydentów, jak również audyty bezpieczeństwa pokazują, że w przypadku ochrony aplikacji Web zabezpieczenia te nie są wystarczające. Wynika to z faktu, że aplikacje Web tworzone są zwykle na zamówienie firm w celu spełnienia ich indywidualnych wymagań biznesowych. W efekcie pisane na zamówienie aplikacje Web posiadają unikalne podatności, szczególnie w odniesieniu do logiki biznesowej aplikacji.

W praktyce nie ma możliwości napisania uniwersalnych sygnatur IPS do skutecznej identyfikacji podatności, które są specyficzne dla określonej aplikacji Web. Co prawda rodzaje ataków na aplikacje Web są znane (np. Cross Site Scripting, SQL-Injection), ale sposobów ich wykorzystania jest bardzo wiele. Skuteczną ochronę dla aplikacji Web zapewnia nowa kategoria zabezpieczeń Web Application Firewall (WAF). System zabezpieczeń WAF bazuje w głównej mierze na automatycznie tworzonym i aktualizowanym profilu chronionej aplikacji Web. WAF "uczy się" struktury aplikacji, URL, parametrów, cookie, itp. Tworzenie profilu ma na celu niezależne odwzorowanie oczekiwanych, poprawnych zachowań użytkowników przy dostępie do aplikacji Web. Poniższy rysunek pokazuje fragment profilu aplikacji Web utrzymywanego przez zabezpieczenia WAF.



The screenshot displays the 'Parameter Properties' configuration window for a WAF. On the left, a tree view shows the application structure with files like /active_auctions.php, /browse.php, /closed_auctions.php, /contactus.php, and /edit_data.php. The 'id' parameter is selected under /browse.php. The right pane shows the configuration for this parameter:

| Edit Parameter | |
|----------------------------|-------------------------------------|
| Parameter Name | [Explicit] id |
| Parameter Level | URL Parameter |
| URL Path | HTTP /feedback.php |
| Perform Staging | <input type="checkbox"/> |
| Allow Empty Value | <input checked="" type="checkbox"/> |
| Allow Repeated Occurrences | <input type="checkbox"/> |
| Sensitive Parameter | <input type="checkbox"/> |
| Parameter Value Type | User-input value |

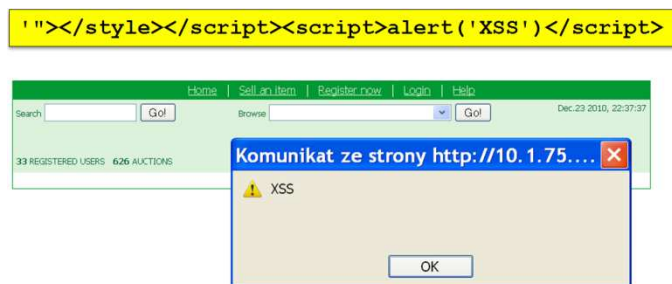
| Data Type | |
|----------------------|--------------------------|
| Data Type | Integer |
| Check Minimum Value | 0 |
| Check Maximum Value | 100000 |
| Check Maximum Length | 6 |
| Regular Expression | <input type="checkbox"/> |

W celu dokładniejszego zweryfikowania możliwości zabezpieczeń IPS i WAF w zakresie ochrony aplikacji Web zostały wykonane ataki na testową aplikację Web, wykorzystujące rzeczywiste błędy tej aplikacji. Do testów zostało użyte rozwiązanie WAF firmy F5 Networks (moduł Application Security Manager w urządzeniu BIG-IP) oraz rozwiązanie IPS jednego z wiodących światowych producentów zabezpieczeń (włączone wszystkie dostępne techniki detekcji i sygnatury).

W trakcie testów zostały wykonane następujące ataki na aplikację Web:

1. Prosty XSS Reflected
2. XSS Reflected z enkodowaniem znaków
3. Prosty XSS Stored
4. Kombinacje XSS Stored
5. XSS Stored z enkodowaniem znaków
6. Prosty SQL-Injection
7. SQL-Injection z kombinacją zapytania SQL
8. Manipulacja parametru aplikacji Web
9. Forceful Browsing
10. Information Leakage
11. Atak D/DoS (zalewanie zapytaniami HTTP)
12. Atak brutalny na login aplikacji Web

Test 1. Prosty atak XSS Reflected



1a. Reakcja IPS

-- IPS wykrywa elementy skryptu w parametrze URL, brak dokładniejszych danych.

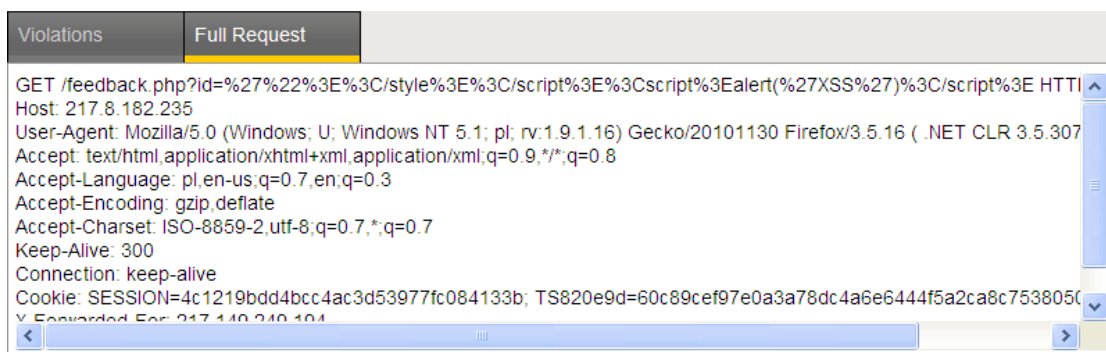
| | |
|-------------|--|
| Category | Predefined |
| Subcategory | HTTP: HTML Script Tag Embedded in URL Parameters |
| Severity | Major |
| Device | IPS |

1b. Reakcja WAF

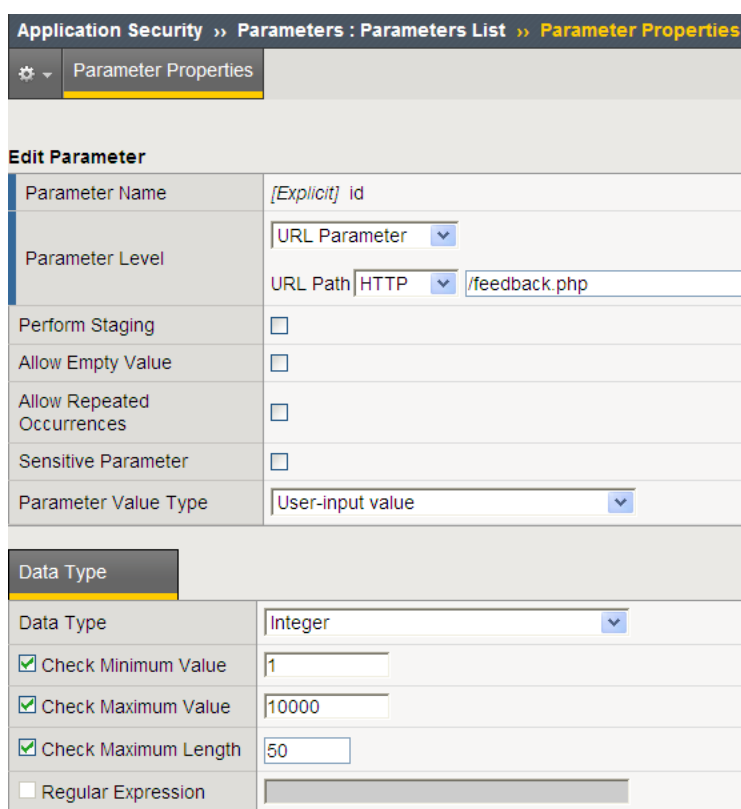
-- WAF wykrywa i pokazuje nielegalną wartość parametru w URL.

| Illegal parameter data type violation details | | | | | | | | | | | | | |
|---|--|------------|--------------|-----------|----------|-------|-------|-------|-----------------------------|-------|----|-----|----|
| Parameter Level | URL | | | | | | | | | | | | |
| Parameter Name | id | | | | | | | | | | | | |
| Parameter Value | "></style></script><script>alert('XSS')</script> | | | | | | | | | | | | |
| Expected Data Type | Integer | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Violations</th> <th>Full Request</th> </tr> </thead> <tbody> <tr> <td>Violation</td> <td>Severity</td> <td>Learn</td> <td>Alarm</td> <td>Block</td> </tr> <tr> <td>Illegal parameter data type</td> <td>Error</td> <td>No</td> <td>Yes</td> <td>No</td> </tr> </tbody> </table> | | Violations | Full Request | Violation | Severity | Learn | Alarm | Block | Illegal parameter data type | Error | No | Yes | No |
| Violations | Full Request | | | | | | | | | | | | |
| Violation | Severity | Learn | Alarm | Block | | | | | | | | | |
| Illegal parameter data type | Error | No | Yes | No | | | | | | | | | |
| Requested URL | [HTTP] /feedback.php | | | | | | | | | | | | |
| Web Application | Clico | | | | | | | | | | | | |
| Support ID | 1533487467099849418 | | | | | | | | | | | | |
| Source IP Address | 217.149.249.194:2531 | | | | | | | | | | | | |
| Destination IP Address | 217.8.182.235:80 | | | | | | | | | | | | |
| Country | Poland | | | | | | | | | | | | |
| Time | 2011-02-11 16:21:57 | | | | | | | | | | | | |
| Flags | ✗ | | | | | | | | | | | | |
| Severity | Error | | | | | | | | | | | | |
| Response Status Code | 200 | | | | | | | | | | | | |
| Potential Attacks | Parameter Tampering | | | | | | | | | | | | |

-- Administrator WAF może dokładnie przeanalizować zapytanie.

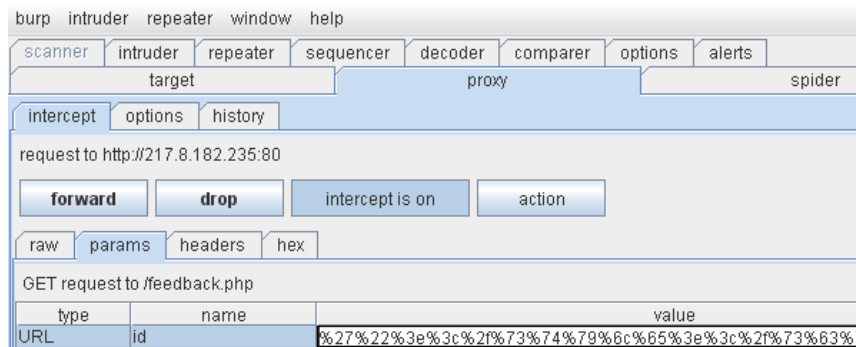
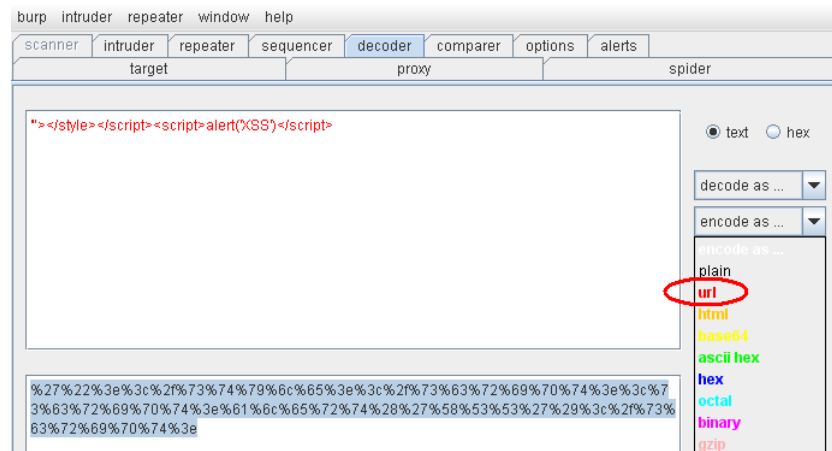


-- W razie potrzeby administrator WAF może dobrać wyuczony profil.



Test 2. XSS Reflected z enkodowaniem znaków

-- Kod XSS enkodujemy w URL i przekazujemy do aplikacji Web.

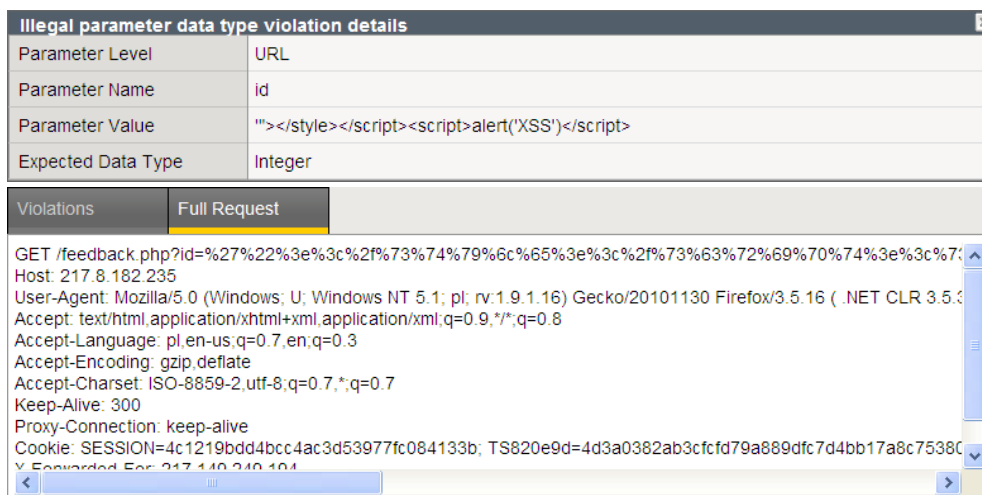


2a. Reakcja IPS

Nie wykrywa ataku.

2b. Reakcja WAF

-- Wykrywa atak i pokazuje zastosowane enkodowanie.



Test 3. Prosty XSS Stored

```
<script language="javascript">
document.write('');
</script>
```

-- W formularzu Web wpisujemy kod do pobrania pliku graficznego z zewnętrznego serwera.

Sell an item

Item title

Item description (HTML allowed)

-- W efekcie użytkownikowi aplikacji Web wyświetla się plik graficzny pobrany z Internetu.

3a. Reakcja IPS

Nie wykrywa ataku.

3b. Reakcja WAF

-- WAF wykrywa nielegalną zawartość zapytania URL.

| Violations | Full Request | Severity | Learn | Alarm | Block |
|---|--|----------|-------|-------|-------|
| Attack signature detected | Learn | Error | Yes | Yes | Yes |
| Failed to convert character | Learn | Error | No | Yes | No |
| Illegal meta character in parameter value | Learn | Error | No | Yes | Yes |
| Illegal parameter value length | Learn | Error | No | Yes | Yes |
| Requested URL | [HTTP] /sell.php | | | | |
| Web Application | Clico | | | | |
| Support ID | 1533487467099849495 | | | | |
| Source IP Address | 217.149.249.194:1890 | | | | |
| Destination IP Address | 217.8.182.235:80 | | | | |
| Country | Poland | | | | |
| Time | 2011-02-11 17:23:24 | | | | |
| Flags | ✘👤 | | | | |
| Severity | Error | | | | |
| Response Status Code | N/A | | | | |
| Potential Attacks | Cross Site Scripting (XSS) | | | | |

-- Administrator WAF może dokładnie przeanalizować powody zablokowania zapytania URL.

---- nielegalna wielkość danych wejściowych,

| Illegal parameter value length violation details | |
|--|---|
| Parameter Level | URL |
| Parameter Name | description |
| Parameter Value | NIESPODZIANKA[0xd] [0xd] <script[0x20]language="javascript">[0xd] [0xd] document.write('<img[0x20]src="http://www.clico.pl/logo.gif">'); [0xd] [0xd] </script> |
| Detected Value Length | 123 |
| Expected Value Length | 100 |

---- nielegalna zawartość danych wejściowych,

| Illegal meta character in parameter value violation | | |
|---|-----|---------------------------------|
| Details | Hex | Details |
| CR | 0xd | View details... |

| Parameters with CR (0xd) meta character in value | |
|--|---|
| Parameter Level | URL |
| Parameter Name | description |
| Parameter Value | NIESPODZIANKA[0xd][0xa][0xd][0xa]<script[0x20]language="javasc ript">[0xd][0xa][0xd][0xa]document.write('<img[0x20]src="http ://www.clico.pl/logo.gif">');[0xd][0xa][0xd][0xa]</scrip t> |

---- identyfikacja ataku XSS poprzez sygnatury.

| Attack signature detected violation details | | | | | | |
|---|--------------|-------|-------|-------|---------------------------------|--|
| Signature Name | Signature ID | Learn | Alarm | Block | Details | |
| XSS script tag (Parameter) | 200000098 | Yes | Yes | Yes | View details... | |
| XSS script tag end (Parameter) | 200000092 | Yes | Yes | Yes | View details... | |
| document.write (Parameter) | 200001356 | Yes | Yes | Yes | View details... | |
| img tag: src/dynsrc (Parameter) | 200000128 | Yes | Yes | Yes | View details... | |
| src http: (Parameter) | 200001139 | Yes | Yes | Yes | View details... | |

| Context Details for Attack Signature 200000092 | |
|--|--|
| Context | Parameter |
| Parameter Level | URL |
| Parameter Name | description |
| Parameter Value | NIESPODZIANKA[0xd] [0xd] <script[0x20]language="javascript">[0xd] [0xd] document.write('<img[0x20]src="http://www.clico.pl/logo.gif">'); [0xd] [0xd] </script> |
| Detected Keywords | description=NIESPODZIANKA[0xd][0xa][0xd][0xa]<script[0x20]lang uage="java[script]">[0xd][0xa][0xd][0xa]document.write('<i mg[0x20]src="http://www.clico.pl/logo.gif">') [0xd][0xa][0xd][0xa]</script> |

Test 4. Kombinacje XSS Stored

Atak wykonujemy na takiej samej zasadzie jak w teście 3.

-- Wykorzystujemy funkcje onload, onmouseover i onerror.

```
<body onload=alert('XSS')>
```

```
<b onmouseover="window.location.href='http://www.clico.pl';">
!!! ODCZYTAJ SZCZEGOLY !!!</b>
```

```

```

4a. Reakcja IPS

Nie wykrywa ataku.

4b. Reakcja WAF

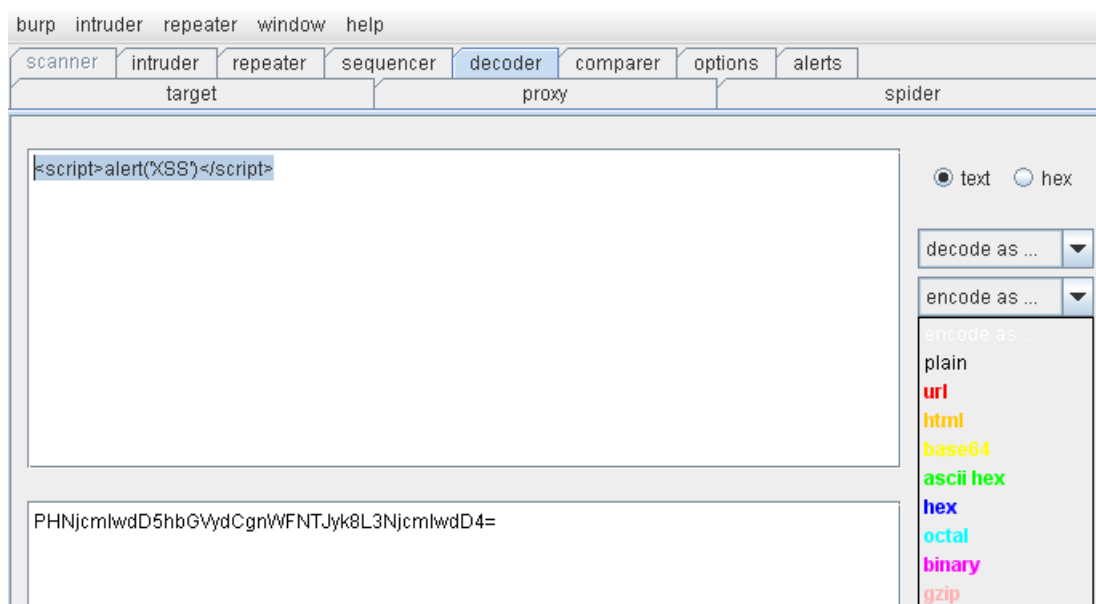
-- WAF wykrywa nielegalną zawartość zapytania URL.

| Violations | Full Request |
|---|--|
| Violation | |
| Attack signature detected | Learn |
| Failed to convert character | Learn |
| Illegal meta character in parameter value | Learn |
| Illegal parameter value length | Learn |
| Requested URL | [HTTP] /sell.php |
| Web Application | http_secured |
| Support ID | 5375709212317845459 |
| Source IP Address | 10.1.75.1:2153 |
| Destination IP Address | 10.1.75.102:80 |
| Country | N/A |
| Time | 2011-03-02 11:12:58 |
| Flags | ✘ |
| Severity | Error |
| Response Status Code | 200 |
| Potential Attacks | Cross Site Scripting (XSS) |

| Violations | Full Request |
|--|--------------|
| -----41184676334 | |
| Content-Disposition: form-data; name="description" | |
| ↗ | |
| <b onmouseover="window.location.href='http://www.clico.pl';">!!! ODCZYTAJ SZCZEGOLY !!! | |

Test 5. XSS Stored z enkodowaniem znaków

-- Skrypt enkodujemy w base64 i umieszczamy w META Tagu.



```
<META HTTP-EQUIV="refresh"
CONTENT="0 ;url=data :text/html ;base64 ,PHNjcmlwdD5hbGVydCgnWFNTJyk8L3NjcmlwdD4=">
```

5a. Reakcja IPS

Nie wykrywa ataku.

5b. Reakcja WAF

-- Wykrywa atak i pokazuje zastosowane enkodowanie /jak dla poprzedniego ataku/.

| Violations | Full Request |
|---|-----------------------|
| Violation | |
| Attack signature detected | Learn |
| Failed to convert character | Learn |
| Illegal meta character in parameter value | Learn |
| Illegal parameter value length | Learn |

| Attack signature detected violation details | | | | | | |
|---|--------------|-------|-------|-------|---------------------------------|--|
| Signature Name | Signature ID | Learn | Alarm | Block | Details | |
| meta tag (Parameter) | 200000137 | Yes | Yes | No | View details... | |
| data: base64 (Parameter) (2) | 200001483 | Yes | Yes | No | View details... | |
| data: base64 (Parameter) | 200001312 | Yes | Yes | No | View details... | |

Test 6. Prosty SQL-Injection

Wprowadzamy zapytanie SQL w pole logowania aplikacji Web i uzyskujemy nieupoważniony dostęp do aplikacji.

User's login

Your name

Password

[Forgot your password?](#)

6a. Reakcja IPS

-- IPS wykrywa atak SQL-Injection w URL, brak dokładniejszych informacji o ataku.

| | |
|-------------|----------------------------|
| Category | Predefined |
| Subcategory | HTTP: SQL Injection In URL |
| Severity | Major |
| Device | IPS |

6b. Reakcja WAF

-- Wykrywa atak i pokazuje szczegółowe informacje na jego temat /jak dla poprzednich ataków/.

| Violations | Full Request | | | | |
|---|--------------------------------------|-------|-------|-------|---------------------------------|
| Violation | | | | | |
| Attack signature detected | <input type="button" value="Learn"/> | | | | |
| Failed to convert character | <input type="button" value="Learn"/> | | | | |
| Illegal meta character in parameter value | <input type="button" value="Learn"/> | | | | |
| Illegal parameter value length | <input type="button" value="Learn"/> | | | | |
| Requested URL | [HTTP] /user_login.php | | | | |
| Web Application | http_secured | | | | |
| Support ID | 5375709212317845487 | | | | |
| Source IP Address | 10.1.75.1:2842 | | | | |
| Destination IP Address | 10.1.75.102:80 | | | | |
| Country | N/A | | | | |
| Time | 2011-03-02 11:24:11 | | | | |
| Flags | ✘ | | | | |
| Severity | Error | | | | |
| Response Status Code | 302 | | | | |
| Potential Attacks | SQL-Injection | | | | |
| Attack signature detected violation details | | | | | |
| Signature Name | Signature ID | Learn | Alarm | Block | Details |
| SQL-INJ expressions like "or 1=1" (3) | 200002147 | Yes | Yes | No | View details... |
| SQL-INJ expressions like "' or 1 --" | 200002419 | Yes | Yes | No | View details... |

| Context Details for Attack Signature 200002147 | |
|--|------------------------------|
| Context | Parameter |
| Parameter Level | URL |
| Parameter Name | password |
| Parameter Value | ***** |
| Context | Parameter |
| Parameter Level | URL |
| Parameter Name | username |
| Parameter Value | '0x20or0x201=10x20# |
| Detected Keywords | username='0x20or0x201=10x20# |

| Illegal meta character in parameter value violation detail | | |
|--|------|---------------------------------|
| Char | Hex | Details |
| ' | 0x27 | View details... |
| Space | 0x20 | View details... |

Test 7. SQL-Injection z kombinacją zapytania SQL

Wprowadzamy zapytanie SQL w zmodyfikowanej formie.

User's login

Your name

Password

[Forgot your password?](#)

7a. Reakcja IPS

Nie wykrywa ataku.

7b. Reakcja WAF

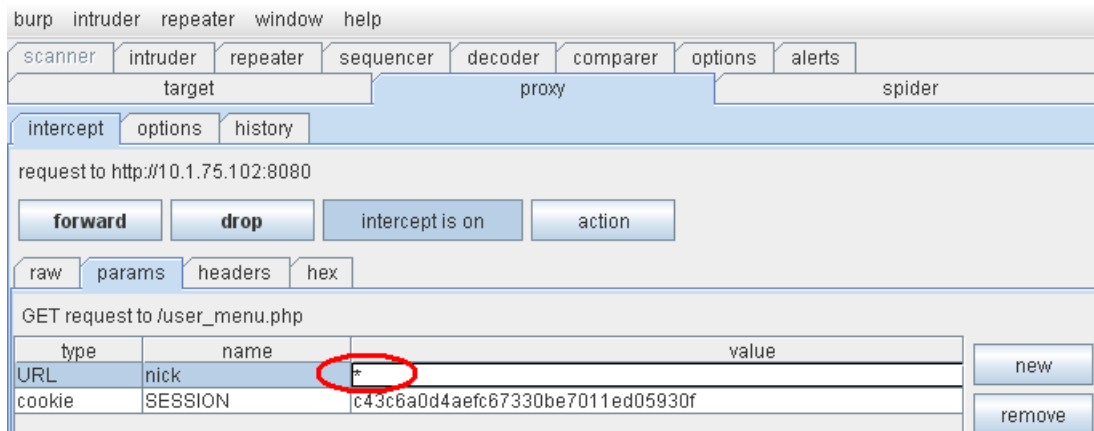
-- Wykrywa atak i pokazuje szczegółowe informacje na jego temat /jak dla poprzednich ataków/.

| Context Details for Attack Signature 200002430 | |
|--|---------------------------------|
| Context | Parameter |
| Parameter Level | URL |
| Parameter Name | password |
| Parameter Value | ***** |
| Context | Parameter |
| Parameter Level | URL |
| Parameter Name | username |
| Parameter Value | '0x20or0x20'a='a'0x20# |
| Detected Keywords | username='0x20or0x20'a='a'0x20# |

Test 8. Manipulacja parametru aplikacji Web

Przechwyтуjemy zapytanie do aplikacji Web i modyfikujemy wartość parametru.

-- Modyfikujemy wartość parametru na znak wildcard (*).





-- W efekcie nielegalnie odczytujemy dane wszystkich zarejestrowanych użytkowników.

8a. Reakcja IPS

Nie wykrywa ataku.

8b. Reakcja WAF

-- Wykrywa atak i pokazuje szczegółowe informacje na jego temat /jak dla poprzednich ataków/.

| Violations | Full Request |
|---|-------------------------------------|
| Violation | |
|  Illegal meta character in parameter value | Learn |
|  Information leakage detected | Learn |
| Requested URL | [HTTP] /user_menu.php |
| Web Application | http_secured |
| Support ID | 5375709212317845551 |
| Source IP Address | 10.1.75.1:2798 |
| Destination IP Address | 10.1.75.102:80 |
| Country | N/A |
| Time | 2011-03-02 11:35:13 |
| Flags | ✘ |
| Severity | Error |
| Response Status Code | 200 |
| Potential Attacks | Information Leakage |

| Illegal meta character in parameter value violation detail ✕ | | |
|---|------|---------------------------------|
| Char | Hex | Details |
| * | 0x2a | View details... |

| Parameters with * (0x2a) meta character in value | |
|--|------|
| Parameter Level | URL |
| Parameter Name | nick |
| Parameter Value | * |

| Information leakage detected violation details | |
|--|------------------|
| Detected Pattern | Context |
| Credit Card Number | nbsp;***** |

Test 9. Forceful Browsing

Bez logowania do aplikacji wywołujemy URL strony z konfiguracją PHP </includes/config.inc.php.old>

9a. Reakcja IPS

Nie wykrywa ataku.

9b. Reakcja WAF

-- Wykrywa atak i pokazuje szczegółowe informacje na jego temat /jak dla poprzednich ataków/.

| Violations | Full Request |
|--|--|
| Violation | |
| ⓘ Illegal URL Learn | |
| Requested URL | [HTTP] /includes/config.inc.php.old |
| Web Application | http_secured |
| Support ID | 5375709212317845555 |
| Source IP Address | 10.1.75.1:2600 |
| Destination IP Address | 10.1.75.102:80 |
| Country | N/A |
| Time | 2011-03-02 11:38:32 |
| Flags | ✘ |
| Severity | Error |
| Response Status Code | 200 |
| Potential Attacks | Forceful Browsing |

Test 10. Wyciek poufnych danych (Information Leakage)

Odczyt numerów kart kredytowych z bazy danych za pomocą ataku z testu 8.

10a. Reakcja IPS

Nie wykrywa ataku.

10b. Reakcja WAF

-- WAF maskuje numery kart kredytowych i wykrywa atak oraz pokazuje szczegółowe informacje na jego temat /jak dla poprzednich ataków/.

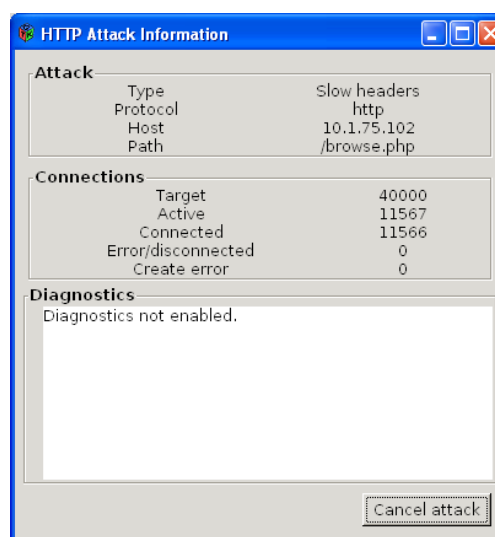
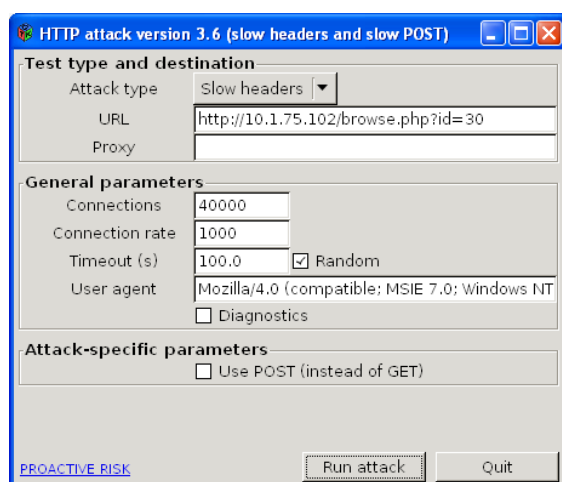
| Name | Credit Card | Email | Tel | Address | City | Country |
|-------|-------------|------------|-------|---------|-----------|---------|
| adamk | ***** | sadadasads | 12312 | asdfgh | asdfgh123 | 3 |

Test 11. Atak D/DoS (zalewanie zapytaniami HTTP)

Do wykonania testu DoS na aplikację Web można wykorzystać różne ogólnie dostępne narzędzia, np. aplikację OWASP HTTP Post Tool¹.

Ataki D/DoS skierowane na konkretną aplikację Web wykonywane z wykorzystaniem odpowiednio dobranych zapytań HTTP potrafią spowodować jej niedostępność za pomocą niewielkiej liczby zapytań, spreparowanych tak, aby maksymalnie angażować CPU i pamięć serwera. Konwencjonalne zabezpieczenia firewall i IPS nie mają możliwości zablokowania takich ataków DoS, ponieważ liczba otwieranych sesji nie jest duża i są to legalne zapytania do aplikacji.

Identyfikacja aplikacyjnych ataków D/DoS w systemach WAF jest możliwa poprzez analizę czasu odpowiedzi serwera (opóźnienia) oraz częstości napływających zapytań do serwera. Istotne zwiększenie opóźnienia w odpowiedziach serwera lub częstości zapytań napływających do serwera może wskazywać na atak D/DoS. Analiza ruchu dokonywana jest w odniesieniu do danych historycznych. Ataki D/DoS wykonywane są za pomocą narzędzi. Nie odbywa się to z wykorzystaniem przeglądarek Web. Skuteczną metodą ochrony jest identyfikacja czy zapytania do aplikacji wysyłane są przez przeglądarkę, czy inne narzędzia. Zabezpieczenia WAF posiadają zaimplementowane metody rozpoznawania czy zapytania wysyłane są przez przeglądarki Web (m.in. dodają kod JavaScript do odpowiedzi serwera i weryfikują kolejne zapytania) i umożliwiają filtrowanie ruchu generowanego przez inne narzędzia.



¹ OWASP HTTP Post Tool – aplikacja do generowania zapytań do aplikacji Web. Więcej informacji dostępne na stronie: <http://www.pentestit.com/2010/11/28/owasp-http-post-tool/>

11a. Reakcja IPS

Nie wykrywa ataku.

11b. Reakcja WAF

-- WAF identyfikuje dużą liczbę zapytań do strony browse.php, wykrywa zwiększenie opóźnienia odpowiedzi serwera i pokazuje statystykę ataku DoS.

Application Security >> Reporting : Anomaly Statistics : DoS Attacks

Requests | Charts | **Anomaly Statistics** | PCI Compliance | CPU Utilization

Filter: Show All [Go]

DoS Attacks

| <input type="checkbox"/> | Web Application | Start Time | End Time |
|--------------------------|-----------------|---------------------|---------------------|
| <input type="checkbox"/> | Ochrona2 | 2011-03-17 10:37:08 | 2011-03-17 10:39:14 |

| | |
|----------------------|----------------------|
| Legitimate Latency | 19 ms |
| Detected Latency | 15 ms |
| Latency End Time | 18 ms |
| Dropped Connections | 0 |
| Current Mitigation | N/A |
| Previous Mitigations | N/A |
| IP Addresses | View |
| URLs | View |

| URL | Legitimate |
|--------------------|------------|
| [HTTP] /browse.php | 0 |

Test 12. Atak brutalny na login aplikacji Web

Do wykonania testu brutalnego na login aplikacji Web można wykorzystać różne ogólnie dostępne narzędzia, np. aplikację THC-Hydra².

```
hydra -L users.txt -P words.txt 10.1.75.102 http-post-form
"/user_login.php:username=^USER^&password=^PASS^&=Wy%B6lij+zapytanie&action
=login:Login incorrect" &
```

W celu ochrony aplikacji Web przed atakami odgadywania haseł zabezpieczenia WAF muszą wiedzieć, która strona i jakie parametry służą do logowania do aplikacji. Administrator WAF wprowadza takie informacje w konfiguracji.

| Brute Force Protection Configuration | |
|---|-------------------------------|
| Brute Force Protection Configuration | |
| Login URL | Explicit HTTP /user_login.php |
| Authentication Type | HTML Form |
| Username Parameter Name | username |
| Password Parameter Name | password |
| Session-based Brute Force Protection (Blocking Settings) | |
| Login Attempts From The Same Client | 3 |
| Re-enable Login After | 6 seconds |

Administrator WAF ustala także kryteria i sposób identyfikacji ataku na login aplikacji Web.

| Dynamic Brute Force Protection | |
|--------------------------------------|--|
| Operation Mode | Transparent |
| Detection Criteria | Failed Login Attempts increased by 10 % |
| | Failed Login Attempts Rate reached 11 per second |
| | Minimum Failed Login Attempts 10 per second |
| Prevention Policy | <input type="checkbox"/> Source IP-Based Client Side Integrity Defense |
| | <input type="checkbox"/> URL-Based Client Side Integrity Defense |
| | <input checked="" type="checkbox"/> Source IP-Based Rate Limiting |
| | <input checked="" type="checkbox"/> URL-Based Rate Limiting |
| Suspicious Criteria (per IP address) | Failed Login Attempts increased by 10 % |
| | Failed Login Attempts Rate reached 10 per second |
| Prevention Duration | <input checked="" type="radio"/> Unlimited <input type="radio"/> Maximum 0 seconds |
| IP Address Whitelist | IP Address <input type="text"/> |
| | Subnet Mask <input type="text"/> <input type="button" value="Add"/> |
| | <input type="text"/> |
| | <input type="button" value="Delete"/> |

| Access Validation | |
|--|----------------------|
| A string that should appear in the response | <input type="text"/> |
| A string that should NOT appear in the response | Login incorrect |
| Expected HTTP response status code | <input type="text"/> |
| Expected validation header name and value (for example, Location header) | <input type="text"/> |
| Expected validation domain cookie name | <input type="text"/> |
| Expected parameter name (added to URI links in the response) | <input type="text"/> |

² THC-Hydra – aplikacja do wykonywania różnego rodzajów ataków brute force, także na aplikacje Web. Więcej informacji dostępne na stronie: <http://freeworld.thc.org/thc-hydra/>

12a. Reakcja IPS

Nie wykrywa ataku.

13b. Reakcja WAF

-- WAF identyfikuje atak na login aplikacji Web i pokazuje statystykę ataku.

Application Security » Reporting : Anomaly Statistics : Brute Force Attacks


Requests | Charts | **Anomaly Statistics** | PCI Compliance | CPU Utilization

Filter Show All Go

Brute Force Attacks

| <input type="checkbox"/> | Web Application | Login URL | Start Time | End Time | Details |
|--------------------------|-----------------|------------------------|---------------------|---------------------|----------------------|
| <input type="checkbox"/> | Ochrona2 | [HTTP] /user_login.php | 2011-03-16 16:28:06 | 2011-03-16 16:32:54 | View |

| | |
|----------------------------------|----------------------|
| Average Historical Failed Logins | 1 |
| Detected Failed Logins | 10 |
| Dropped Connections | 0 |
| Current Mitigation | N/A |
| Previous Mitigations | N/A |
| IP Addresses | View |

| IP Address | Average Historical Failed Logins | Detected Failed Logins | Dropped Connections |
|---|----------------------------------|------------------------|---------------------|
|  10.1.75.1 | 1 | 10 | 0 |

Podsumowanie testów

Poniżej zamieszczona tabelka przedstawia podsumowanie testów skuteczności zabezpieczeń IPS i WAF przed atakami na aplikację Web.

| Lp. | Atak na aplikację Web | IPS | WAF |
|-----|--|--|---|
| 1. | Prosty XSS Reflected | <ul style="list-style-type: none"> Atak wykryty jako skrypt w parametrze URL Brak dokładniejszych informacji o ataku | <ul style="list-style-type: none"> Wykryta nielegalna wartość parametru HTTP GET Szczegółowe informacje o ataku |
| 2. | XSS Reflected z enkodowaniem znaków | <ul style="list-style-type: none"> Nie wykrywa ataku³ | <ul style="list-style-type: none"> Wykryta nielegalna wartość parametru URL Szczegółowe informacje o ataku, w tym pokazuje zastosowane enkodowane |
| 3. | Prosty XSS Stored | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Wykryta nielegalna zawartość zapytania HTTP POST Szczegółowe informacje o ataku |
| 4. | Kombinacje XSS Stored | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Wykryta nielegalna zawartość zapytania HTTP POST Szczegółowe informacje o ataku |
| 5. | XSS Stored z enkodowaniem znaków | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Wykryta nielegalna zawartość zapytania HTTP POST Szczegółowe informacje o ataku, w tym pokazuje zastosowane enkodowane |
| 6. | Prosty SQL-Injection | <ul style="list-style-type: none"> Wykryty atak SQL-Injection w URL Brak dokładniejszych informacji o ataku | <ul style="list-style-type: none"> Wykryta nielegalna wartość parametru HTTP GET Szczegółowe informacje o ataku |
| 7. | SQL-Injection z kombinacją zapytania SQL | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Wykryta nielegalna wartość parametru HTTP GET Szczegółowe informacje o ataku |
| 8. | Manipulacja parametru aplikacji Web | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Wykryta nielegalna wartość parametru Szczegółowe informacje o ataku |
| 9. | Forceful browsing | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Wykrywa dostęp do nieznanego URL |
| 10. | Wyciek informacji (Information Leakage) | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Identyfikuje atak i maskuje poufne dane |
| 11. | Atak D/DoS (zalewanie zapytaniami HTTP) | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Identyfikuje dużą liczbę zapytań do okr. URL, wykrywa zwiększenie opóźnienia odpowiedzi serwera i pokazuje statystykę ataku DoS |
| 12. | Atak brutalny na login aplikacji Web | <ul style="list-style-type: none"> Nie wykrywa ataku | <ul style="list-style-type: none"> Identyfikuje atak na login aplikacji Web i pokazuje statystykę ataku |

W niniejszych testach zostały wykorzystane najbardziej popularne i proste do wykonania ataki. Aplikacje Web mogą zostać poddane wielu innym atakom, dla których konwencjonalne zabezpieczenia (FW, IPS, UTM) także nie stanowią zabezpieczenia, m.in.: Cross Site Request Forgery (CSRF), nielegalny dostęp do plików i danych serwera Web (enumeracja plików), cookie poisoning, manipulacja ukrytych pól.

³ Ten i inne przedstawione w tabelce ataki Web mogłyby zostać wykryte przez IPS po napisaniu odpowiednich sygnatur dla tych konkretnych ataków. Do napisania sygnatur IPS wymagane są jednak informacje jak ataki zostaną wykonane. W rzeczywistości jest to niemożliwe, ponieważ producenci IPS nie wiedzą jakie dokładnie błędy mają napisane dla firm aplikacje Web i w jaki sposób te błędy mogą zostać wykorzystane. Ataki XSS, SQL-Injection, itd. mogą zostać wykonane na wiele różnych sposobów. Nie ma możliwości napisać dla wszystkich tych ataków sygnatur IPS. Dlatego też IPS za pomocą tzw. uniwersalnych sygnatur wykrywa tylko podstawowe ataki Web wykonywane w typowy sposób.