



Александар Јевремовић  
Младен Веиновић  
Марко Шарац  
Горан Шимић

# ЗАШТИТА У РАЧУНАРСКИМ МРЕЖАМА

Београд, 2018.

*УНИВЕРЗИТЕТ СИНГИДУНУМ*  
*Факултет за информатику и рачунарство*

*Александар Јевремовић*  
*Младен Веиновић*  
*Марко Шарац*  
*Горан Шимић*

# **Заштита у рачунарским мрежама**

*Друго издање*

*Београд, 2018.*

---

# ЗАШТИТА У РАЧУНАРСКИМ МРЕЖАМА

*Аутори:*

*др Александар Јевремовић*

*др Младен Веиновић*

*др Марко Шарац*

*др Горан Шимић*

*Рецензенти:*

*др Милан Милосављевић*

*др Бранко Ковачевић*

*др Саша Адамовић*

*Издавач:*

*УНИВЕРЗИТЕТ СИНГИДУНУМ*

*Београд, Данијелова 32*

*www.singidunum.ac.rs*

*За издавача:*

*Проф. др Милован Станишић*

*Лектор:*

*Данило Јевремовић*

*Техничка обрада:*

*Александар Јевремовић*

*Дизајн корица:*

*Александар Михајловић*

*Година издања:*

*2018.*

*Тираж:*

*750 примерака*

*Штампа:*

*Калиграф, Београд*

*ISBN: 978-86-7912-565-1*

*Copyright:*

*© 2018. Универзитет Сингидунум*

*Издавач задржава сва права.*

*Репродукција појединих делова или целине ове публикације није дозвољена.*

# Садржај

1. Увод.....	1
1.1. Криптолошке основе.....	1
1.2. Димензије напада.....	2
1.3. Фазе напада и одбране.....	6
1.3.1. Напади извиђања.....	7
1.3.2. Анализа напада у филму „Матрица“.....	12
1.4. Прислушкивање и снимање саобраћаја.....	14
1.5. Типови нападача.....	15
2. Контрола приступа у рачунарским мрежама.....	16
2.1. Аутентификација и ауторизација корисника.....	16
2.1.1. Основни модел аутентификације.....	17
2.1.2. Вишефакторска аутентификација.....	18
2.1.3. Системи за разликовање робота и људи.....	20
2.1.4. Основни модел ауторизације.....	22
2.1.5. Контрола приступа заснована на тикетима.....	24
2.2. Филтери пакета.....	25
2.2.1. Хардверски и софтверски филтери пакета.....	28
2.2.2. Филтрирање пакета у Линукс оперативном систему.....	30
2.2.3. Нивои рада филтера пакета.....	35
2.2.4. Филтери пакета и тројански коњи.....	38
2.2.5. Концепт демилитаризоване зоне.....	39
2.2.6. Прокси сервери.....	41
2.3. Системи за откривање и спречавање напада.....	43
2.4. Бележење активности у дневнике догађаја.....	43
2.5. Контрола приступа приватним мрежама.....	45
2.5.1. Виртуалне мреже на локалном подручју.....	46
2.5.2. Контрола приступа путем IEEE 802.1X протокола.....	49
3. Безбедност на физичком нивоу.....	50
3.1. Компромитијуће електромагнетно зрачење.....	53
3.2. Пасивни мрежни вентили.....	54
3.3. Шифровани системи фајлова.....	56
4. Безбедност приватних рачунарских мрежа.....	58
4.1. Напади у Етернет мрежама.....	59
4.1.1. Тровање ARP кеша.....	59
4.1.2. Препуњавање адресне меморије комутатора.....	61
4.1.3. Напади на STP протокол.....	62
4.1.4. Напад прескакањем у друге виртуалне локалне мреже.....	64
4.2. Безбедност у бежичним рачунарским мрежама.....	65
4.2.1. Контрола приступа и прислушкивање.....	66
4.2.2. Проблем вишеструко умрежених рачунара.....	68

4.3. Безбедност Bluetooth мрежа.....	69
4.4. Напади на DHCP сервис.....	71
4.4.1. Лажни DHCP сервер.....	72
4.4.2. Испорљивање адреса.....	75
5. Заштита међумрежних комуникација.....	77
5.1. Повезивање удаљених приватних мрежа.....	79
5.2. Виртуалне приватне мреже.....	82
5.2.1. Нивои рада виртуалних приватних мрежа.....	83
5.2.2. Безбедносне функције виртуалних приватних мрежа.....	85
6. Безбедност система доменских имена.....	87
6.1. Основни принципи рада система доменских имена.....	87
6.1.1. Кеширање код система доменских имена.....	91
6.2. Напади путем хостс фајла.....	93
6.3. Тровање кеша DNS сервера.....	94
6.4. Измена информација на серверу.....	94
6.5. Напади извиђања и DNS сервис.....	95
6.5.1. DNS Cache Snooping.....	96
6.6. Безбедносна проширења система доменских имена.....	97
7. Безбедност Веб сервиса.....	98
7.1. Напади на Веб корисничке агенте.....	98
7.1.1. Преотимање сесије.....	98
7.1.2. Приступ програмима и подацима на клијенту.....	99
7.2. Напади на Веб сервере.....	99
7.2.1. Фајл robots.txt.....	100
7.3. Напади на Веб апликације.....	101
7.3.1. Напади уметањем SQL наредби.....	102
7.3.2. Унакрсно скриптовање.....	104
7.3.3. Откривање верзије Веб апликација.....	105
7.3.4. Откривање рањивости у претходним верзијама.....	108
8. Напади са циљем ускраћивања услуге.....	110
8.1. Пинг поплаве.....	112
8.2. Смурф напад.....	113
8.3. Напади на нивоу апликације.....	114
8.4. Напади спорим пријемом/слањем.....	115
9. Шпијунирање рачунарских система.....	116
9.1. Снимање уноса са тастатуре.....	116
9.2. Надгледање активности миша.....	119
9.3. Шпијунски софтвер.....	120
9.4. Шпијунирање мобилних телефона.....	121
10. Анализа безбедности система и мрежа.....	122
10.1. Провера пробојности.....	123
10.2. Концепт ћупа са медом.....	124

10.3. Алати за откривање пропуста.....	125
11. Литература.....	126





# 1. Увод

Безбедност и заштита рачунарских телекомуникација данас представљају све значајније и све популарније питање. Пројекат Викиликс је показао на који начин и у којој мери објављивање поверљивих информација на Интернету може имати утицај на највише државне функције, па чак и на спољну политику земаља. Последње оптужбе на рачун америчке државне безбедносне агенције по питању прислушкивања коришћењем савремене технологије изнешене су јавно и из више извора. Кина на својој територији не дозвољава употребу Интернет сервиса америчке компаније Гугл, турски политичари су својим грађанима забранили коришћење Твитер сервиса, а неке државе као једну од критичних мера одбране разматрају и комплетно укидање приступа Интернету.

Пораст значаја заштите у рачунарским телекомуникацијама не дешава се неочекивано, с обзиром на то да смо сведоци све веће миграције вредности из стварног света у виртуални. На пример, пре тридесетак година је Интернет мрежа коришћена углавном за размену јавно доступних информација унутар академских кругова. Данас, насупрот томе, разлика у пар позиција код рангирања на Веб претраживачу компаније Гугл може бити основа за давање отказа хиљадама запослених.

Сваки пораст вредности уредно прати и пораст жеље да се та вредност преотме, односно пораст ризика да ће вредност бити изгубљена. Нападаци из дана у дан проналазе нове начине за остваривање напада, док рачунарски експерти као одговор на то представљају нова безбедносна решења.

Основна идеја овог уџбеника је да студентима пружи квалитетну основу за проучавање домена заштите у савременим рачунарским мрежама и оспособи их за суочавање са практичним изазовима из ове области. У уџбеник су уграђена општа знања из ове области, али и искуства аутора стечена дугогодишњим бављењем овом облашћу у пракси.

## 1.1. Криптолошке основе

Криптографија чини важну основу безбедности у савременим рачунарским телекомуникацијама. Применом криптографских метода остварују се принципи као што су поверљивост и веродостојност, док се посредно може повољно утицати и на повећање доступности коришћених ресурса. Своју примену у заштити рачунарских телекомуникација и система нашли су и симетрични и



асиметрични алгоритми, као и алгоритми и за двосмерно и за једносмерно шифровање (хеш функције).



Слика 1.1-1 - Улога криптографије за заштиту поверљивости

Са друге стране, криптоанализа представља једну од основа за извршавање напада на рачунарске системе и мреже. На исти начин на који се криптолошким методама штите рачунарске телекомуникације, оне се од стране нападача покушавају да разбију применом криптоаналитичких метода. Међутим, с обзиром да савремени шифарски алгоритми нуде висок степен заштите, разбијање шифрованих комуникација (код којих су коришћене најбоље препоруке!) практично је немогуће. Из тог разлога нападачи, у таквим ситуацијама, најчешће прибегавају проналажењу алтернативних начина за постизање својих циљева (коришћење грешака у имплементацији криптографских алгоритама, затим грешака у софтверским имплементацијама мрежних протокола и слично).

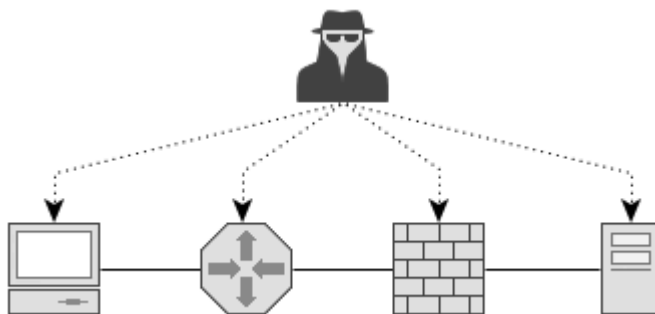
## 1.2. Димензије напада

Две основне димензије напада у рачунарским мрежама везане су за њихову **мету** (који део рачунарске мреже се напада) и **сврху** (шта се жели постићи нападом). При том, већина напада је сложена, односно, да би се постигао коначан циљ напада потребно је напасти више различитих делова система и угрозити различите апсекте њихове безбедности. У основне мете напада у рачунарским мрежама спадају:

- мрежни уређаји (крајњи и посредни)
- комуникациони канали
- крајњи корисници

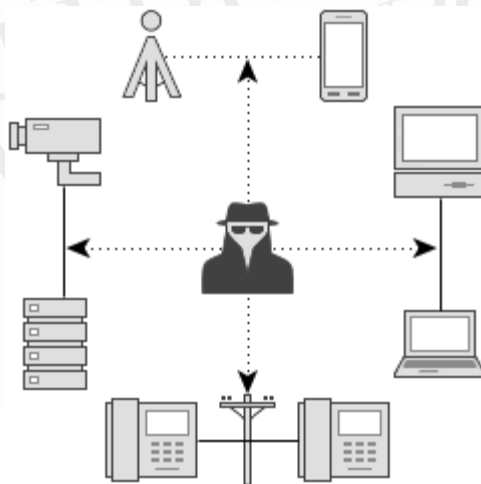
Најчешће вршени напади у савременим рачунарским мрежама свакако су напади на мрежне уређаје. Мета оваких напада могу бити крајњи уређаји

(сервери, клијентски рачунари и слично) или посредни (комутатори, рутери, филтери пакета, бежичне приступне тачке и слично). При том треба имати у виду да се могу нападати сви слојеви *OSI* референтног комуникационог модела, од физичког па до слоја апликације.



Слика 1.2-1 - Напади на мрежне уређаје

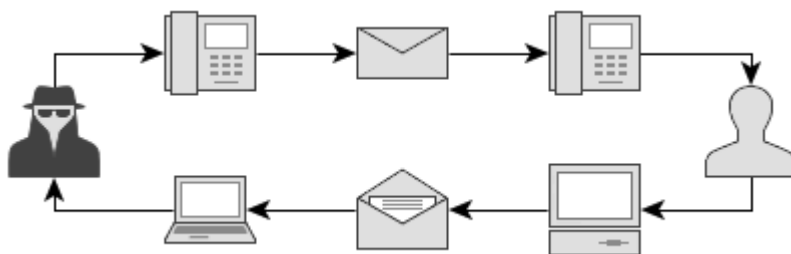
У неким ситуацијама нападач нема могућност или потребу да напада мрежне уређаје, већ напада комуникационе канале. Напади на комуникационе канале могу бити засновани на пасивној анализи њихових сигнала, или на њиховом активном мењању, било спољним утицајем, било уметањем посредујућег мрежног уређаја. Додатно, нападачи имају и могућност да комплетно онемогуће комуникацију пресецањем необезбеђених комуникационих канала.



Слика 1.2-2 - Напади на комуникационе канале

У пракси се често срећу ситуације у којима нападачи нису у стању да ефикасно изврше напад на мрежне уређаје, нити на комуникационе канале, већ

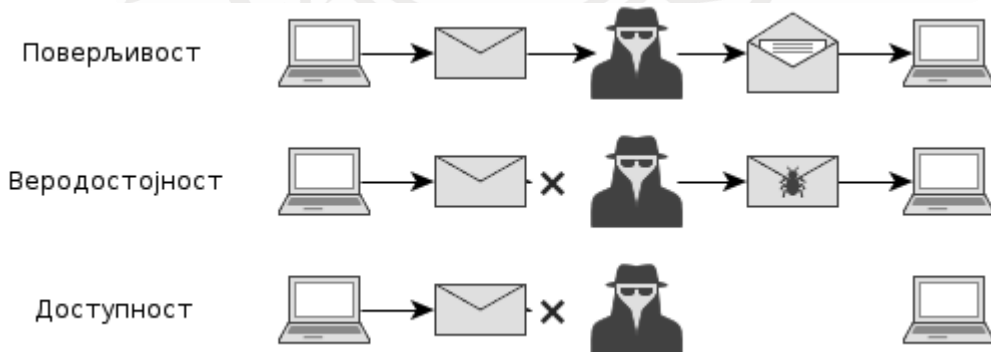
покушавају да свој циљ остваре деловањем на крајње кориснике. У зависности од тога који циљ желе да постигну, нападачи се лажно представљају, траже од корисника поверљиве податке, подмећу лажне податке, претражују смеће, затрпавају кориснике великим бројем података и томе слично. У својој књизи *Уметност обмане* Кевин Митник, један од најпопуларнијих рачунарских криминалаца 20. века, наводи како је већи број напада управо извео обмањивањем корисника и коришћењем њихових пропуста.



Слика 1.2-3 - Напади на крајње кориснике

Следећу димензију напада чини његова сврха, односно која се заштићена карактеристика напада. У основне категорије овде спадају поверљивост, веродостојност и доступност.

Сврха **напада на поверљивост** (енгл. *Confidentiality*) је да се неовлашћено приступи одређеним подацима, било да се они чувају у меморији неког рачунарског система, било да се преносе путем рачунарске мреже. У основи, овим нападом се подаци само преузимају, односно не мењају се, нити уклањају, тако да је успешно изведене нападе овог типа често веома тешко открити.



Слика 1.2-4 - Напади на поверљивост, веродостојност и доступност

**Напади на веродостојност** (енгл. *Integrity*) имају за циљ да лажирају садржаје комуникација. Да би веродостојност била остварена потребно је да прималац

података буде сигуран да подаци које је добио заиста потичу од наведеног пошиљаоца, као и да су изворни, односно да током транспорта нису мењани. У складу са тим, ови напади се извршавају тако што се нападач уметне у комуникацију као посредник и мења њене поруке, или се у потпуности представи као једна од страна у комуникацији. Треба имати у виду да се код неких типова напада (на пример, напад типа *човек у средини*) путем напада на веродостојност посредно постиже и напад на поверљивост.

**Напади на доступност** (енгл. *Availability*) имају за циљ да регуларним корисницима одређеног ресурса онемогуће нормалан рад са њим. На пример, у нападима са циљем ускраћивања услуге (енгл. *Denial-of-Service*) често се мрежни сервери затрпавају огромном количином бесмислених захтева са циљем да се исцрпу доступни ресурси (процесорско време, меморија, комуникациони канал) тако да сервер постане недоступан регуларним клијентима.

У складу са наведене две основне димензије напада може се развити матрица у коју се могу сврстати сви до сада познати напади у рачунарским мрежама:

	Канали	Уређаји	Корисници
Поверљивост	прислушкивање	преузимање контроле и података	обмањивање
Веродостојност	измена саобраћаја	измена података и програма	лажно представљање, крађа идентитета
Доступност	пресецање, преоптерећивање	заузимање ресурса, слом	узнемиравање, досађивање, затрпавање подацима

Приликом коришћења ове матрице треба имати у виду да су неки напади сложенији, односно да испуњавају више критеријума. Такође, код неких апстрактнијих типова напада (какви су, у последње време, веома популарни напади на приватност корисника) нема директних прекршаја у вези са поверљивошћу, веродостојношћу или доступношћу, већ се они базирају на неетичкој употреби података који су постали доступни нападачима услед технолошких пропуста или непажње корисника.

### 1.3. Фазе напада и одбране

Напади на рачунарске системе и мреже у основи имају пет фаза, од којих свака има своју улогу у остваривању циљева напада, као и своје карактеристичне активности. У ових пет фаза спадају:

1. извиђање,
2. планирање и симулација напада,
3. извршење напада,
4. остваривање циљева напада и
5. уклањање трагова.

Наравно, немају сви напади све фазе, а углавном је успешност у извршавању одређене фазе предуслов за прелазак на следећу фазу. На пример, уколико нападач пронађе озбиљан и добро познат пропуст у фази извиђања, фаза планирања и симулације напада може бити изостављена. Са друге стране, уколико током фазе извиђања не пронађе одговарајући пропуст у систему који се напада, нападач неће моћи да приступи наредним фазама.

#### **Пример**

*Узмимо за пример напад код кога нападач покушава да измени стање на свом банковном рачуну. Први корак нападача је извиђање, односно упознавање са организацијом рачунарске мреже и информационог система банке. Нападач, затим, на основу прикупљених информација планира напад и проверава вероватноћу његовог успеха на основу симулација. Уколико је задовољан резултатима симулације приступа извршавању напада, односно покушава да неовлашћено приступи бази података у којој жели да мења податке. Уколико у томе успе, нападач постиже циљ напада - мења стање на свом банковном рачуну. Да би спречио откривање ове измене, као и успех евентуалне истраге која би водила до њега, нападач затим уништава све трагове свог напада (уклања евентуално инсталиране алате, записе из лог фајлова и слично).*

Насупрот фазама напада стоје фазе одбране чији је задатак да онемогуће извршавање напада или да, макар, умање његове ефекте. У основне фазе одбране спадају:

1. Развој безбедносне полисе
2. Имплементација полисе, избор и инсталација система за заштиту
3. Евалуација ефикасности изабране заштитне конфигурације
4. Откривање активних или окончаних напада
5. Утврђивање и отклањање штете изазване нападом
6. Опоравак система
7. Унапређивање безбедносне полисе и система за заштиту

Дакле, у питању је једна континуална и циклична активност јер је за успешну одбрану неопходно ажурирање безбедносне полисе и инсталираних система за заштиту. Такође, треба имати у виду и да се често јавља неслагање између безбедносне полисе и стварног стања ствари, односно да се безбедносна полиса (често и квалитетно развијена) у пракси не поштује.

### 1.3.1. Напади извиђања

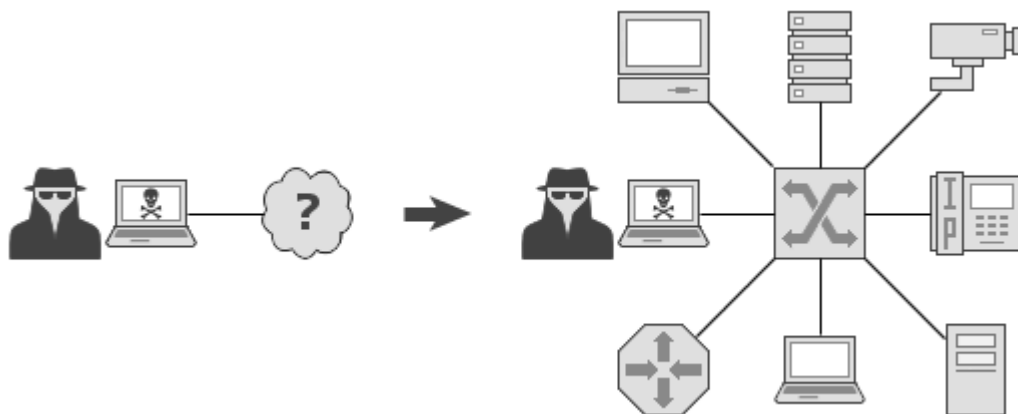
Пре самих акција за постизање коначног циља напада нападач мора да поседује довољно информација о циљном систему, као и о његовом окружењу. У ове информације спадају мрежна организација окружења, инсталирани софтвер, верзије и подешавања, период у коме систем није ресетован, активни сервиси и слично, односно све што може садржати рањивости које се могу искористити за потребе напада. За добијање ових информација користи се извиђање, односно тзв. напади извиђања.



Слика 1.3.1-1 - Српски војник извиђач (Драгутин Матић) у I светском рату

Једна од основних активности приликом напада извиђања је **скенирање рачунарских система и мрежа**. Извиђање се најчешће обавља на три нивоа: хоризонтално, вертикално и дубинско. **Хоризонтално извиђање** је процес у

коме нападач испитује који се мрежни уређаји и рачунари налазе у рачунарској мрежи и која је њена топологија. Овакво извиђање се може извести на различите начине али је његов најједноставнији облик коришћење *ICMP* протокола, односно тзв. „пинговање“ свих *IP* адреса из опсега које користи рачунарска мрежа на коју се врши напад.



Слика 1.3.1-2 - Улога хоризонталног извиђања

У наставку је дат пример коришћења *Nmap* алата за скенирање опсега приватних адреса у циљу проналажења активних чланова мреже:

```
bash-4.2# nmap -sP 192.168.1.*
Starting Nmap 5.51 ( http://nmap.org ) at 2012-08-28 11:11 CEST
Nmap scan report for 192.168.1.7
Host is up (0.00012s latency).
MAC Address: 00:0D:60:32:EA:3E (IBM)
Nmap scan report for 192.168.1.10
Host is up (0.00083s latency).
MAC Address: 00:15:5D:6E:6B:00 (Microsoft)
Nmap scan report for 192.168.1.12
Host is up (0.00059s latency).
MAC Address: 00:22:6B:EF:B1:2F (Cisco-Linksys)
Nmap scan report for 192.168.1.15
Host is up (0.00018s latency).
MAC Address: 00:16:E6:D9:05:76 (Giga-byte Technology Co.)
Nmap scan report for 192.168.1.20
```

```
Host is up (0.00027s latency).
MAC Address: 00:15:5D:6E:6B:04 (Microsoft)
Nmap scan report for 192.168.1.34
Host is up (0.00081s latency).
Nmap scan report for 192.168.1.122
Host is up (0.00017s latency).
MAC Address: 00:1A:4D:93:54:8B (Giga-byte Technology Co.)
Nmap done: 256 IP addresses (14 hosts up) scanned in 2.56 seconds
bash-4.2#
```

*Листинг 1 - Пример хоризонталног извиђања коришћењем Nmap алата*

**Вертикално извиђање** има за циљ да нападачу пружи информацију о томе који су портови на појединачним мрежним уређајима (најчешће рачунарима) отворени, односно који се мрежни сервиси на њима извршавају.

```
bash-4.2$ nmap 192.168.1.122 -p 1-65535
Starting Nmap 5.51 ( http://nmap.org ) at 2012-08-28 10:59 CEST
Nmap scan report for 192.168.1.122
Host is up (0.00024s latency).
Not shown: 65533 closed ports
PORT STATE SERVICE
22/tcp open  ssh
80/tcp open  http
Nmap done: 1 IP address (1 host up) scanned in 336.15 seconds
bash-4.2$
```

*Листинг 2 - Пример вертикалног извиђања - проналажења отворених портова*

**Дубинско извиђање** има за циљ да нападачу пружи што детаљније информације о софтверу који се извршава на одређеном мрежном уређају. У ове информације спадају тип, верзија и подешавање софтвера, а односе се и на системски и на кориснички софтвер. Ове информације имају највиши значај јер на основу њих нападач открива потенцијалне безбедносне рупе које може искористити за постизање циљева напада.

```
bash-4.2# nmap -O 192.168.1.122
```



```
Starting Nmap 5.51 ( http://nmap.org ) at 2012-08-28 11:10 CEST
Nmap scan report for 192.168.1.122
Host is up (0.00022s latency).
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.19 - 2.6.36
Network Distance: 1 hop
OS detection performed.
Nmap done: 1 IP address (1 host up) scanned in 12.07 seconds
bash-4.2#
```

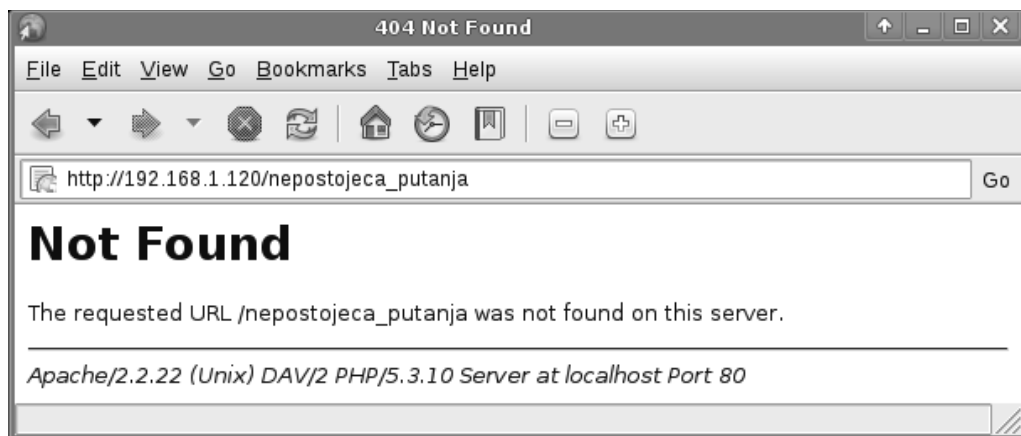
### *Листинг 3 - Откривање оперативног система удаљеног рачунара*

На листингу 3 је дат пример дубинског извиђања којим се утврђује тип и верзија инсталираног оперативног система, док је на листингу 4 дат пример коришћења телнет алата за утврђивање верзије софтвера који омогућава сервис на порту 22 (*Secure Shell*, алат за удаљену администрацију *UNIX/Linux* система).

```
bash-4.2# telnet 192.168.1.122 22
Trying 192.168.1.122...
Connected to 192.168.1.122.
Escape character is '^]'.
SSH-2.0-OpenSSH_5.9
^]
telnet> quit
Connection closed.
bash-4.2#
```

### *Листинг 4 - Откривање типа и верзије софтвера који стоји иза мрежног сервиса*

На слици 1 приказано је испитивање верзије и подешавање Веб сервера путем намерног захтевања ресурса који не постоји на серверу. На основу одговора сервера може се закључити да Веб сервис на њему омогућава *Apache* Веб сервер (верзија 2.2.22) који се извршава на *UNIX/Linux* оперативном систему и на коме је активиран *PHP* модул верзије 5.3.10. Ово су драгоцене информације за нападача јер на основу њих он може претражити базе познатих рањивости и пронаћи нападе који се могу применити.



*Слика 1.3.1-3 - Испитивање верзије и подешавања Веб сервера*

Напади извиђања не морају бити усмерени на само један циљни систем већ се могу користити за откривање више система који садрже одговарајуће рањивости. На пример, нападачи често користе аутоматизоване програме типа агената који проналазе Веб сајтове на Интернету засноване на софтверским решењима са јавно доступним изворним кодом (*Joomla*, *WordPress* и слично) и у локалну базу података смештају податке о њима (најчешће адресе сајтова и информације о инсталираној верзији софтвера). Онога тренутка када се за коришћену верзију софтвера пронађе нова рањивост нападачи покрећу аутоматизоване скрипте које приступају сајтовима забележеним у бази и мењају садржај, преузимају податке и слично.

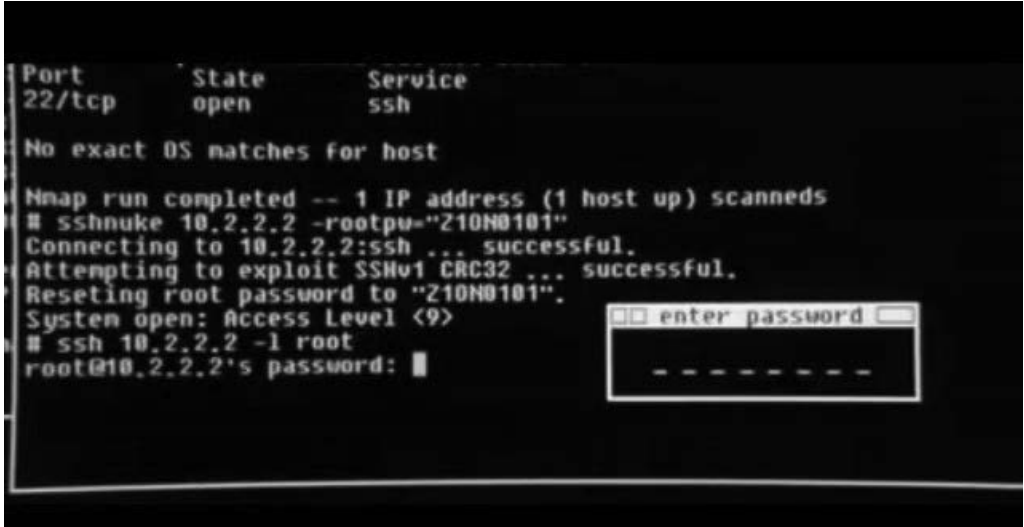
Иако напади извиђања сами по себи не причињавају значајну штету (осим мањег трошења ресурса система и мреже који се нападају) они су најава напада којим ће таква штета бити начињена. Из тог разлога је пожељно предузети два превентивна корака:

1. инсталирати безбедносне алате који препознају нападе извиђања и о томе извештавају администратора, и
2. подесити сопствене системе тако да нападач путем напада извиђања добије што оскуднију информацију.

На пример, коришћење *ICMP* протокола је веома ретко потреба регуларних корисника рачуарског система и мреже па се тај протокол може потпуно искључити или се о његовој употреби може обавештавати администратор. Или, као што је у претходном примеру приказано, од јавног одавања информација о верзији и карактеристикама Веб сервера најчешће ће само злонамерни корисници имати корист па га је пожељно потпуно искључити.

### 1.3.2. Анализа напада у филму „Матрица“

Напади на рачунарске системе и мреже често су коришћени мотиви у кинематографији. Међутим, они су често потпуно нереално приказани са циљем постизања спектакуларности и могућности „разумевања“ од стране технички необразоване публике. Један од ретких филмова у којима се процес напада на удаљени рачунарски систем приказује приближно реалном је „Матрица“ (енгл. *The Matrix*), снимљен 1999. године.



Слика 1.3.2-1 - Изглед екрана приликом напада на удаљени рачунарски систем

У једној сцени поменутог филма Тринити (један од главних јунака, са симболичним именом) покушава да преузме контролу над рачунарским системом који управља снабдевањем електричном енергијом. Напад садржи три основне фазе:

1. извиђање,
2. напад и
3. преузимање контроле над нападнутим системом.

У првој фази Тринити изводи вертикално извиђање коришћењем *Nmap* алата на основу кога утврђује да је порт 22 отворен (адреса која се скенира је у приватној мрежи што значи да се напад обавља изнутра или коришћењем компромитованог посредника из приватне мреже):

```
# nmap -O 10.2.2.2
```

Port	State	Service
22/tcp	open	ssh
No exact OS matches for host		
nmap run completed		

Након тога, Тринити извршава експлоит под називом *SSH Nuke* којим искоришћава рањивост у првој верзији *SSH* протокола и мења лозинку администратора у вредност „Z10N0101“:

```
# sshnuke 10.2.2.2 -rootpw="Z10N0101"
connecting to 10.2.2.2:ssh ... successful.
attempting to exploit SSHv1 CRC32 ... successful.
reseting root password to "Z10N0101".
system open: Access Level <9>
```

Након измене лозинке администратора Тринити се регуларно повезује на удаљени систем, користећи администраторски налог за добијање највиших привилегија на систему, и на њему извршава жељену наредбу:

```
# ssh 10.2.2.2 -l root
root@10.2.2.2's password:

RTF-CONTROL> disable grid nodes 21 - 48
Warning: Disabling nodes 21-48 will disconnect sector 11 (27 nodes)
```

Два корака која недостају у нападу извиђања јесу хоризонтално и дубинско извиђање. Што се хоризонталног извиђања тиче, оно је могло бити извршено и пре него што је приказан екран нападача. Са друге стране, између вертикалног извиђања и самог напада недостају два корака:

1. дубинско извиђање и
2. проналажење рањивости за откривене верзије софтвера.

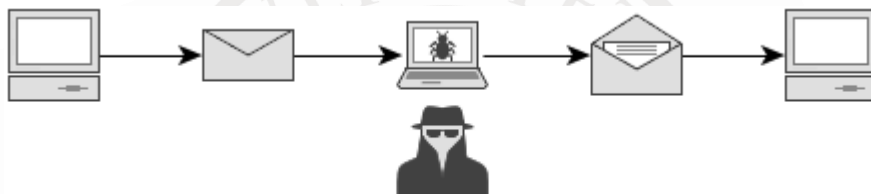
У приказаној поставци Тринити на свом рачунару већ има припремљен експлоит под називом *SSH Nuke* и подразумева да се на серверу користи *SSH* серверски софтвер који је рањив на тај тип напада. У реалном случају би до ових информација нападач дошао кроз поменуте изостављене кораке.

## 1.4. Прислушкивање и снимање саобраћаја

Напади на поверљивост у рачунарским мрежама најчешће се остварују „прислушкивањем“, односно снимањем мрежног саобраћаја. Снимање мрежног саобраћаја се углавном врши на три начина:

1. Нападач се уметне као посредник у комуникацији тако да саобраћај пролази кроз његов рачунар.
2. Нападач компромитује један од рачунара који комуницирају или неки од посредних мрежних уређаја тако да му шаљу копију саобраћаја.
3. Нападач користи зрачење проузроковано слањем података преко физичких медија и на основу њега реконструише садржај комуникације.

С обзиром на све већу сложеност рачунарских система и мрежа, прва два начина постају све једноставнија за извођење.



Слика 1.4-1 - Снимање саобраћаја путем посредовања

Да би нападач могао да прочита садржај снимљеног саобраћаја потребно је да тај саобраћај не буде шифрован или да је шифрован на такав начин да га је могуће разбити. У суштини, готово све шифарске алгоритме који се данас користе у рачунарским телекомуникацијама је могуће разбити, али је за то потребно нерационално много времена и рачунарске снаге.



Слика 1.4-2- Пример снимања саобраћаја путем Wireshark алата

Највећи безбедносни проблем код прислушкивања чини коришћење застарелих шифарских алгоритама и безбедносних система, а пре свега њихово не коришћење уопште. На пример, *HTTP* протокол, један од најпопуларнијих мрежних протокола данас, подразумевано не користи никакву заштиту. На слици 2 је приказан изглед екрана програма *Wireshark*, односно његова употреба за снимање и преглед података послатих путем овог протокола.

## 1.5. Типови нападача

За потребе ефикасније одбране од нападача потребно је раздвојити различите профиле и разумети специфичности сваког од њих. У пракси се често за све типове нападача користи израз **хакер** (енгл. *hacker*). Међутим, таква употреба термина је непрецизна јер он означава особу која поседује врхунско познавање информационих технологија, али чије понашање подразумевано није злонамерно. Насупрот хакерима стоје **крекери** (енгл. *cracker*) који такође поседују висока знања али их користе за остваривање сопствене користи, најчешће материјалне, кроз наношење штете другима.

Следећу категорију нападача чине тзв. **лејмери** (енгл. *lamer*) који такође путем рачунара и рачунарских мрежа остварују своју корист и наносе штету другима, али не поседују висока знања. Овакви нападачи нису у стању да сами пронађу пропусте у рачунарским системима већ користе резултате озбиљнијих појединаца и група који се баве рачунарском безбедношћу. Њихов рад се углавном заснива на праћењу откривених пропуста и проналажењу система код којих се ти пропусти могу искористити. Треба имати у виду и да је удео оваквог типа нападача највећи, а да њихове жртве могу представљати само системи код којих се безбедносне закрпе не примењују редовно.

Ипак, треба имати у виду да се већина озбиљних напада на рачунарске системе обави у потпуности од стране **инсајдера** или у сарадњи са њима. Инсајдери су особе које имају регуларан приступ систему (на пример, запослени у организацији) а који имају зле намере. Ове особе своје привилегије користе тако да самостално обаве неку злонамерну активност или да спољном нападачу са којим сарађују обезбеде улаз у систем. Из тог разлога је код пројектовања безбедносних система потребно да се у виду има и овај проблем, односно да се систем не штити само од напада споља, већ и од потенцијалних инсајдера.

## 2. Контрола приступа у рачунарским мрежама

Један од основних задатака код заштите рачунарских система и мрежа је контролисање ко има право да им приступа и које акције има право да извршава над одређеним објектима у њима. У суштини, потпуно безбедним системом може да се сматра систем код кога је развијен и одржава се савршен скуп правила за приступ, а која се увек поштују, односно није их могуће заобићи. Међутим, у стварности потпуно безбедних система нема, управо због тога што или није могуће дефинисати идеалан скуп правила, или није могуће обезбедити њихово безусловно поштовање у свим ситуацијама. Са друге стране, постоји више приступа и решења која, уколико су уграђена у рачунарски систем или мрежу која се штити, пружају прилично висок ниво заштите, односно омогућавају висок ниво контроле приступа.

### 2.1. Аутентификација и ауторизација корисника

Контрола приступа ресурсима у рачунарским мрежама може се посматрати као питање „**ко** сме, **шта**, и **са чим** да ради“. Дакле, два критична задатка у вези са контролом приступа су провера идентитета приступаоца и утврђивање да ли он има права да изведе захтевану акцију над одређеним објектом.



Слика 2.1-1 - Знање, поседовање, биће

Провера идентитета је акција којом се потврђује или оспорава истоветност изјављеног и стварног идентитета приступаоца. Под приступаоцем се могу подразумевати особе али и процеси који се извршавају на локалном или удаљеном рачунарском систему, односно њихови власници.

За утврђивање идентитета користе се три основне категорије: (тајно) **знање**,

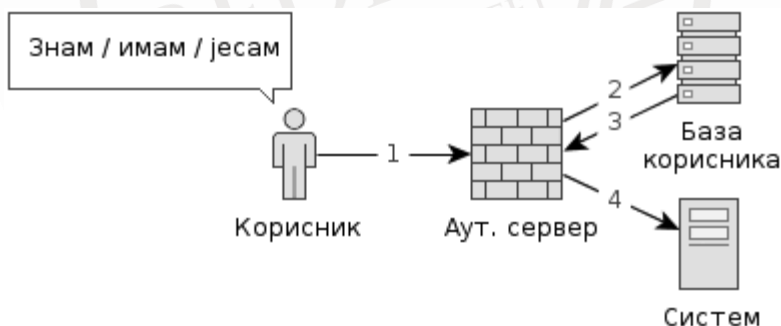


(искључиво) **поседовање** и (јединствене карактеристике за) **постојање**. Другим речима, одређена особа може потврдити свој идентитет нечим што **зна**, нечим што **има** или нечим што **јесте**. На пример, корисник вишекорисничког рачунарског система свом налогу може приступити уносом лозинке, убацивањем смарт-картице или скенирањем отиска прста.

Следећи задатак, након утврђивања идентитета приступаоца, јесте ауторизација, односно утврђивање његових права за извођење захтеваних акција над одређеним објектима. На пример, функција уклањања фајлова са фајл-система је опште доступна корисницима, али само администратор има право да уклања системске фајлове, док је рад обичних корисника ограничен на фајлове у њиховом личном директоријуму.

### 2.1.1. Основни модел аутентификације

У основном моделу за аутентификацију корисника у заштићеном репозиторијуму чувају се везе између идентитета корисника и онога што он зна/има/јесте, односно онога чиме ће се корисник аутентификовати. Активну компоненту за аутентификацију чини подсистем за аутентификацију који поседује интерфејс за задавање идентитета (на пример, унос корисничког имена) и податка којим се врши аутентификовање (на пример, унос лозинке, читач токена, биометријски скенер и слично).



Слика 2.1.1-1- Основни модел аутентификације

На слици 2. приказан је модел за основну аутентификацију на основу коришћења корисничког имена и лозинке. Честа пракса код овакве аутентификације је да се лозинка не чува у изворном облику, већ да се користи њена хеш вредност. На тај начин се нападачу, у случају компромитовања базе за аутентификацију, значајно отежава реконструкција лозинке јер се до ње у том случају долази разбијањем путем коришћења речника често коришћених фраза



или путем провере свих могућих комбинација.

Корисник
Корисничко име
Лозинка
Име
Презиме
...

Слика 2.1.1-2 - Модел базе података за основну аутентификацију

Подсистем за аутентификацију може имати могућност да блокира даље напредовање корисника уколико није успешно утврђен идентитет, а може га и пустити да анонимно извршава даље акције уколико је такво понашање прихватљиво у систему. Пример за овакво понашање су Интернет форуми код којих аутентификовани корисници имају право да постављају нове поруке и одговоре, док су анонимни корисници ограничени на прегледавање постојећег садржаја.

### 2.1.2. Вишефакторска аутентификација

#### ***5 million 'compromised' Google accounts leaked***

*A database of what appears to be some 5 million login and password pairs for Google accounts has been leaked to a Russian cyber security internet forum. It follows similar leaks of account data for popular Russian web services.*

10.9.2014. [www.rt.com](http://www.rt.com)<sup>1</sup>

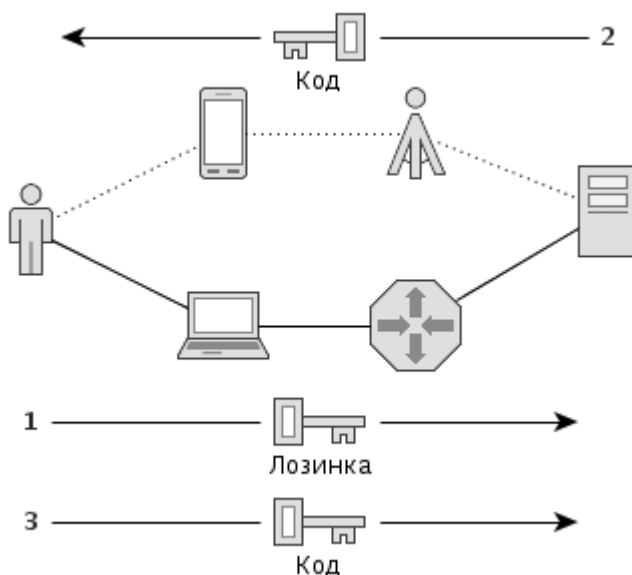
*(Провером наведених тврдњи аутори овог уџбеника су дошли до закључка да у питању није компромитовање налога за Гмаил сервис, већ других сервиса на којима су коришћени Гмаил налози.)*

Приликом аутентификације корисника могуће је користити и више података различитих категорија у циљу потврђивања идентитета. На пример, од корисника се може тражити давање токена или биометријских података, уз додатну верификацију уносом тајне лозинке. Овакав начин аутентификације се назива **вишефакторском аутентификацијом**.

<sup>1</sup> <http://rt.com/news/186580-millions-google-accounts-leaked/>

Предност вишефакторске аутентификације је обично у далеко вишем нивоу безбедности, односно већој поузданости. Наиме, вероватноћа истовременог компромитовања два аутентификациона објекта различитих категорија далеко је мања него када је у питању један објекат, односно једна категорија. На пример, нападач може открити лозинку корисника уколико оствари одговарајући приступ његовом рачунару (што у пракси обично није превише тешко). Међутим, уколико корисник приликом пријављивања на систем, осим лозинке, мора да користи и аутентификациону смарт-картицу, вероватноћа да ће нападач успети да и њу да украде далеко је мања.

Треба имати у виду да вишефакторска аутентификација не подразумева коришћење вишеструких лозинки, токена и биометрије, односно вишеструких аутентификационих података исте категорије, већ податке различитих категорија. При том се одређени акценат налази и на коришћењу вишеструких комуникационих канала за достављање идентификационих података различитих категорија.



Слика 2.1.2-1 - Модел двофакторске аутентификације коју користи Гугл

Пример вишеструке аутентификације је систем који је компанија Гугл међу првима понудила корисницима својих услуга. Укључивањем ове заштите корисници су дужни да приликом пријављивања на систем, осим лозинке, унесу и специјални шестоцифрени број. Овај број је случајно генерисан на страни сервера а кориснику се доставља путем одвојене, телефонске линије, у облику

СМС поруке или изговарањем од стране аутомата. На овај начин се за приступ сервисима које нуди компанија Гугл од корисника захтева да, поред познавања лозинке, он има и приступ свом мобилном телефону. Наведени систем је посебно вредан за кориснике који свом налогу често приступају са различитих рачунара - на пример, из Интернет кафеа приликом путовања.

### 2.1.3. Системи за разликовање робота и људи

Када је у питању моћ обраде података једноставним алгоритмима, ту рачунари показују неупоредиво боље резултате од људског интелекта. Из тог разлога се рачунари и користе да би одменили људе у обављању одређених задатака које је могуће аутоматизовати.

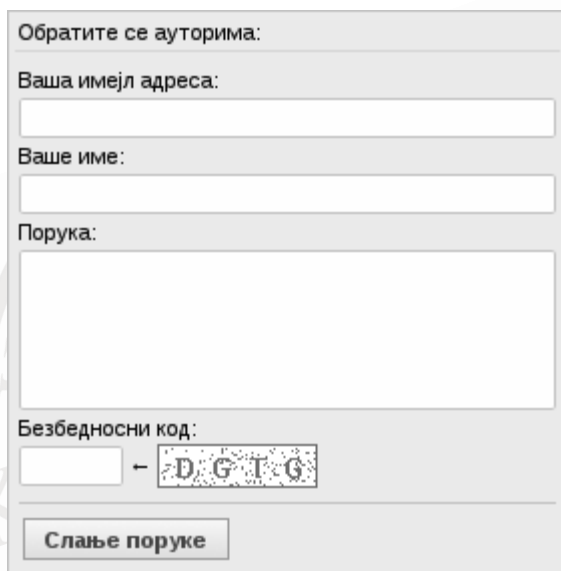


*Слика 2.1.3-1 - Сцена из филма „Вештачка интелигенција“*

Међутим, на описани начин се рачунари могу и злоупотребљавати путем маргинализације људских резултата. На пример, човеку је потребно неколико десетина секунди до неколико минута да смисли и откуца коментар за неку вест на информативном порталу, док рачунари могу да генеришу на хиљаде бесмислених порука те дужине у само једној секунди. Таквом употребом је могуће затрпа(ва)ти одређени портал рачунарски генерисаним коментарима тако да смислени коментари стварних аутора престану да буду видљиви.

За одбрану од овог типа напада потребно је коришћење одређених функција

(слање коментара из претходног примера) дозволити само људима, а забранити рачунарима (роботима). Први корак у томе чини препознавање да ли акцију покушава да изведе човек или рачунар, а основу за то чини тест под називом „потпуно аутоматизован јавни Турингов тест за разликовање људи од рачунара“ (енгл. *Completely Automated Public Turing test to tell Computers and Humans Apart*, *CAPTCHA*). Суштина овог теста огледа се у томе да се од друге стране у комуникацији захтева да реши одређени проблем, за који је људима потребно прихватљиво мало времена, док је рачунарима потребно нерационално много времена, или то уопште нису у могућности.



Обратите се ауторима:

Ваша имејл адреса:

Ваше име:

Порука:

Безбедносни код:

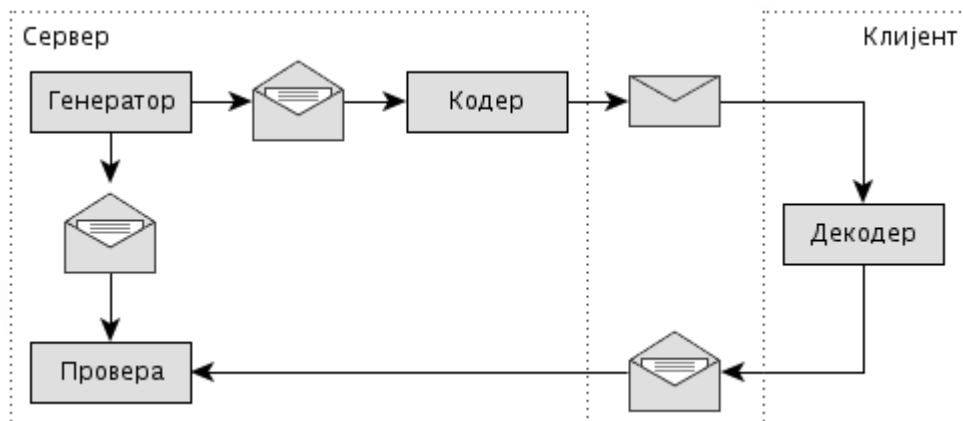
Слање поруке

Слика 2.1.3-2 - Коришћење CAPTCHA система на сајту *racunarskemreze.com*

Тестови овог типа се данас углавном реализују кроз препознавање одређених симбола чији је изглед у тој мери деформисан да су алгоритми за оптичко препознавање карактера (енгл. *Optical Character Recognition*, *OCR*) неефикасни, док је за људе препознавање ипак могуће (пример на слици 2). Међутим, треба имати у виду да CAPTCHA тестови нису ограничени само на овај тип садржаја већ се могу користити било који - аудио, логичка питања, опште знање и друго.

Могуће је направити паралелу између CAPTCHA система и шифарских система. Код шифарских система поруком располаже само онај ко је успео да је прими, односно дешифрује. За остале, оне који не располажу потребним инструментима за пријем - шифарским алгоритмом и кључевима - порука представља неразумљив скуп података. На исти начин функционишу и класични CAPTCHA системи - сервер генерише одређену поруку и кодује је у одговарајући

облик (који је прилагођен људској обради, а код кога је отежана рачунарска обрада). Након пријема кодоване поруке, на страни клијента се врши њено декодовање. Људи поседују већ развијене функције за то (уклањање снега, исправљање симбола и слично) и могу примљену поруку да декодују у веома кратком временском периоду (за разлику од рачунара код којих је потребно развити одговарајуће алгоритме, а чија је ефикасност и успешност углавном испод нивоа прихватљивог). Коначно, клијент серверу враћа кодовану поруку у изворном облику и тиме потврђује своју могућност њеног декодовања.

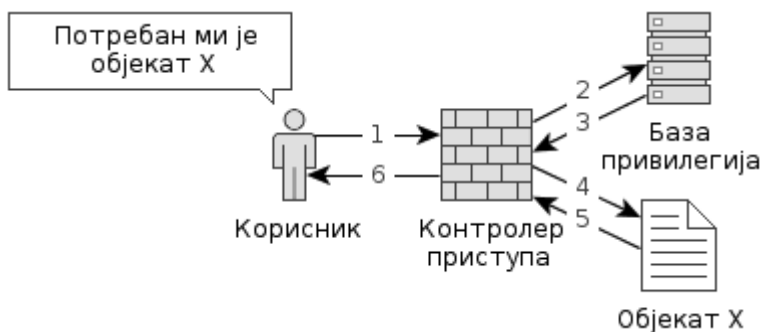


Слика 2.1.3-3 - Модел рада CAPTCHA система је близак шифарском моделу

Треба имати у виду да се у општем случају, код поређења брзине рада рачунара и човека увек користи човеков интелектуални центар, који је најспорији. Моторички и емотивни центри човека раде неупоредиво брже од интелектуалног, и то са много већом количином података. Из тог разлога је човек у могућности да обавља изузетно сложене процесе (нпр. вожња аутомобила са ручним мењачем уз разговор мобилним телефоном и контролом свих несвесних функција организма) док резултати покушаја да се створе роботи који ће се макар само кретати попут људи још увек делују прилично комично.

## 2.1.4. Основни модел ауторизације

Под ауторизацијом се подразумева утврђивање и поштовање права и ограничења које корисник има у раду са одређеним објектима рачунарског система или мреже. Она може бити дефинисана на нивоу рачунарске мреже, радне групе, појединачног система, софтверског пакета и томе слично.

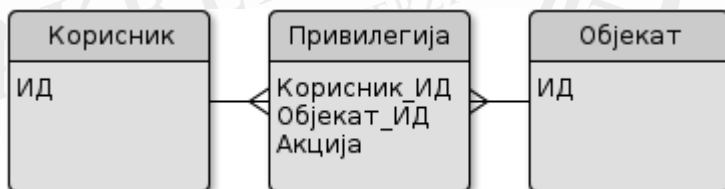


Слика 2.1.4-1 - Модел контроле приступа објектима (ауторизација)

На слици 1. приказан је основни модел ауторизације, односно контроле приступа (извођења акција над) објектима. У оквиру модела су приказани следећи кораци:

1. Корисник покушава да изведе акцију над одређеним објектом.
2. Контролер проверава у бази привилегија да ли је корисник овлашћен да изврши тражену акцију над задатим објектом.
3. У случају да је провера из претходног корака успешна, кориснику се одобрава приступ или извршавање акције над објектом.

Да би се овакав модел подржао предложена структура базе података са привилегијама је следећа:



Слика 2.1.4-2 - Релациони модел података за ауторизацију

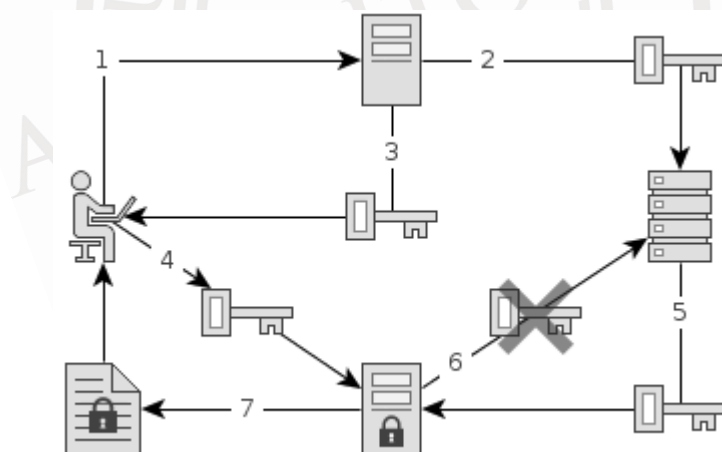
Дакле, у бази података се између ентитета *Корисник* и *Објекат* креира веза под називом *Привилегија* у оквиру које се дефинише акција која је кориснику дозвољена над одређеним објектом. Основни недостатак овог модела је потреба да се база привилегија допуњује код сваког додавања нових објеката, корисника или акција.

### 2.1.5. Контрола приступа заснована на тикетима

Систем тикета омогућава ауторизацију без аутентификације, односно омогућава контролу приступа без утврђивања идентитета. Доказ да корисник има право да изврши захтевану акцију је само поседовање тикета. Са друге стране, тикет је креиран и кориснику додељен од стране администратора ресурса коме се приступа, најчешће заштићеним каналом.

Сам идентитет корисника може, директно или индиректно, бити уграђен у тикет, али то није неопходно. При том, недостатак аутентификације омогућава и трансферабилност приступа, односно омогућава приступ другом кориснику којем је тикет прослеђен. Ово је веома корисно у ситуацијама када се жели избећи отварање новог налога кориснику, као и откривање приступних параметара неког од већ постојећих налога. Са друге стране, трансферабилност тикета се може укинути његовим везивањем за идентитет корисника (спрега са аутентификацијом) или за тренутну сесију.

Посебна карактеристика тикета је да, осим што ограничавају приступ само на кориснике који их поседују, они ограничавају и сам број приступа. Наиме, тикети су најчешће намењени за једнократну употребу и није их могуће користити више пута.



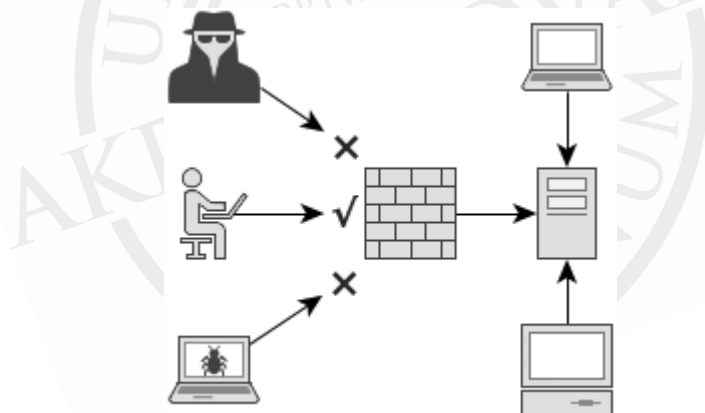
Слика 2.1.5-1 - Принцип рада система који користи тикете

Чест пример коришћења тикета јесу Веб апликације у којима се кориснику одобрава једнократан приступ заштићеном ресурсу (на пример, преузимање фајла, попуњавање анкете и слично). У овим сценаријима корисник приступа једној страници која генерише тикет са случајном вредношћу (коју је тешко

погодити) и складишти га у позадинској бази података. Након тога се кориснику прослеђује тикет и адреса за преузимање заштићеног ресурса (аутоматским преусмеравањем, слањем електронске поште и слично) на којој се проверава да ли достављени тикет постоји у бази као неискоришћен. Уколико такав тикет постоји он се поништава а кориснику се шаље заштићен ресурс. Овакав приступ је добар јер кориснику онемогућава да више пута преузме ресурс (на пример, освежавањем странице) или да другој особи омогући преузимање ресурса путем копирања адресе са које је заштићени ресурс преузет.

## 2.2. Филтери пакета

Један од најзначајнијих концепата у заштити рачунарских мрежа чине **филтери пакета**. Филтери пакета, понекад називани и ватреним зидовима (енгл. *firewall*), заштитним зидовима или мрежним баријерама, омогућавају да се задавањем одређених критеријума неке мрежне комуникације дозволе, а неке забране. С обзиром да су рачунарске телекомуникације засноване на пакетском преносу података, принцип рада филтера пакета заснива се на пропуштању или одбијању одређеног пакета, у зависности од дефинисаних правила.



Слика 2.2-1 - Улога филтера пакета

Да би се успешно користили филтери пакета потребно је испунити више услова, од којих су најзначајнији:

1. потребно је изабрати одговарајући филтер пакета - са одговарајућим функцијама и капацитетом;
2. само постојање филтера пакета не подиже ниво заштите док год он није адекватно подешен;



3. сама мрежа у којој се филтер пакета користи мора бити организована на одговарајући начин да би заштита била могућа;

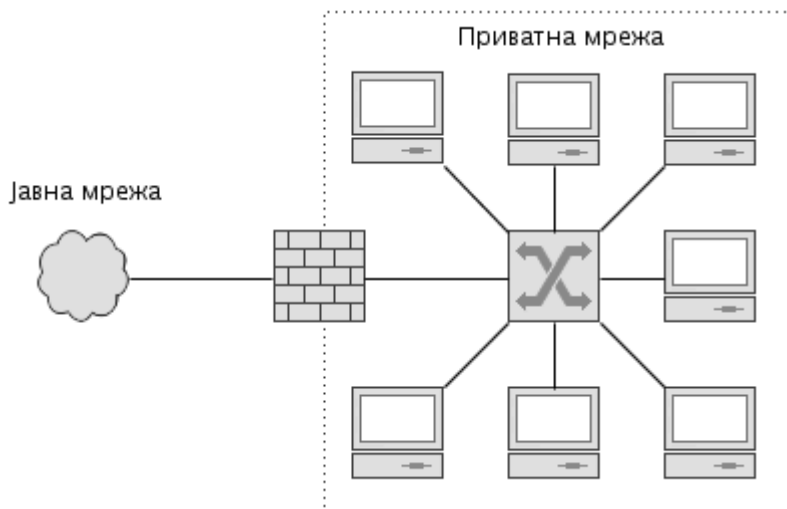
Дакле, уколико посматрамо једну приватну рачунарску мрежу у којој се налази сервер, коме је потребно да приступају и наши запослени који се налазе ван приватне мреже, улога филтера пакета би била да ограничи комуникације са спољном мрежом само на оне које потичу од наших запослених. Другим речима, задатак филтера пакета би био да спречи успостављање комуникација са сервером од стране непознатих и, највероватније, злонамерних чланова спољне мреже.

Да би се наведени задатак успешно извршио морају бити испуњена три претходно наведена услова. Као прво, морамо искористити одговарајући филтер пакета, у којем имамо могућност да дефинишемо ко су чланови спољне мреже у које имамо поверење и које привилегије њима дајемо. Додатно, сам капацитет филтера мора бити такав да он не представља уско грло у комуникацији - на пример, уколико са спољним корисницима водимо видео конференцију, филтер пакета мора радити довољно брзо да не проузрокује пад квалитета слике и говора, као ни приметно кашњење.

Следећи услов је исправно подешавање филтера пакета. Филтери пакета стижу са подразумеваним општим подешавањима, која су углавном или прихватање, или одбијање свих пакета. Филтер пакета који пропушта све пакете не нуди никакву заштиту, док филтер пакета који одбија све пакете у потпуности онемогућује комуникацију. Дакле, након избора одговарајућег филтера пакета потребно га је прецизно подесити тако да дозвољава само комуникацију коју смо ми одобрили.

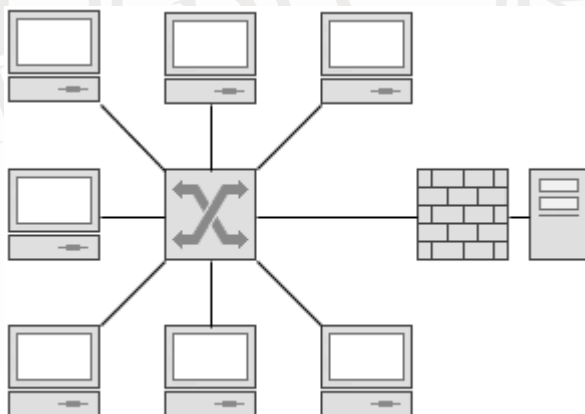
Конечно, потребно је испунити и трећи услов, а то је да је примена филтера пакета могућа, односно да га није могуће заобићи или преварити. На пример, уколико нападач има могућност алтернативне путање до сервера, кроз приватну мрежу, филтер пакета ће бити заобиђен и неће пружити никакву заштиту. Или, уколико је у подешавањима филтера адреса клијената у спољној мрежи статично задата, а нападач има могућност отимања те адресе, његови пакети ће проћи кроз филтер.

Филтерима пакета могу се штитити целе мреже (слика 2) или појединачни чланови (слика 3). Такође, број коришћених филтера пакета у мрежи није ограничен, тако да је идеалан модел употребе филтера пакета (приказан на слици 3) онај у коме се филтери пакета постављају и између различитих сегмената мреже, и испред појединачних чланова (слика 4).



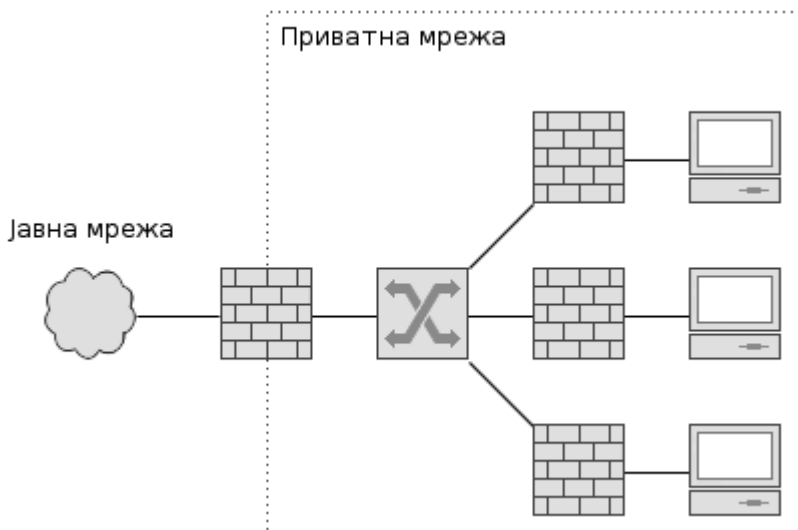
Слика 2.2-2 - Заштита приватне мреже једним филтером пакета

За заштиту појединачних чланова мреже, уколико су у питању радне станице или мањи сервери, најчешће се користе софтверски филтери пакета. Са друге стране, за заштиту сервера под великим оптерећењем, као и за заштиту сегмената мреже, углавном се користе засебни уређаји са сопственим хардвером.



Слика 2.2-3 - Заштита појединачних чланова мреже филтером пакета

Основни недостатак коришћења филтера пакета као на моделу на слици 4, односно постављања филтера пакета испред сваког члана и сегмента мреже, чини потреба да се брине о њиховој конфигурацији. Са друге стране, нека решења овог типа нуде централизовану администрацију.



Слика 2.2-4 - Идеалан модел употребе филтера пакета

Треба имати у виду да се филтерима пакета може ограничити не само то који пакети имају право да путују ка систему који се штити, већ и то које пакете тај систем има право да шаље. Овакав начин употребе филтера пакета је посебно значајан за спречавање напада са система у чије се кориснике нема довољно поверења, као и са система који су заражени малициозним софтвером.

Још једна важна карактеристика филтера пакета је то да се при пројектовању мреже и њиховом подешавању они морају користити тако да не постоје периоди у којима мрежа није заштићена. На пример, уколико дође до нестанка електричне енергије, након њеног повратка не сме се десити да прво прораде мрежне комуникације а након тога филтрирање пакета. У принципу, филтери пакета би требало да представљају једине везе између делова мреже између којих се налазе, а да подразумевано блокирају сав саобраћај док не уђу у регуларан режим рада.

### 2.2.1. Хардверски и софтверски филтери пакета

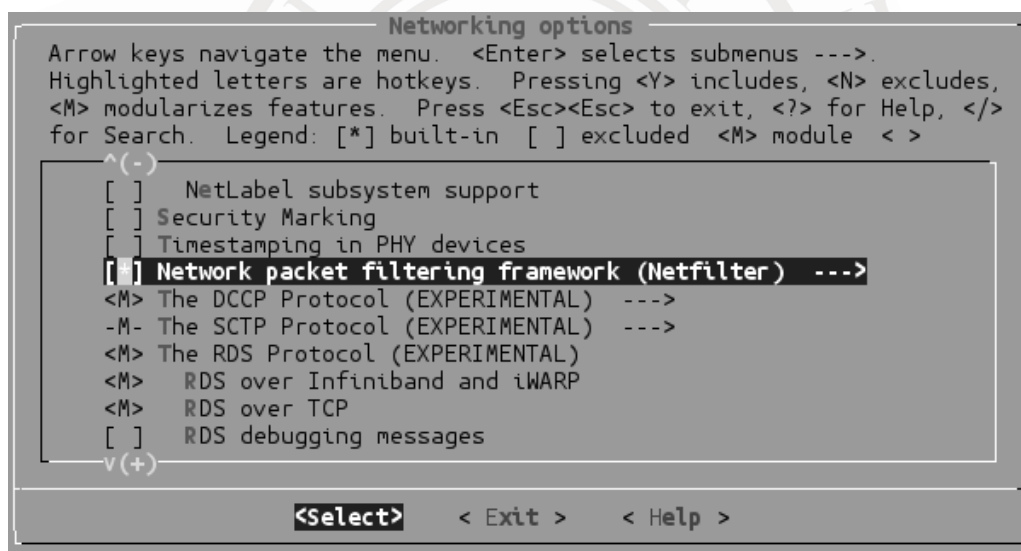
Постоје хардверски и софтверски филтери пакета. Хардверски филтери пакета представљају заокружена решења у виду засебних мрежних уређаја на које је инсталиран наменски софтвер за филтрирање пакета. На слици 1. је приказан пример хардверског филтера пакета, уређај са ознаком ASA 5505 који производи компанија Циско. Уређај је намењен за примену код мањих, канцеларијских пословних система, може да обрађује до 150Mb саобраћаја по секунди, до

10.000 паралелних сесија и до 4.000 паралелних веза, а помоћу њега је могуће и креирати виртуалне приватне мреже.



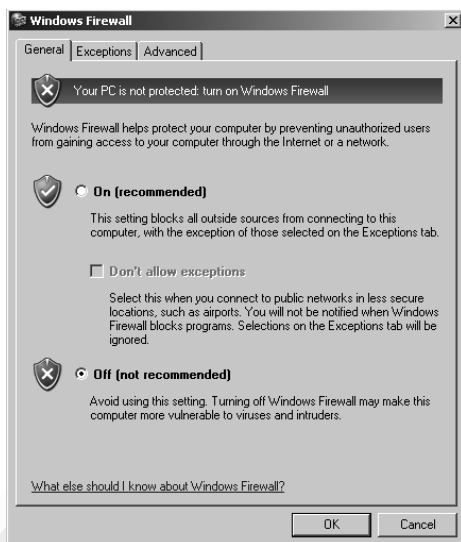
Слика 2.2.1-1- Пример хардверског филтера пакета, уређај Cisco ASA 5505

Насупрот хардверским филтерима пакета стоје филтери пакета који су реализовани у виду софтвера и могу се инсталирати на постојеће рачунарске системе. Оваквих решења има заиста много а посебну пажњу треба обратити на заштитне филтере пакета који су уграђени у популарне оперативне системе. На пример, Линукс оперативни систем поседује функцију филтрирања мрежних пакета уграђену у језгро још од верзије 2 (1998. година).



Слика 2.2.1-2 - Укључивање функције филтрирања пакета у језгру Линукса

У области корисничких оперативних система компанија Мајкрософт нуди функцију филтрирања пакета почев од *Windows XP SP2* система, а пре ове верзије је та функција била делимично доступна под именом *Internet Connection Firewall*. Када су у питању серверски оперативни системи, први систем који је добио овакву подршку је *Windows Server 2003*.



Слика 2.2.1-3 - Филтер пакета у MS Windows XP оперативном систему

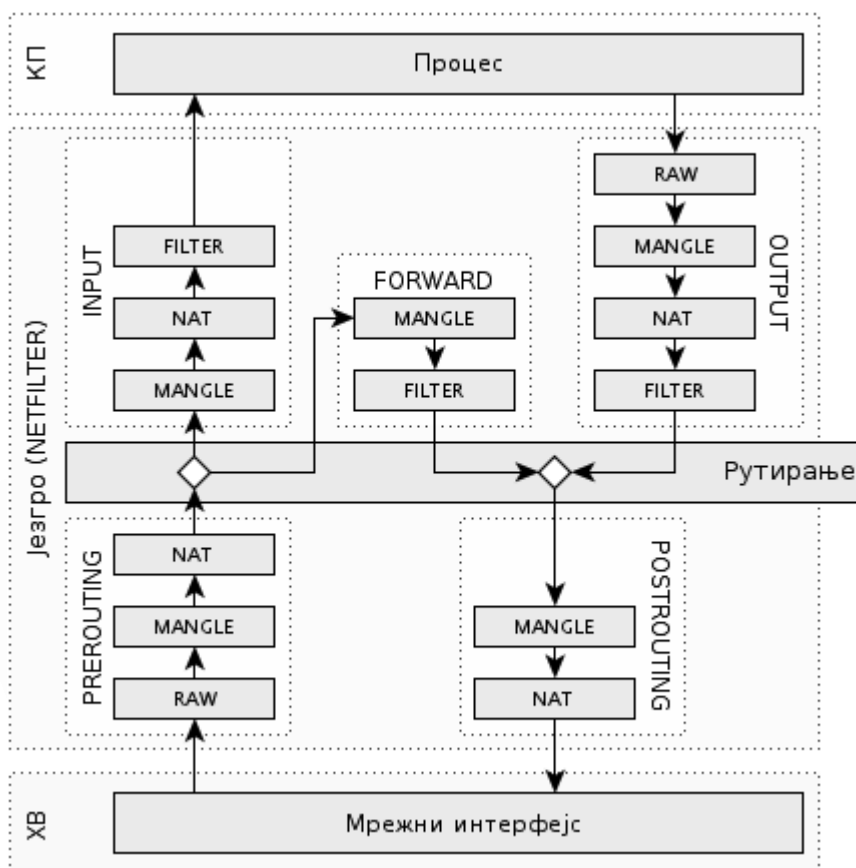
Осим функција филтрирања пакета интегрисаних у језгро оперативног система, пре свега код *Windows* оперативних система, постоји и велики број решења која су развили други произвођачи (на пример, произвођачи антивирусног софтвера).

## 2.2.2. Филтрирање пакета у Линукс оперативном систему

Линукс оперативни систем поседује филтер пакета уграђен у само језгро још од верзије 2. Филтер пакета Линукс оперативног система је реализован преко више компонената, од којих се неке налазе у језгру оперативног система, а неке као алати у корисничком простору. Основна компонента овог филтера пакета је *netfilter*, подсистем језгра, који директно комуницира са стеком мрежних протокола и мрежним хардвером. Са друге стране, у корисничком простору се налазе алати као што су *iptables* (за дефинисање правила за пакете *IPv4* протокола), *ip6tables* (за дефинисање правила за пакете *IPv6* протокола), *arptables* (за дефинисање правила за пакете *ARP* протокола) и други.

Једна од основних предности овакве архитектуре филтера пакета, односно његове реализације на ниском нивоу и унутар језгра оперативног система, јесу високе перформансе и безбедност. Наиме, рад на ниском нивоу омогућава рано одлучивање о даљој судбини пакета, а и скраћује пут који он пролази кроз систем. Додатно, модуларна архитектура омогућава развој различитих корисничких интерфејса, тако да данас имамо комплетне дистрибуције Линукса

које служе као филтери пакета за мрежу, а које имају графички или Веб кориснички интерфејс ка *netfilter-y*.



Слика 2.2.2-1 - Архитектура филтера пакета у језгру Линукса

Основна организациона јединица унутар *Netfilter*-а је табела. У оквиру табела се дефинишу ланци правила а основне табеле су:

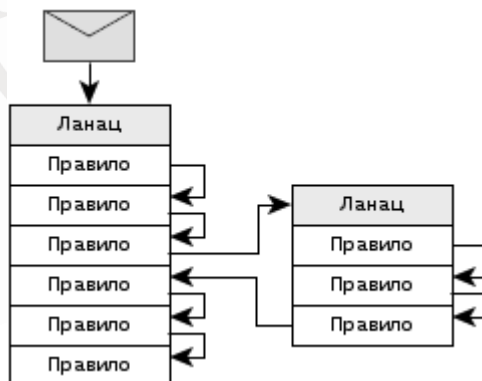
- *raw* - рад са пакетима код којих још увек није укључено праћење везе, односно искључивање праћења.
- *mangle* - рад са пакетима код којих се мењају подаци у заглављу (на пример, приоритет).
- *nat* - рад са пакетима код којих је потребно превођење мрежне адресе (изворишне или одредишне).
- *filter* - основно филтрирање улазних и излазних пакета.

Правила за филтрирање пакета групишу се у одговарајуће ланце правила. Два основна ланца правила код филтера пакета у Линуксу су улазни (*INPUT*), који се односи на пакете који су пристигли кроз одређени мрежни интерфејс а чије је одредиште локални рачунар, и излазни (*OUTPUT*), који се односи на све пакете који напуштају локални рачунар. Међутим, с обзиром на то да се филтер пакета у Линуксу, осим за заштиту локалног рачунара, може користити и за заштиту целокупних сегмената мреже или целих мрежа, трећи основни ланац (*FORWARD*) се односи на пакете који се прослеђују даље. Додатна два ланца правила, *PREROUTING* и *POSTROUTING*, служе за задавање правила пре, односно након процеса рутирања у језгру (односно одлучивања да ли је улазни пакет намењен локалном рачунару и на који излазни интерфејс треба усмерити излазни пакет).

За свако појединачно правило у ланцу, као и уопште за ланац, дефинише се циљна акција која ће се извршити над пакетима. У основне акције спадају:

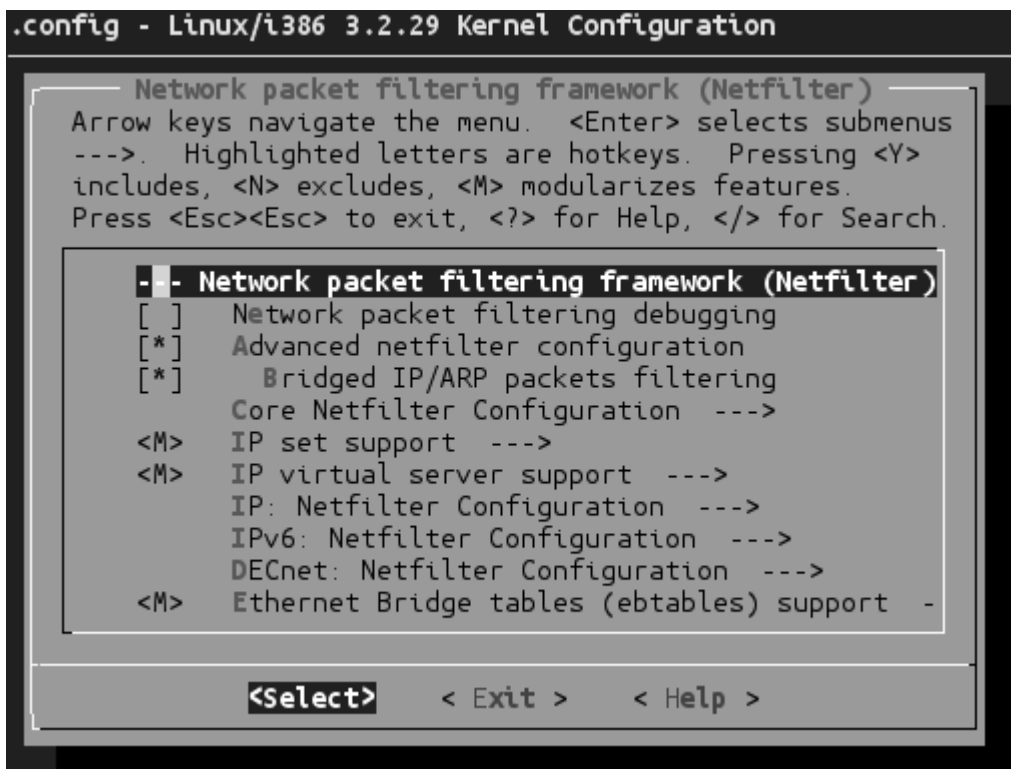
- *ACCEPT* - пакету се омогућава даљи пролазак кроз филтер
- *DROP* - пакет се одбацује без даљих акција.
- *REJECT* - пакет се одбацује а извору се шаље *ICMP* порука о томе.

Поред уграђених, корисници могу дефинисати и своје ланце правила. Кориснички дефинисани ланци правила се позивају путем правила у већ уграђеним ланцима. На овај начин се омогућава лакше управљање правилима, а уједно се избегава понављање истих група правила у различитим ланцима.



Слика 2.2.2-2 - Позивање кориснички дефинисаног ланца правила

Сам *Netfilter* је модуларан, односно састоји се од више компонената које се могу укључивати или искључивати по потреби. Конфигурисање компонената *Netfilter*-а се врши приликом подешавања самог језгра Линукса:



Слика 2.2.2-3 - Основне секције у подешавању подршке за Netfilter

Филтер пакета у Линукс оперативном систему има могућност да ради и у *statefull* и у *stateless* режиму рада, у зависности од тога да ли је укључена подршка за праћење веза:

#### ***IPv4 connection tracking support***

*Connection tracking keeps a record of what packets have passed through your machine, in order to figure out how they are related into connections.*

Осим за филтрирање пакета у циљу заштите, филтер пакета у језгру Линукса може се користити и за балансирање оптерећења путем креирања виртуалних сервера (кластера), што овај систем чини посебно погодним за коришћење код високо оптерећених Веб сајтова за чије је функционисање потребно више физичких сервера:



### **IP Virtual Server support**

*IP Virtual Server support will let you build a high-performance virtual server based on cluster of two or more real servers. This option must be enabled for at least one of the clustered computers that will take care of intercepting incoming connections to a single IP address and scheduling them to real servers.*

*Three request dispatching techniques are implemented, they are virtual server via NAT, virtual server via tunneling and virtual server via direct routing. The several scheduling algorithms can be used to choose which server the connection is directed to, thus load balancing can be achieved among the servers. For more information and its administration program, please visit the following URL: <http://www.linuxvirtualserver.org>*

У наставку су дати примери коришћења алата *iptables* за задавање најчешћих правила. На пример, уколико желимо да променимо општу полису за све пакете који улазе у наш рачунар, наредба је:

```
# iptables -P INPUT DROP
```

Након тога, уколико желимо да одобримо пролаз пакета који стижу кроз прву Етернет картицу, а адресовани су на *IP* адресу 192.168.55.12 и користе *TCP* протокол на порту 80, наредба је:

```
# iptables -A INPUT -i eth0 -p tcp --dport 80 -d 192.168.55.12 -j ACCEPT
```

Треба имати у виду да је редослед правила важан јер се даље упоређивање пакета прекида оног тренутка када пакет испуни све услове за неко правило. На пример, уколико желимо да одобримо све пакете који стижу преко прве Етернет картице и користе *TCP* протокол, али не и оне који користе порт 80, следећи редослед наредби:

```
# iptables -A INPUT -i eth0 -p tcp -j ACCEPT  
# iptables -A INPUT -i eth0 -p tcp --dport 80 -j DROP
```

не би дао жељене резултате. Разлог томе је што би сви пакети, па и они који

користе порт 80, испунили услове првог правила и ту би даља провера била прекинута. Исправан редослед правила за жељени резултат је следећи:

```
# iptables -A INPUT -i eth0 -p tcp --dport 80 -j DROP
# iptables -A INPUT -i eth0 -p tcp -j ACCEPT
```

Алтернативан начин за постизање жељеног редоследа правила, невезано за редослед наредби, је експлицитно навођење позиције на којој правило треба да се нађе у ланцу коме се додаје:

```
# iptables -I INPUT 2 -i eth0 -p tcp -j ACCEPT
# iptables -I INPUT 1 -i eth0 -p tcp --dport 80 -j DROP
```

Уколико желимо да уклонимо одређено правило из ланца, потребно је да наведемо његове параметре:

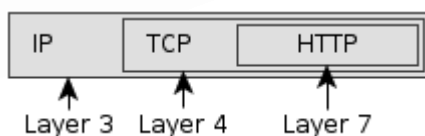
```
# iptables -D INPUT -i eth0 -p tcp --dport 80 -d 192.168.55.12 -j ACCEPT
```

За уклањање свих правила из ланца користи се параметар *-F* (скр. за *flush*):

```
# iptables -F INPUT
```

### 2.2.3. Нивои рада филтера пакета

Најнижи ниво рада филтера пакета подразумевано чини мрежни слој *OSI* комуникационог модела, мада одређена решења имају подршку и за рад са протоколима слоја везе података. Филтери пакета који раде на мрежном нивоу (познати и под именом *layer 3 firewall*) за доношење одлука о томе да ли ће пролаз одређеног пакета бити дозвољен или не, користе податке из заглавља протокола на мрежном слоју - подразумевано верзија 4 и 6 Интернет протокола.



Слика 2.2.3-1 - Нивои рада филтера пакета

У складу са тим, одлуке које овакви филтери пакета доносе засноване су првенствено на *IP* адресама пошиљаоца и примаоца. Другим речима, коришћењем филтера пакета који раде на мрежном слоју могуће је ограничити

који чланови мреже имају права да комуницирају међусобно, као и који су дозвољени смерови тих комуникација. Међутим, имајући у виду да су рачунарске комуникације ретко једносмерне, сама могућност дефинисања смера комуникације није у пракси посебно корисна.



Слика 2.2.3-2 - Структура пакета Интернет протокола верзије 4 (IPv4)

На следећем нивоу рада филтера пакета стоје *stateful* филтери пакета, односно решења која при доношењу одлуке о одређеном пакету у обзир узимају и тренутно стање везе којој тај пакет припада. Ови филтери пакета у обзир не узимају само податке из заглавља мрежног протокола, већ пакете анализирају у контексту везе којој припадају, а користе и податке из заглавља протокола на транспортном слоју (на пример, *Transmission Control Protocol*). Применом оваквих решења може се, на пример, дефинисати смер комуникације на основу тога ко има право да **иницира** успоставу везе. Након успешне успоставе везе пакети који јој припадају могу несметано да пролазе у оба смера.

Следећи ниво рада филтера пакета чине апликативни филтери, односно они који раде на 7. слоју *OSI* модела. Ови филтери, поред свих наведених могућности, имају и ту да могу да анализирају различите параметре пакета апликативног протокола. На пример, коришћењем таквог филтера могуће је забранити слање захтева ка Веб серверу који користе *POST* метод.

Приликом вредновања и избора филтера пакета треба имати у виду и њихове перформансе, односно капацитет. Филтери пакета сваки пакет који кроз њих пролази анализирају засебно, упоређујући га са скупом правила, а таква анализа захтева одређено процесорско време. Код филтера пакета који раде на мрежном слоју анализа је једноставнија, па је самим тим и утрошак

процесорског времена мањи. Насупрот томе, за анализу пакета код филтера пакета који раде на слоју апликације утрошак процесорског времена је већи јер је и сама анализа знатно сложенија. Другим речима, пропусна моћ филтера пакета који ради на слоју мреже већа је од пропусне моћи филтера пакета који ради на слоју апликације (посматрајући исту хардверску основу).



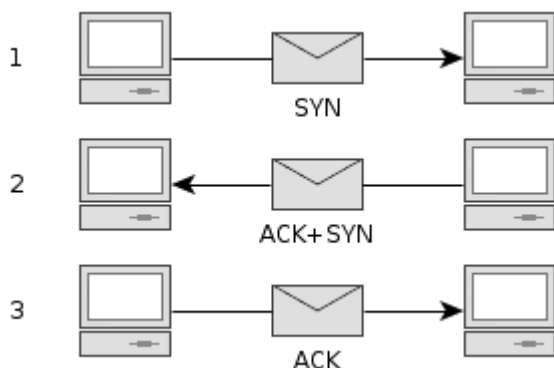
Слика 2.2.3-3 - Структура пакета протокола за контролу преноса (TCP)

Осим процесорског времена критичан ресурс код филтера пакета представља и радна меморија која се користи за складиштење пакета који чекају на обраду, као и за чување радних параметара (активне сесије и слично). У складу са свим наведеним избор филтера пакета треба вршити у складу са стварним и пројектованим потребама да би се оствариле захтеване функције и перформансе, односно да би филтер пакета могао да пружи одговарајући ниво перформанси, а да при том не постане „уско грло“.

На перформансе филтера пакета може значајно утицати то како су они подешени. На пример, два иста уређаја ће показати различите перформансе уколико је на једном унешено свега неколико правила, док се на другом налази велики број њих. Додатно, и редослед правила може имати значајан утицај, тако да ће боље перформансе имати филтер пакета код кога су „општија“ правила (она која се односе на већину пакета) постављена као прва у низу.

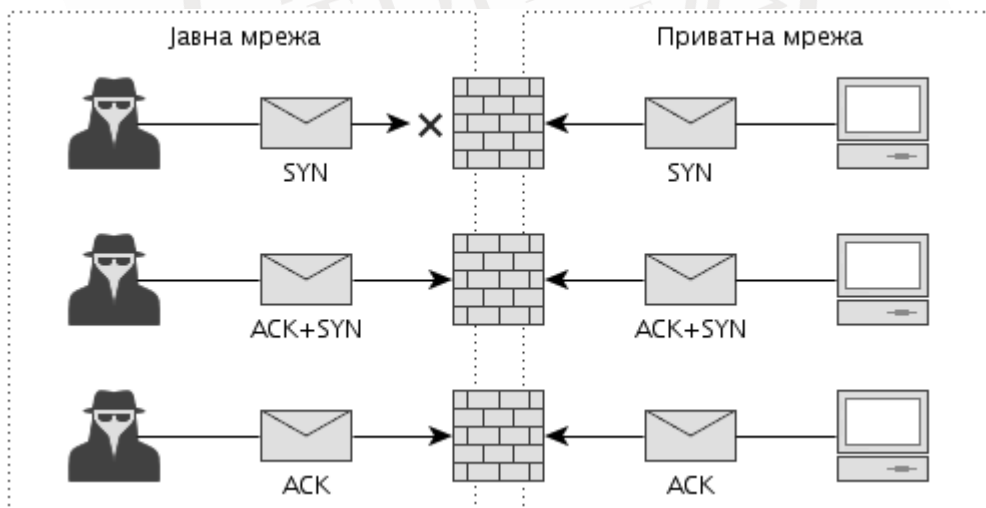
## 2.2.4. Филтери пакета и тројански коњи

Филтери пакета се подразумевано подешавају тако да клијенте у приватној мрежи заштите од приступа од стране потенцијалних нападача, односно рачунара који се налазе у спољној мрежи. Ова заштита се најједноставније постиже анализом заставица (енгл. *flag*) у заглављу *TCP* протокола (слика 1).



Слика 2.2.4-1 - Модел успоставе везе код *TCP* протокола

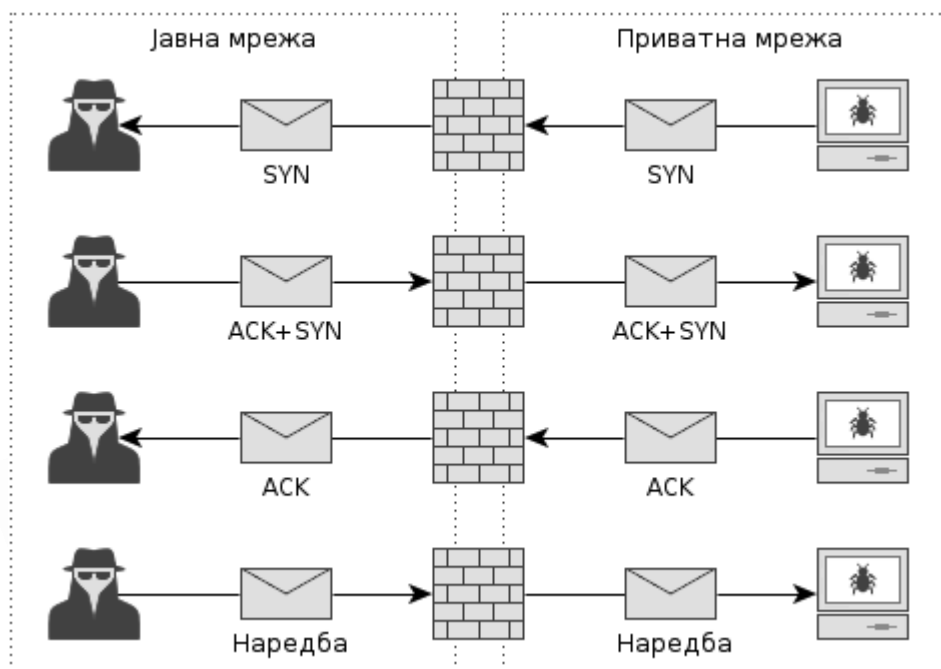
С обзиром на то да се за иницијализацију *TCP* везе користи пакет са укљученом **само** *SYN* заставицом, пролаз таквим пакетима се одобрава само уколико је њихов извор унутрашња мрежа (слика 2).



Слика 2.2.4-2 - Филтер пакета онемогућава иницирање комуникације споља

Да би превазишли овај проблем нападачи користе тзв. *back door* алате, односно алате који, симболично речено, отварају „задња врата“ у филтеру пакета. Ови

алати превазилазе описану заштиту тако што комуникацију са рачунаром у приватној мрежи (коју штити филтер пакета) не иницирају они, већ је веза иницирана од стране *back door* алата на зараженом рачунару (слика 3).



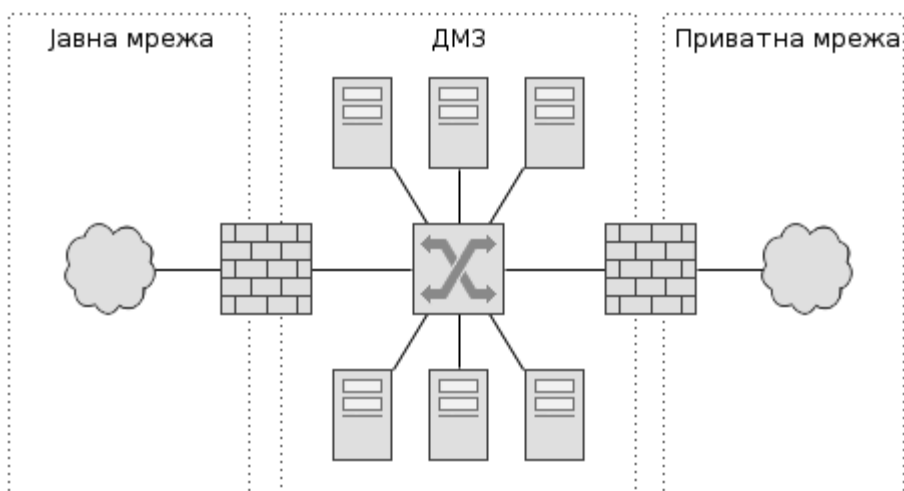
Слика 2.2.4-3 - Тројански коњи омогућавају заобилажење филтера пакета

За одбрану од оваквог типа напада потребно је користити најмање један, а по могућству и оба следећа метода. Као прво, потребно је одржавати рачунаре у приватној мрежи „чистим“, односно путем полиса и антивирусног софтвера онемогућити инсталацију малициозног софтвера описаног типа. Друго, потребно је прецизније дефинисати правила на филтеру пакета, односно ограничити скуп адреса, портова и сервиса које рачунари у приватној мрежи могу користити (што у пракси обично представља прилично сложен задатак).

## 2.2.5. Концепт демилитаризоване зоне

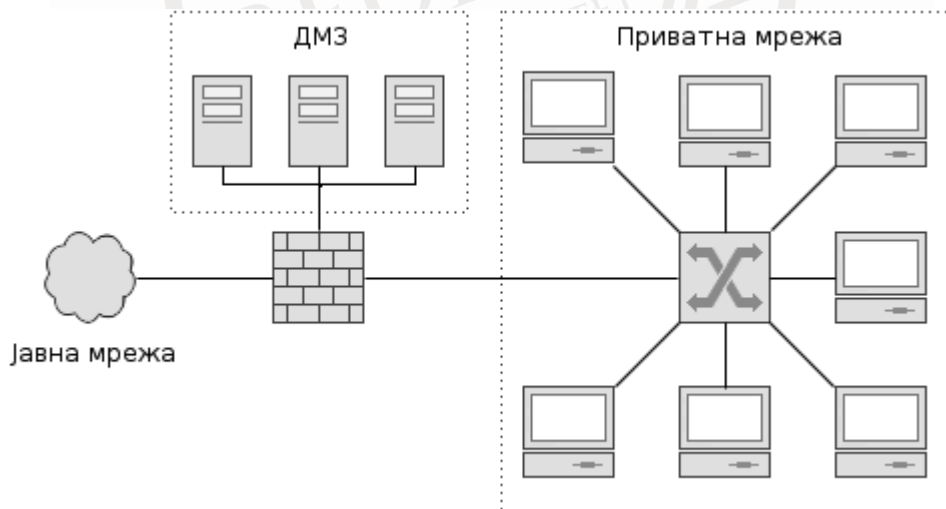
Демилитаризована зона (енгл. *demilitarized zone, DMZ*) је безбедносни мрежни концепт код кога се одређени део приватне мреже истура ка јавној мрежи да би му се из ње могло приступати. У демилитаризовану зону се најчешће смештају сервери са сервисима којима је потребно омогућити приступ из спољашње мреже. Са друге стране, демилитаризованом зоном се спречава даље

напредовање ка приватној мрежи.



Слика 2.2.5-1 - Демилитаризована зона са два филтера пакета

На слици 1 је приказана шема пуне демилитаризоване зоне, односно зоне која користи два филтера пакета - један према спољној а други према приватној мрежи. Ова два филтера пакета имају различита понашања, односно конекције које филтер између јавне мреже и демилитаризоване зоне дозвољава, филтер између демилитаризоване зоне и приватне мреже забрањује.



Слика 2.2.5-2 - Демилитаризована зона са једним филтером пакета

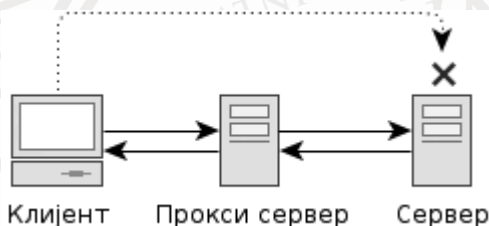
Демилитаризована зона се може креирати и коришћењем једног филтера

пакета (слика 2.). На тај начин се остварује уштеда у погледу ангажоване опреме, али се отежава конфигурисање самог филтера пакета.

Осим демилитаризовања целе зоне, у пракси се понекад користи и **демилитаризовани хост**, односно само један члан приватне мреже се излаже јавној мрежи. Овај концепт, међутим, може бити опасан у случају да дође до компромитовања изложеног хоста, с обзиром да у таквој конфигурацији не постоји додатна заштита између њега и остатка мреже.

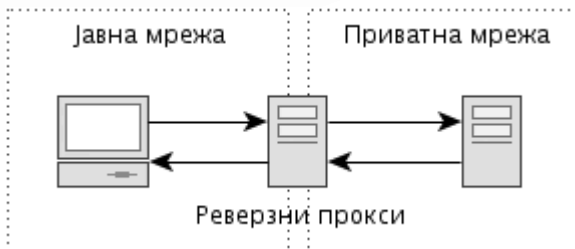
## 2.2.6. Прокси сервери

Прокси сервери су мрежни уређаји, хардверски или софтверски, који омогућавају посредан приступ осталим рачунарима или ресурсима у мрежи. У суштини, у питању су филтери пакета, који најчешће раде на седмом слоју *OSI* модела, односно који имају подршку за разумевање (апликативних) протокола сервиса за које пружају услугу посредовања. У случају рада на нижем нивоу, прокси сервером би се могли сматрати и, на пример, рутери који врше превођење мрежних адреса.



Слика 2.2.6-1 - Улога прокси сервера

У зависности од конфигурације клијента прокси сервери се могу поделити на обичне и реверзне (обрнуте). Код обичних прокси сервера клијент експлицитно на свом рачунару (нпр. унутар Веб браузерa) задаје адресу и остале параметре прокси сервера путем кога жели да индиректно приступа ресурсима.



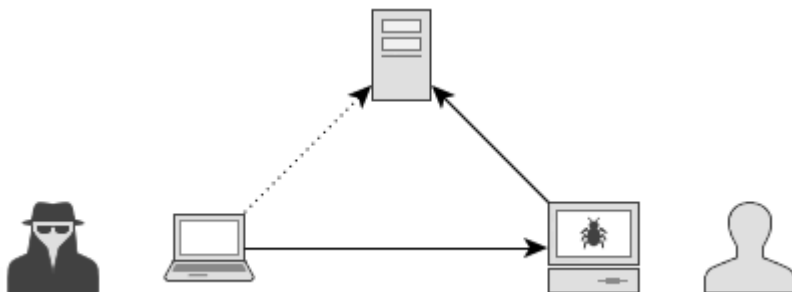
Слика 2.2.6-2 - Реверзни прокси сервер

Са друге стране, за коришћење реверзних прокси сервера (слика 2) није



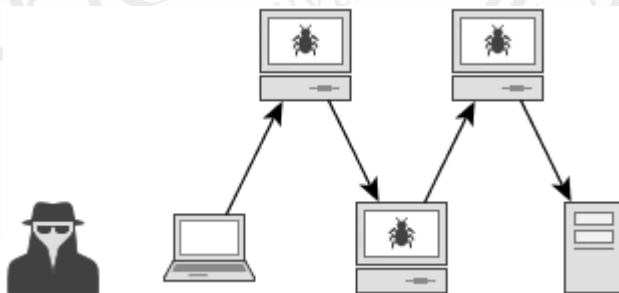
потребна додатна конфигурација клијента, чак он често нема никаквих назнака да је у питању посредовано приступање ресурсу.

Прокси сервери се често користе за сакривање идентитета и локације особе која приступа одређеном мрежном ресурсу. На пример, нападачи често остављају могућност да на претходно компромитованим рачунарима укључе функцију посредовања у комуникацији, како би касније те рачунаре могли да користе за анонимно приступање другим рачунарима на Интернету (слика 3).



Слика 2.2.6-3 - Употреба компромитованих рачунара као прокси сервера

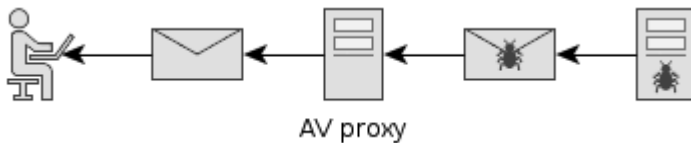
За појачан ефекат постизања анонимности, односно онемогућавања праћења конекције до њеног извора, нападачи често уланчавају компромитоване рачунаре које користе као прокси сервере (слика 4). Последица тога је да се код покушаја праћења извора одређене конекције често долази до тачака које су расуте по различитим континентима.



Слика 2.2.6-4 - Употреба вишеструких посредника са циљем анонимности

Са безбедносног аспекта су, пре свега реверзни прокси сервери корисни и са те стране што могу стварне сервере, односно сервере ка којима врше посредовање, да заштите од различитих врста напада, као и да им побољшају перформансе. На пример, реверзни прокси сервер се може користити за амортизовање разлика у брзини слања података са сервера и могућности

пријема података на клијенту (такозвано „храњење на кашичицу“, енгл. *spoon feeding*). Или, додавањем прокси сервера који врши компресовање садржаја који се шаље растеређује се централни процесор апликативног сервера.



Слика 2.2.6-5 - Антивирусни прокси сервер штити кориснике који сурфују Вебом

Једна од честих улога прокси сервера је и анти-вирусна заштита. Она се постиже тако што се од корисника у приватној мрежи захтева да користе прокси сервер који има могућност анализирања саобраћаја, односно откривања малициозног софтвера. Прокси сервери овог типа, у случају откривања малициозног саобраћаја, не дозвољавају да он стигне до корисника у приватној мрежи.

## 2.3. Системи за откривање и спречавање напада

Системи за откривање напада (енгл. *Intrusion Detection System, IDS*) надгледају дешавања у посматраном рачунарском систему или рачунарској мрежи, откривају сумњиве активности, бележе их у дневнике и о њима обавештавају администраторе. Са друге стране, системи за спречавање напада (енгл. *Intrusion Prevention System, IPS*) омогућавају блокирање даљих активности низа за који је утврђено да представља напад. Оба описана типа система могу радити на нивоу једног рачунара (енгл. *Host-Based IDPS*) или рачунарске мреже (енгл. *Network-Based IDPS*).

## 2.4. Бележење активности у дневнике догађаја

У случају да дође до кvara одређеног рачунарског система, или се посумња да је извршен напад на њега, веома је важно имати податке о активностима које су претходиле јављању проблема. Из тог разлога већина серверског софтвера садржи интегрисану функцију за бележење активности у **дневнике догађаја** одговарајуће текстуалне датотеке или базе података (енгл. *log file*). У наставку је дат пример садржаја фајла `/var/log/secure` у који се бележе активности које могу имати везе са безбедношћу система:

...

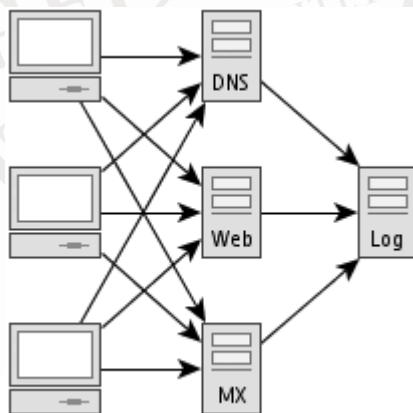
```
Jul 30 17:44:27 sshd[18055]: Accepted password for cp6 from 178.221.230.39
```

```

Jul 30 17:44:27 sshd[18055]: pam_unix: session opened for user cp6
Jul 30 17:48:37 sshd[18055]: Received disconnect from 178.221.230.39
Jul 30 17:48:37 sshd[18055]: pam_unix: session closed for user cp6
Aug 29 09:38:45 sshd[25968]: Invalid user ajevremovic from 82.117.206.61
Aug 29 09:38:45 sshd[25969]: input_userauth_request: invalid user ajevremovic
Aug 29 09:38:45 sshd[25969]: Connection closed by 82.117.206.61
Aug 29 09:45:04 sshd[26154]: pam_unix(sshd:auth): authentication failure;
Aug 29 09:45:06 sshd[26154]: Failed password for cp6 from 82.117.206.61
Aug 29 09:45:11 sshd[26154]: Failed password for cp6 from 82.117.206.61
Aug 29 09:45:15 sshd[26154]: Accepted password for cp6 from 82.117.206.61
...

```

Из приказаног садржаја можемо закључити да се корисник *cp6* успешно пријавио на систем преко *SSH* сервиса 30. јула у 17:44 (часова) и да је на систему радио око четири минута, након чега се одјавио. Након тога није било повезивања са системом путем *SSH* сервиса све до 29. августа када је удаљени корисник прво покушао да се на систем пријави са непостојећим налогом *ajejremovic*, а затим два пута унео погрешну лозинку за налог *cp6*. Након тога је корисник унео исправну лозинку за тај налог и успешно се пријавио за удаљени рад на систему.



Слика 2.4-1 - Евидентирање активности на посебном серверу

Приказане информације су често драгоцене у препознавању напада који су у току, као и напада који су се већ десили. Међутим, један од основних проблема који се јавља код напада који су успешно извршени је измена садржаја у

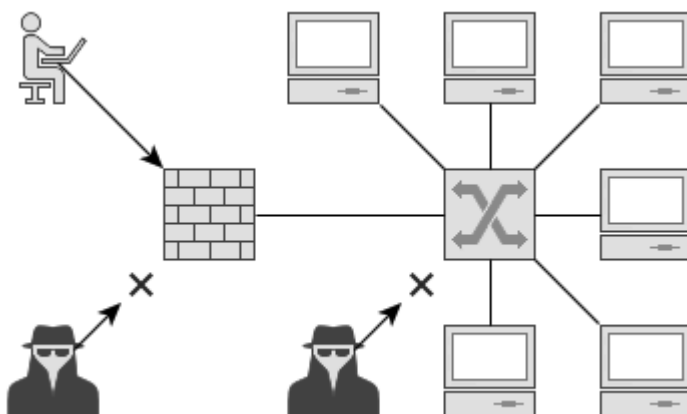
дневницима догађаја од стране нападача. На пример, у претходном примеру је потенцијални нападач два пута унео погрешну лозинку за налог *сrb* а трећи пут, претпоставимо, погодио исправну. Све ово је забележено у дневник догађаја */var/log/secure*. Међутим, уколико нападач након успешног пријављивања има могућност измене овог фајла (конкретно, за измену овог фајла су потребне привилегије администратора), он из њега може уклонити све записе који указују на његов напад. У том случају би администратор сервера на који је извршен напад, након прегледања овог дневника догађаја стекао лажни утисак да је са сервером све у реду, односно да напада није било.

Да би се поменути пропуст спречио у пракси се користе посебни сервери (тзв. *log server*) на којима се чувају дневници догађаја који су се десили на осталим рачунарима (слика 1). Ови сервери су максимално обезбеђени, имају минималан број активних сервиса, а рад на њима преко мреже није омогућен. Због једноставне функције коју обављају немају значајне процесорске и меморијске ресурсе, већ само велику количину спољне меморије за складиштење дневника догађаја. На овај начин се нападачу онемогућава да уклони трагове свог напада. При том је веома важно да часовници свих сервера буду синхронизовани, да би се у каснијој анализи догађаја на различитим рачунарима могли успешно повезати.

## 2.5. Контрола приступа приватним мрежама

Једна од безбедносних претпоставки код приватних рачунарских мрежа је да су њихови канали заштићени од прослушкивања, а да се у њихове чланове има поверења, односно да они немају лоше намере. Оваква претпоставка је често у великој мери тачна - када су у питању мале приватне мреже, које се налазе у физички изолованом простору и које нису повезане са Интернетом. Међутим, велики број приватних рачунарских мрежа данас има и по више стотина чланова, мрежна инфраструктура није физички заштићена а могућност приступа приватних чланова Интернету се данас подразумева.

Да би се исправно заштитила приватна рачунарска мрежа мора се остварити потпуна контрола приступа самој мрежи и ресурсима/функцијама у њој. Под контролом приступа подразумева се приступ мрежи од споља (на пример, са Интернета), али и ко има право да постане њен члан. На пример, уколико је приватна мрежа од спољног приступа правилно заштићена употребом филтера пакета, а нападач има могућност да физички прикључи свој рачунар путем слободне утичнице на комутатору, јасно је да ће безбедност такве мреже бити на веома ниском нивоу.

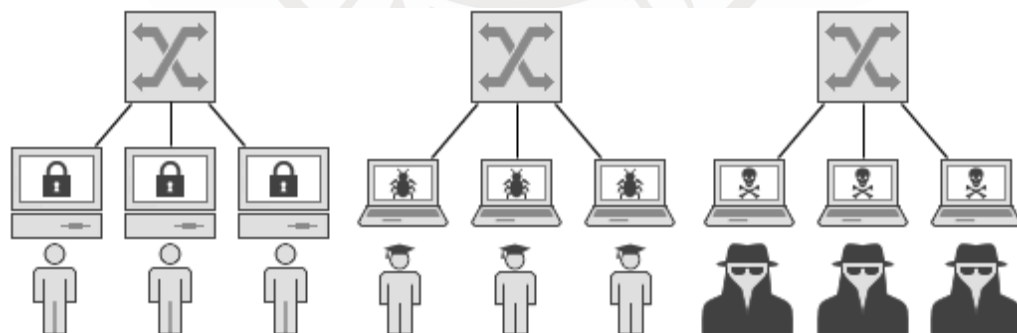


Слика 2.5-1 - Контрола приступа приватној мрежи и споља и изнутра

Две основне технологије за контролу приступа у приватним рачунарским мрежама које користе Етернет технологију су виртуалне мреже на локалном подручју и контрола приступа коришћењем *IEEE802.1X* протокола.

### 2.5.1. Виртуалне мреже на локалном подручју

Једну физичку рачунарску мрежу у пракси често користи велики број корисника различитог профила: администратори, менаџмент, привремено ангажовано особље, гости... Организација која је власник те мреже у ове профиле корисника има различите нивое поверења и циљ је да се њихове могућности ограниче у складу са тим. У идеалној ситуацији ови корисници су физички, односно тополошки груписани те их је једноставно контролисати изоловањем у одвојене мрежне сегменте који су међусобно повезани филтером пакета (слика 1).



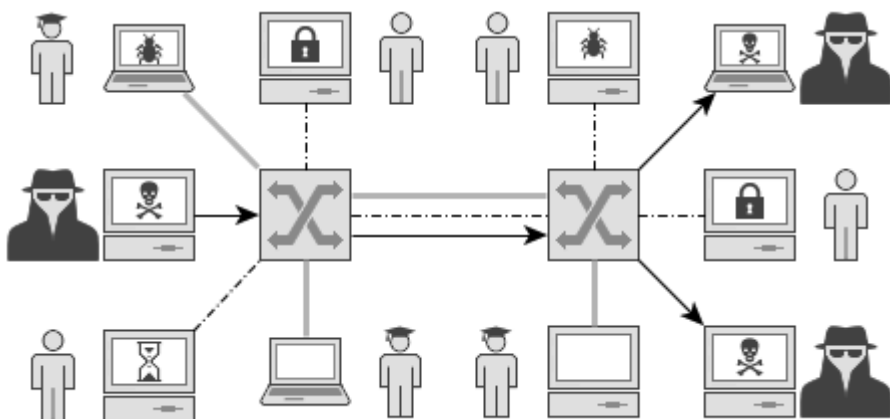
Слика 2.5.1-1 - Идеална ситуација: различити профили корисника су груписани

У пракси се, међутим, приказана идеална ситуација ретко среће. Корисници

могућност да се повезивањем на приватну мрежу оствари контакт са свим осталим члановима у огромној мери поједностављује нападе извиђања и компромитовање циљних ресурса у њој. Треба имати у виду и то да корисници са високим привилегијама углавном немају лоше намере али да се њихови администратори, чиновари, употребом малициозног софтвера, понекад могу искористити за пад на друге ресурсе у мрежи.

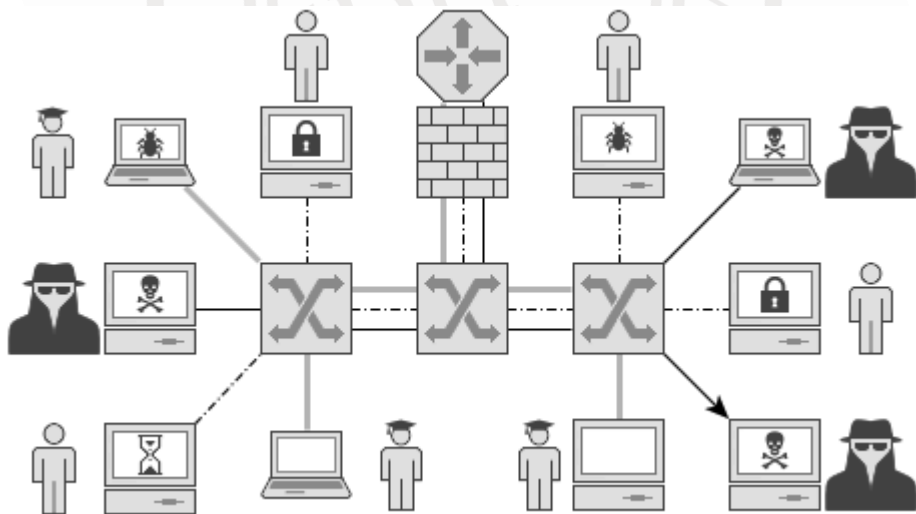
складу са структуром наведеног безбедносног ризика јавила се потреба да се омогући груписање чланова различитих профила, као и да комуникација између тих група стави под контролу. Решење се јавило у виду **виртуалних мрежа на локалном подручју** (енгл. *Virtual LAN, VLAN*) које омогућиле да се унутар једне физичке Етернет мреже дефинише више виртуалних мрежа чији ће саобраћај бити међусобно изолован. Ово решење стандардизовано од стране *IEEE* под ознаком **802.1Q**.

У складу са структуром наведеног безбедносног ризика јавила се потреба да се у пракси омогући груписање чланова различитих профила, као и да се комуникација између тих група стави под контролу. Решење се јавило у виду **виртуалних мрежа на локалном подручју** (енгл. *Virtual LAN, VLAN*) које су омогућиле да се унутар једне физичке Етернет мреже дефинише више виртуалних мрежа чији ће саобраћај бити међусобно изолован. Ово решење је стандардизовано од стране *IEEE* под ознаком **802.1Q**.



Слика 2.5.1-3 - Дозвољена је комуникација само између чланова истог VLAN-а

На први поглед може деловати нелогично потреба да се комуникација између различитих група корисника прво онемогући, укључивањем једне нове технологије, а затим омогућава укључивањем још једне. Међутим, треба имати у виду да се на овај начин комуникација централизује у једну тачку и контролише одговарајућим правилима рутирања и филтера пакета. Такву контролу није могуће обавити на слоју везе, односно самом Етернет технологијом.



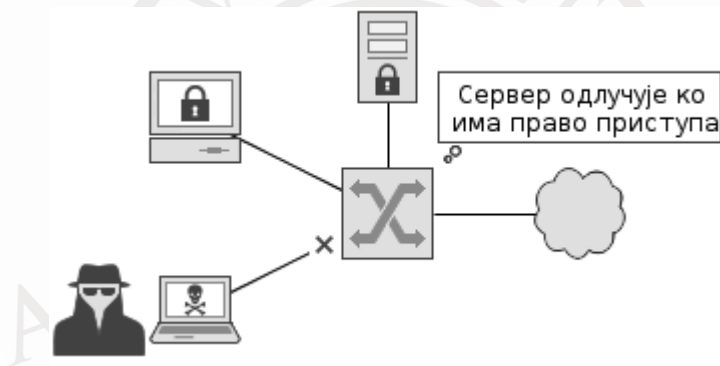
Слика 2.5.1-4 - Комуникација између VLAN-ова иде преко рутера

Следећа ствар коју треба имати у виду код коришћења рутера за комуникацију између различитих виртуалних мрежа је пад перформанси у комуникацији.

Разлог томе је знатно нижа пропусна моћ рутера (најчешће до пар стотина Mb/s) у односу на савремене комутаторе (1Gb/s и више). Да би се ова разлика надокнадила потребно је уместо рутера користити комутаторе који раде на трећем слоју (енгл. *Layer 3 switch - L3 switch*). Такви комутатори су у стању да изведу основне операције рутирања при брзини комутације (енгл. *wire speed routing*). Са друге стране, употреба виртуалних мрежа има и позитиван утицај на перформансе јер се повећава број емисионих домена, односно смањује се њихова величина.

### 2.5.2. Контрола приступа путем IEEE 802.1X протокола

Протокол *IEEE 802.1X* омогућава контролу приступа приватној мрежи на основу корисничког имена и лозинке или дигиталног сертификата. Најчешће се користи код приватних рачунарских мрежа заснованих на Етернет технологији, али се може применити и на бежичне *IEEE 802.11* мреже.



Слика 2.5.2-1 - Комутатор се консултује са аутентикационим сервером

У првој фази овог протокола крајњи уређај који жели да се повеже на мрежу (енгл. *supplicant*) доставља комутатору, са којим је физички повезан (енгл. *authenticator*), одговарајуће корисничко име и лозинку или дигитални сертификат. Комутатор се, затим, обраћа аутентикационом серверу који на основу достављених података одређује да ли ће се поменутом крајњем уређају дозволити повезивање са мрежом или не. Треба имати у виду и то да подршку за овај протокол имају сви популарни кориснички оперативни системи.

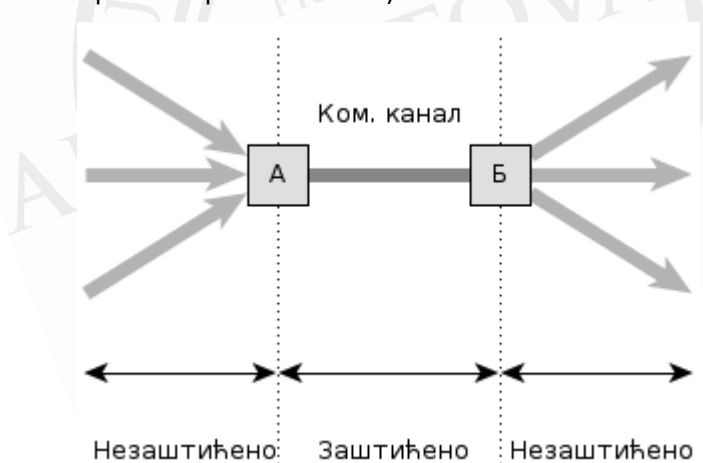


### 3. Безбедност на физичком нивоу

У рачунарским телекомуникацијама заштиту података могуће је остварити и на физичком нивоу. Заштита података на физичком нивоу (комуникационом каналу) примењује се из неколико разлога:

1. Једним уређајем се штите сви подаци на комуникационом каналу
2. Често код комуникационих уређаја не постоји хардверска и софтверска документација и није их могуће модификовати, тј. додавати криптографске функције

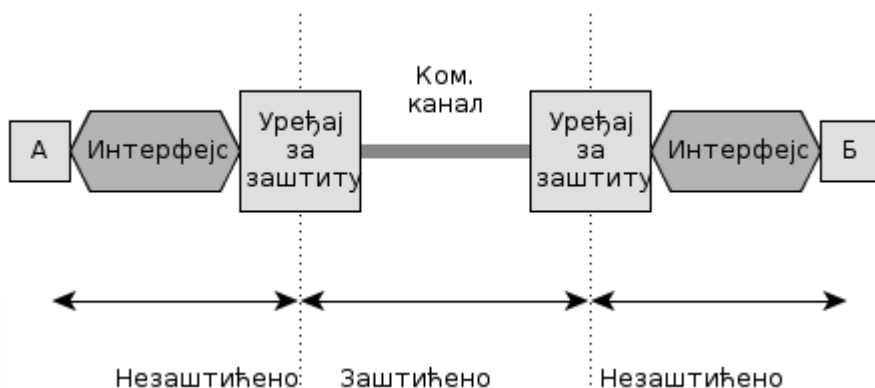
Овакав начин заштите је познат у литератури као заштита од тачке до тачке (енгл. *point-to-point*). Суштина овог принципа је да се једним уређајем штите сви долазећи подаци, који су најчешће мултиплексирани и долазе од различитих извора информација. Најчешћа ситуација је да постоји поверљива (обезбеђена) мрежа једног имаоца (војска, полиција, железница, банка и слично) и да је податке такве мреже потребно штитити када се за комуникацију са удаљеним чворовима користи јавна небезбедна мрежа (као што је Интернет, јавна комутирана телефонска мрежа и слично).



Слика 3-1 - Заштита комуникација од тачке до тачке

Други принцип у заштити комуникација је заштита с краја на крај (енгл. *end-to-end*). Овај принцип је идеал у заштити података - информација се штити од њеног изворишта до одредишта. Међутим, заштита с краја на крај захтева да сваки терминал има уграђене криптографске функције (хардвер или софтвер), што је најчешће веома скупо и неизводљиво.

Заштита на комуникационом каналу се реализује додатним физичким уређајем у чију реализацију постоји поверење. Додатни уређај има одговарајуће интерфејсе према комуникационом каналу и према телекомуникационом уређају чији подаци се штите. Најчешће се поштују индустријски стандарди код реализација интерфејса, а на комуникационом каналу се поштују стандардне брзине (2 Mb/sec, 1 Gb/sec и слично). Према стандардима, то су високе битске брзине, а за пренос битова користе се одговарајући комуникациони протоколи.



Слика 3-2 - Заштита комуникација с краја на крај

Пре постављања уређаја за заштиту, комуникација удаљених чворова се реализује преко одговарајућих комуникационих уређаја А и Б (слика 2). Ови уређаји одржавају телекомуникациону синхронизацију која разрешава проблеме кашњења због преноса бита и утицаја сметњи (шума) и која се одвија по одговарајућем протоколу. Када се додају (повежу) уређаји за заштиту, они такође одржавају телекомуникациону синхронизацију, а пре овог морају да реше криптолошку синхронизацију. Криптолошка синхронизација се односи на договарање криптолошких параметара шифарског система на предајној и пријемној страни, што захтева пренос додатних података (почетни спољашњи кључ, редни број радног симетричног тајног кључа и сл.). Ова синхронизација уноси додатно кашњење које одговара времену потребном за припрему алгоритама за шифровање и мора да буде веома робусна на неминовне сметње и шум који постоје на комуникационом каналу.

Дакле, пре успостављања телекомуникационе синхронизације у оваквим системима се прво успоставља криптолошка синхронизација. Оваква криптолошка синхронизација је позната под називом почетна криптолошка синхронизација. То значи да када се договоре криптолошки параметри и када уређаји за заштиту остваре криптолошки синхронизам и пређу у синхроно

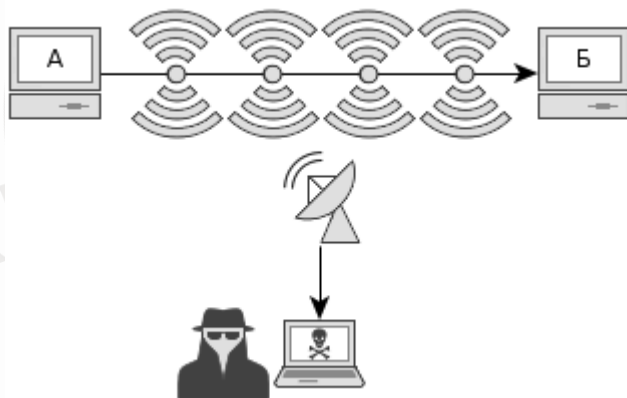
шифровање и дешифровање, на даље се одржава телекомуникациона синхронизација уређаја А и Б. Након почетне криптолошке синхронизације нема више додатних података за пренос и комплетан саобраћај (низ бита) који долази од уређаја А шифрује се и преноси до следећег уређаја за заштиту података који врши дешифровање и прослеђивање таквих података одредишту Б, а затим се подаци демултиплексирају и прослеђују крајњим терминалима. Наравно, у комуникацијама са пуним дуплексом заштита података се обавља у оба смера и постоји независна криптолошка синхронизација за сваки од смерова комуникације.

Због утицаја сметњи на комуникационом каналу дешава се да се уређаји за заштиту расинхронизују (погрешно шифровање и дешифровање). Сами уређаји за заштиту то тешко могу да примете. Међутим, због овог долази до прекида телекомуникационе синхронизације, што одмах региструју уређаји А или Б и преко одговарајућег интерфејса дају сигнал уређајима за заштиту да изврше поновну криптолошку синхронизацију. Овакав процес се назива криптолошка ресинхронизација и она најчешће одговара почетној криптолошкој синхронизацији. Последица је да се због криптолошке ресинхронизације уноси додатно кашњење у комуникацију. Пошто су овакви системи намењени за рад непрекидно у току 24 часа, процес криптолошке ресинхронизације се обавља аутоматски. У условима када су комуникационе линије лоше и постоји висок ниво шума на каналу број криптолошких ресинхронизација може да утиче на деградирање квалитета телекомуникационих сервиса.

Постоје различити сценаријуми око криптолошке ресинхронизације. На пример, у условима када су комуникациони канали идеални, као што је случај код оптичких комуникација, из криптолошких ралога би било лоше да се криптолошка ресинхронизација никада не обави. Због тога се уводи и концепт периодичне криптолошке синхронизације, која се намерно изазива како би уређаји за заштиту обновили криптолошке параметре и наставили шифровање и дешифровање са нових почетних позиција. Такође, код полудуплексних или чистих симплексних комуникација, неопходно је да се обави периодична криптолошка синхронизација. Наиме, када би предајни уређај из неког разлога почео погрешно да шифрује не би постојао начин да га пријемни уређај за заштиту обавести о томе и да га натера да пређе у режим криптолошке ресинхронизације.

### 3.1. Компромитујуће електромагнетно зрачење

Електронски уређаји, као што су рачунари и рачунарска опрема, за обављање инструкција и операција над подацима на најнижем нивоу користе дигиталне сигнале који се преносе путем магистрала између дигиталних компонената. Рад магистрала је усклађен са одговарајући тактом који се генерише на основу кристала кварца. Управо је једна од најважнијих карактеристика савремених рачунара радни такт (процесора, осталих чипова, магистрале и слично) и он се данас изражава у гигахерцима (GHz). На тако високим фреквенцијама настају интерференције сигнала на магистралама. Дигитални сигнали којим се преносе подаци увек имају ограничен спектар и тачно се зна у ком делу спектра се налази спектар сигнала који се преноси. Међутим, због интерференције сигнала овај спектар се шири, а последица је да се неке информације могу наћи у нежељеним деловима спектра. Како се магистрале могу посматрати и као скуп жица (водова), оне се понашају као антена, преко које се путем електромагнетног зрачења нежељени сигнали појављују у околини самог електронског уређаја.



Слика 3.1-1 - Компромитујуће ЕМ зрачење каблова омогућава прислушкивање

Компромитујуће електромагнетно зрачење је термин који се најчешће користи на уређајима који врше шифровање/дешифровање. Иако се операције шифровања и дешифровања раде без грешака, у току обраде података на електронском уређају настаје компромитујуће електромагнетно зрачење (нежељена емисија). Последица је да се у околини уређаја, анализом спектра нежељеног ЕМ зрачења може издвојити (детектовати) отворена информација, тајни кључ шифарског алгоритма или неки други криптографски параметар. У западној литератури се овај проблем посебно третира кроз одговарајуће

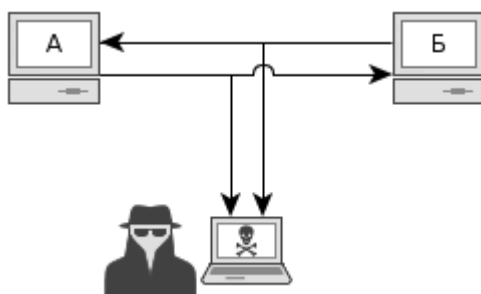
стандарде и носи назив *TEMPEST (Transient Electro Magnetic Pulse Emanation Standard)*.

Разликују се два термина: кондукционо и емисионо зрачење. Кондукционо зрачење се односи на отицање нежељених информација кроз “жице”, као што су комуникациони каблови или уземљење. Емисионо зрачење се односи на нежељено електромагнетно зрачење у околини уређаја. Заштита од кондукционог зрачења се спроводи галванским раздвајањем извора напајања од самог уређаја. Поред тога сви излази из уређаја морају да буду филтрирани нископропусним (НФ) филтрима, како би се потиснула евентуална отворена информација у неком високом делу спектра сигнала који се преноси. Наравно, НФ филтри су подешени да пропуштају користан сигнал.

Заштита од емисионог зрачења је знатно компликованија. Спроводи се оклапањем уређаја, где је оклоп тако израђен (материјал, дебљина и сл.) да максимално потисне сигнале у високом делу спектра. Поред тога, пожељно је коришћење оптичких каблова према свим периферијама, како би се отклонила могућност да они буду извор нежељеног зрачења. Познато је да се свет данас дели на развијене и неразвијене. Развијени свет располаже такозваном високом технологијом, која је најчешће под рестриктивним извозним дозволама. Најрестриктивнија је технологија којом се могу открити сигнали ниског нивоа у веома високим деловима спектра. Данас је једноставно прислушкивати када се опрема може поставити у близини уређаја за заштиту, али се поставља питање да ли је могуће прислушкивати компромитујуће електромагнетно зрачење са огромне удаљености (као што су сателити). Ако је могуће, тада не постоји потреба за дешифровањем, које је увек сложено. Управо због овога, у професионалним шифарским системима уређаји за шифровање се обавезно постављају у обезбеђен простор (као што су Фарадејеви кавези). Постоји и алтернатива, а то су оклопљени рачунари и други уређаји, који би могли да се користе у собним условима. Међутим, њихова цена је неколико десетина пута већа од цене стандардне рачунарске опреме.

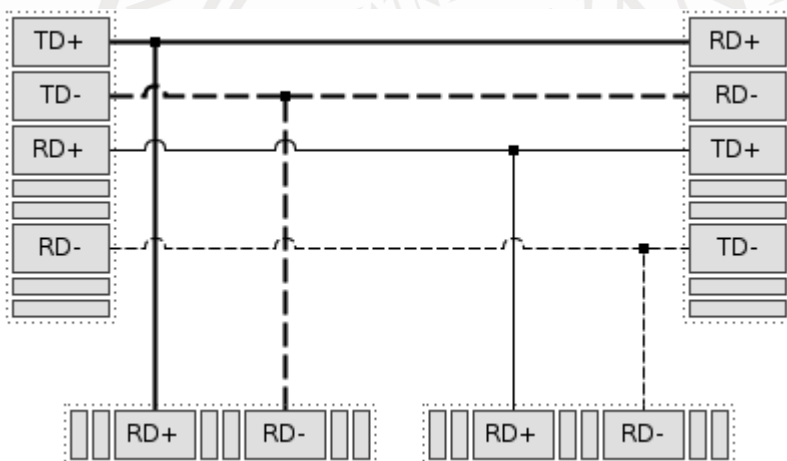
## **3.2. Пасивни мрежни вентили**

Пасивни мрежни вентили су физички уређаји који омогућавају копирање података који се преносе мрежним каналом, без давања икаквог наговештаја о томе странама које комуницирају. У пракси се често користе за „прислушкивање“, односно за неовлашћено снимање садржаја мрежних комуникација.



Слика 3.2-1 - Принцип рада пасивног мрежног вентила

Један од најједноставнијих пасивних мрежних вентила је онај који се прави за Етернет везе брзине 10-100Mb/s путем каблова са упреденим парицама. Код оваквих вентила се за жице мрежног кабла који се прислушкује везују жице два додатна мрежна кабла (слика 2). Правилним повезивањем сви подаци који су послати од стране једног пошиљаоца шаљу се и кроз први мрежни кабл, а подаци послати од стране другог учесника у комуникацији прослеђују и кроз други мрежни кабл.



Слика 3.2-2- Пасивно прислушкивање 10-100Mb/s каблова са упреденим парицама

Неки мрежни уређаји, као на пример неки комутатори и рутери компаније Циско, на себи већ поседују могућност да се сав или одређени саобраћај који пролази кроз уређај ископира и пошаље кроз одређени порт. Међутим, треба бити опрезан при коришћењу ове функције пошто копирање саобраћаја са више линкова може довести до загушења и пада брзина комуникације.

### 3.3. Шифровани системи фајлова

*Масовна повреда података, којом је угрожена лична и медицинска документација милиона војних пацијената и њихових породица, догодила се у Сан Антонију када је евиденција украдена из аутомобила радника компаније која је пружала услугу складиштења података. Информације око 4,6 милиона активних и пензионисаних војних лица и њихових породица, налазиле су се на бекап тракама електронске здравствене базе која се користи од 1992. године...*

*Ројтерс, 29. септембар 2011. године<sup>2</sup>*

*3. маја 2006. године, аналитичар података компаније Ветеранс Аферс понео је кући лаптоп и екстерни хард-диск са нешифрованим подацима 26,5 милиона ветерана америчке војске и њихових породица. Рачунарска опрема је украдена приликом провале у кућу аналитичара.<sup>3</sup>*

*Consumer Affairs, мај 2006. године*

Једна од важних заштита на физичком нивоу је шифровање података приликом њиховог складиштења на спољну меморију. Ово се може постићи на различите начине, на пример шифровањем података директно у апликацији у којој корисници раде. Овакав приступ је добар из разлога што се скраћује пут података кроз систем у отвореном облику. Међутим, постоје два основна недостатка шифровања података на слоју апликације:

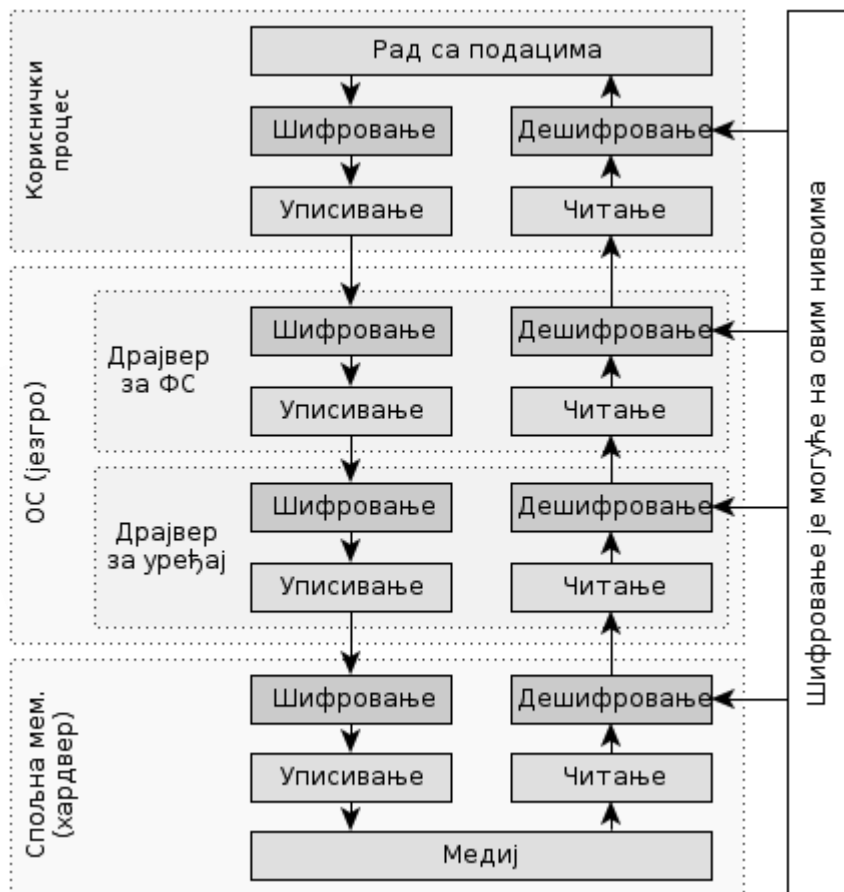
1. за сваку апликацију се мора појединачно уграђивати подсистем за шифровање и
2. изворни код многих апликација није доступан да би се могла извршити захтевана модификација - уградња подсистема за шифровање.

Због наведених недостатака функција шифровања и дешифровања података се измешта у језгро оперативног система, најчешће у модул (драјвер) за рад са одређеним системом фајлова. На тај начин се омогућава шифровање свих података који се смештају на спољну меморију, без обзира на то из које апликације долазе.

<sup>2</sup> <http://www.reuters.com/article/2011/09/29/us-data-breach-texas-idUSTRE78S5JG20110929>

<sup>3</sup> [http://www.consumeraffairs.com/news04/2006/05/va\\_laptop.html](http://www.consumeraffairs.com/news04/2006/05/va_laptop.html)

Алтернатива коришћењу шифрованих фајл система је шифровање података приликом самог уписа на спољну меморију. То се може извести стављањем шифарских функција у драјвер уређаја за спољну меморију или у сам уређај. Предност измештања шифарских функција у сам уређај спољне меморије је растеређивање централног процесора рачунара.



Слика 3.3-1- Могући нивои шифровања података

Треба имати у виду да шифровање фајл-система не пружа никакву додатну заштиту код активних рачунарских система, односно када се рачунарском систему код кога је шифровани фајл-систем монтиран приступа кроз мрежни или кориснички интерфејс. Додатна заштита постоји само у случајевима када нападач физички преузме уређај са шифрованим фајл-системом и покуша да прочита податке монтирањем фајл-система.



## 4. Безбедност приватних рачунарских мрежа

Приватне рачунарске мреже поседују другачије безбедносне карактеристике од јавних мрежа (Интернета), пре свега из тог разлога што се најчешће налазе у власништву једне особе или организације. Под тим се подразумева власништво над рачунарима, мрежним уређајима, комуникационим каналима и свим осталим што је у директној вези са рачунарском мрежом. На основу тога се приватне рачунарске мреже углавном сматрају безбедним, односно сматра се да нема потребе за инсталирањем додатних система заштите.

У комуникационим каналима приватних рачунарских мрежа, као и у меморији рачунара у њој, често се налазе веома важни и безбедносно осетљиви подаци. На пример, запослени у већим пословним организацијама често званично комуницирају искључиво путем електронске поште, којом и прослеђују једни другима поверљиве пословне документе. Пратећи пут једног таквог слања поверљивог документа може се закључити да се он налази:

1. на екстерној меморији рачунара пошиљаоца,
2. затим у његовој радној меморији,
3. комуникационом каналу између рачунара пошиљаоца и приступног мрежног уређаја,
4. меморији тог уређаја, комуникационим каналима између посредујућих мрежних уређаја,
5. затим радној и екстерној меморији сервера електронске поште, поново комуникационим каналима и мрежним уређајима до примаоца и,
6. коначно, у радној и спољној меморији рачунара примаоца.

Дакле, ово је прилично велики број места на којима се привремено или трајно може налазити један поверљиви документ. При том, **свака** од наведених локација се може нападати са циљем приступа садржају или измене садржаја документа.

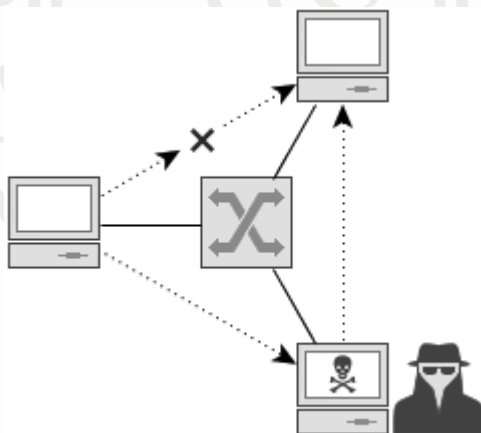
При том, изолованих приватних рачунарских мрежа је све мање, односно данас је готово подразумевано повезивање приватне рачунарске мреже са Интернетом. То значи да нападача од претходно наведених локација са поверљивим документом дели углавном само један рутер са интегрисаним заштитним зидом или функцијом превођења мрежних адреса. Другим речима, у случају било каквог продора у приватну мрежу претпоставка њене сигурности представља за нападача огромно олакшање при извођењу напада.

## 4.1. Напади у Етернет мрежама

Већина приватних рачунарских мрежа је заснована на Етернет технологији. Ова технологија је у употреби преко 40 година и током свог постојања је доживела више значајних еволутивних скокова, како по питању скока перформанси, тако и по питању увођења нових функција. Међутим, једноставност дизајна и потреба за новим функцијама довеле су до могућности за извршавање напада, како на чланове рачунарске мреже, тако и на саму мрежну инфраструктуру. У наставку су дати неки од најчешћих типова напада на рачунарске мреже засноване на Етернет технологији.

### 4.1.1. Тровање ARP кеша

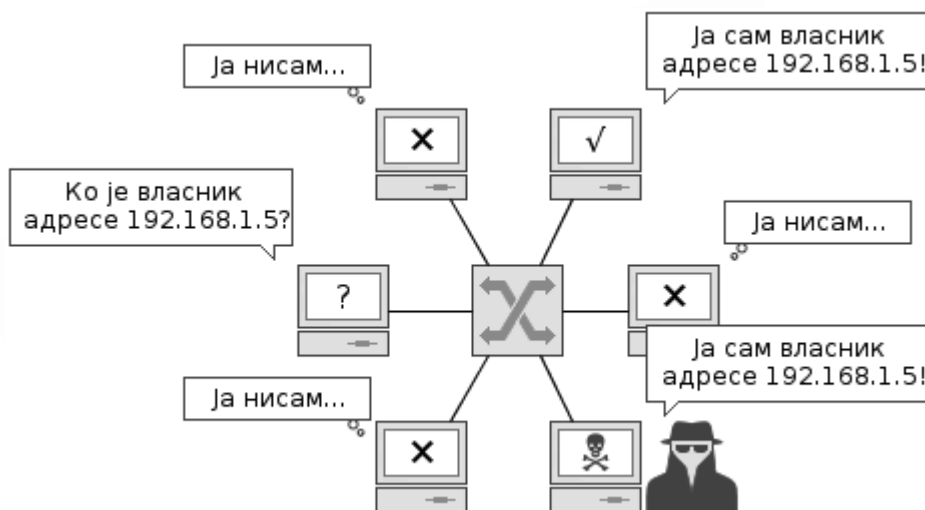
Напад обманивањем *ARP* протокола (енгл. *ARP Spoofing*), који се често среће и под називом **напад тровањем *ARP* кеша** (енгл. *ARP Cache Poisoning*) или само **тровање *ARP*-а** (енгл. *ARP Poisoning*) има за циљ да преусмери изворни правилан ток комуникације уношењем погрешних вредности у *ARP* табеле њених учесника. Нападач овим нападом обично постиже то да се уметне у комуникацију између мрежних чланова и да на тај начин има потпуни увид у поруке које они размењују.



Слика 4.1.1-1 - Резултат успешног извршавања *ARP spoofing* напада

На слици 2 приказан је принцип извршавања овог напада, односно уметања лажних вредности у *ARP* табеле чланова мреже. Рачунар који жели да иницира комуникацију са рачунаром чија је *IP* адреса 192.168.1.5 шаље *ARP* упит, односно питање која је физичка адреса рачунара који има ту *IP* адресу. У

нормалном случају би само један рачунар, прави власник те *IP* адресе, одговорио на такво питање. Међутим, када је напад у питању, рачунар нападача такође потврђује да је он власник те адресе, трудећи се да његов одговор стигне пре него одговор регуларног рачунара. Уколико у томе успе, рачунар који је поставио питање у своју *ARP* табелу ће унети погрешну физичку адресу (адресу нападача) као везу са *IP* адресом 192.168.1.5 и убудуће ће поруке намењене тој адреси слати рачунару нападача. Рачунар нападача те поруке може прослеђивати исправном одредишту, али при том имати увид у њих и евентуално их мењати пре прослеђивања.



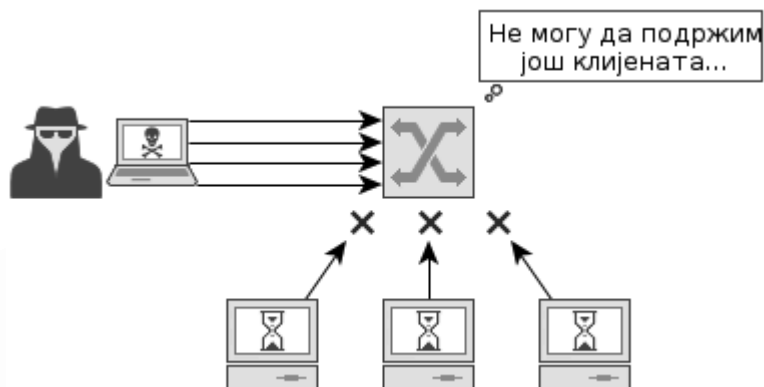
Слика 4.1.1-2 - Принцип извођења *ARP spoofing* напада

Одбрана од *ARP poisoning* напада је прилично захтевна, пре свега због тога што је прво потребно открити да такав напад постоји у мрежи. Са аспекта корисника, током напада све се одвија несметано, уз евентуална, често неприметна, кашњења и благи пад перформанси.

Напреднији комутатори које производи компанија Циско имају функцију тзв. **динамичког *ARP* испитивања** (енгл. *Dynamic ARP Inspection, DAI*) која се заснива на провери валидности веза између физичких и *IP* адреса и која је повезана са подацима добијеним током добијања *IP* адресе од *DHCP* сервера.

### 4.1.2. Препуњавање адресне меморије комутатора

Један од разлога због кога Етернет комутатори функционишу ефикасније од разводника је тај што они „памте“ који клијенти им се налазе на којим портovima. На основу тога они не прослеђују све пакете свим члановима, већ их усмеравају само на адресоване примаоце. Да би знао који клијенти се налазе на којим портovima, комутатор у посебној меморији складишти парове информација типа физичка адреса - порт.



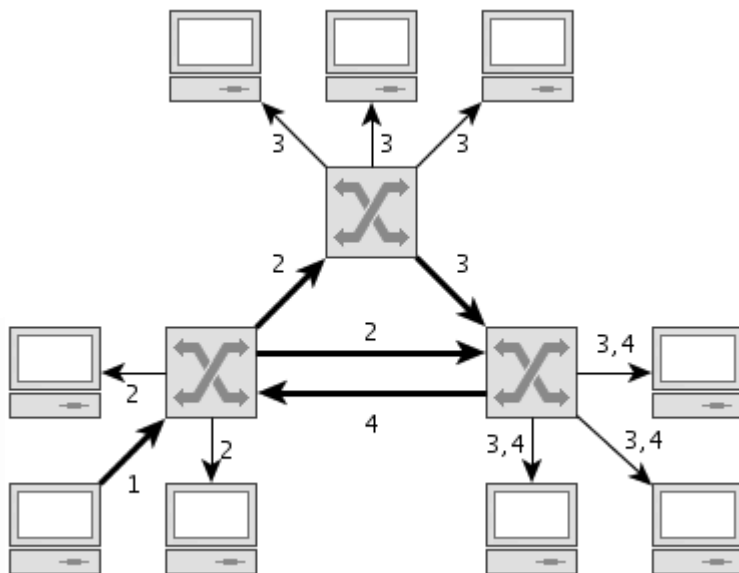
Слика 4.1.2-1 - Блокирање исправних клијената препуњавањем адресне табеле

С обзиром на то да је свака меморија у рачунарским системима ограниченог капацитета, па у складу са тим и меморија комутатора у којој се чувају поменуте везе између адреса и портова, она се може попунити. Након попуњавања ове меморије комутатор више није у могућности да додаје нове записе у њу, односно да прихвата везе нових клијената. Последица тога је да, након препуњавања адресне табеле, комутатор или блокира саобраћај нових клијената, или почиње да се понаша као разводник, односно да примљене пакете прослеђује на све портове.

Извршавање овог типа напада је веома једноставно с обзиром да је једино потребно да нападач комутатору пошаље велики број пакета са различитим изворишним физичким адресама, пошто комутатор адресну табелу формира правећи везе између порта и адресе на основу примљених пакета. Са друге стране, и заштита од оваквог типа није сложена и подржана је на свим озбиљнијим савременим комутаторима. У принципу, заштита се своди на блокирање портова на којима се јави више од једне различите изворишне физичке адресе. Са друге стране, да би везе са другим комутаторима исправно функционисале, потребно је правилно одредити улоге портова на комутатору.

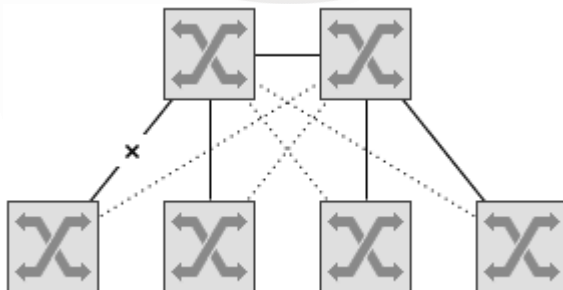
### 4.1.3. Напади на STP протокол

У Етернет рачунарским мрежама постојање цикличних петљи, односно вишеструких путања између комутатора, подразумевано доводи до бесконачног кружења и умножавања пакета. Овакви пакети прво доводе до успореног рада мреже а затим и до њеног потпуног отказивања.



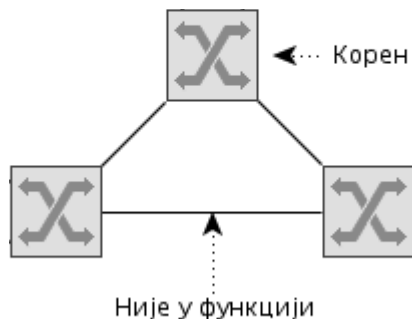
Слика 4.1.3-1 - Кружење и умножавање broadcast пакета у Етернет мрежи

Са друге стране, редундантне везе у Етернет мрежама су потребне у циљу подизања доступности мреже. Хијерархијска мрежна структура прописује постојање најмање две путање између свака два комутатора у мрежи, тако да се у случају отказивања једног комутатора може наставити несметан рад путем другог.



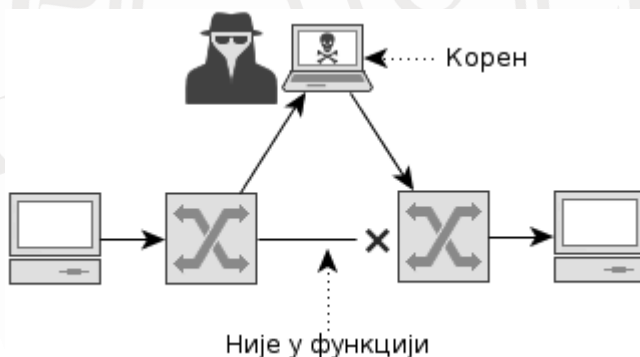
Слика 4.1.3-2 - Редунданса у Етернет мрежама у циљу повећања доступности

У циљу решавања наведеног конфликта развијен је протокол разгранатог стабла (енгл. *Spanning Tree Protocol, STP*). Овај протокол омогућава откривање постојања паралелних веза и привремено искључивање коришћења неких од њих, у циљу избегавања стварања цикличних петљи.



Слика 4.1.3-3 - STP протокол привремено искључује редундантне канале

Применом STP протокола логичка топологија Етернет мреже преводи се у стабло које садржи један корен (корени комутатор) и не садржи вишеструке путање између чворова. Процес избора кореног комутатора и веза које ће се искључити одвија се аутоматски, разменом података између комутатора.

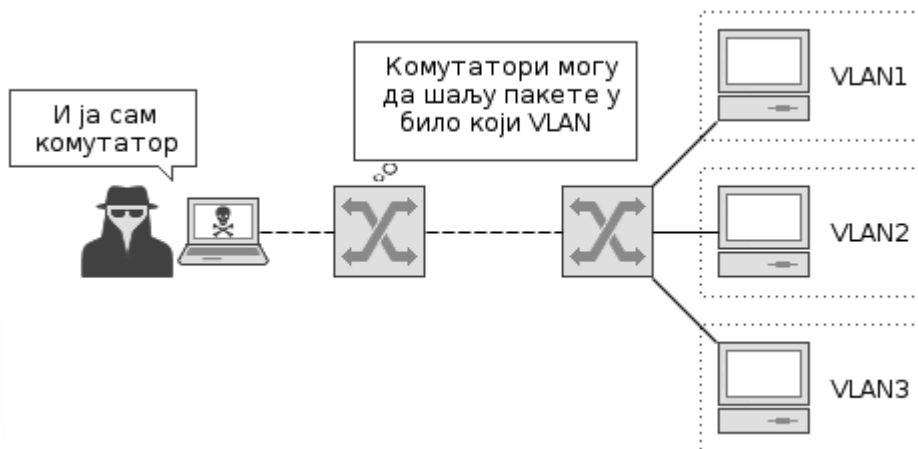


Слика 4.1.3-4 - Резултат напада на STP протокол

Напад на STP протокол врши се управо у процесу „гласања“, односно избора кореног комутатора. У том процесу се нападач представља као комутатор и другим комутаторима шаље параметре који га проглашавају кореном, или барем један део саобраћаја преусмеравају на њега уместо на исправне везе. На тај начин нападач може да надгледа саобраћај који пролази кроз сегмент мреже у коме се он налази. За одбрану од овог типа напада потребно је користити комутаторе код којих се може дефинисати улога порта (кориснички или транк).

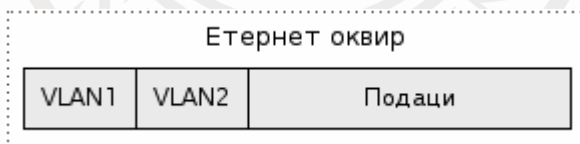
#### 4.1.4. Напад прескакањем у друге виртуалне локалне мреже

Коришћење виртуалних мрежа на локалном подручју омогућава изоловање корисника у целине чија међусобна комуникација није дозвољена. Међутим, постоји и тип напада који нападачу омогућава да „прескочи“ у другу виртуалну мрежу (енгл. *VLAN hopping*). Овај напад се може извршити на два начина и то само у мрежама чији комутатори нису исправно подешени.



Слика 4.1.4-1 - Прескакање путем представљања као комутатор

Први начин за извођење овог напада је лажно представљање нападача као комутатора. На тај начин се обмањује комутатор на који је нападач повезан тако да он, сматрајући да му подаци стижу од другог комутатора, прихвата пакете који су адресовани на чланове било које виртуалне мреже.

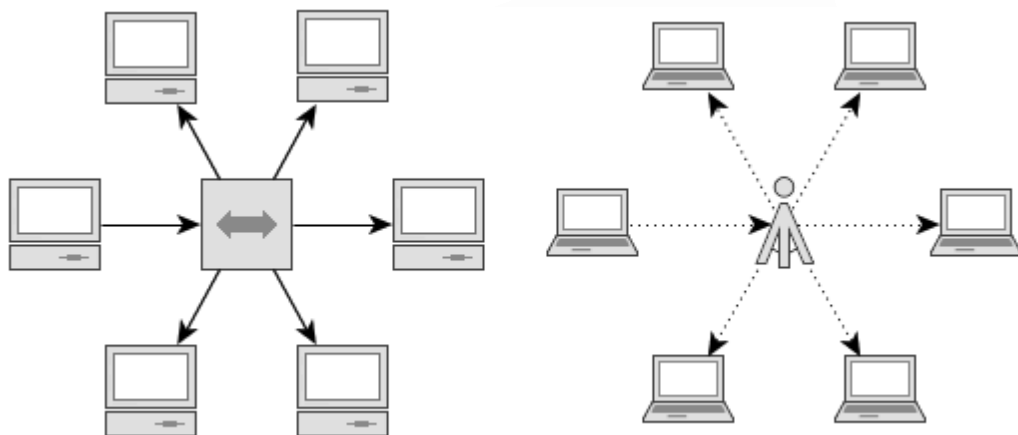


Слика 4.1.4-2 - Напад двоструким навођењем VLAN-а

Други начин за извршавање овог напада је двоструко означавање припадности пакета виртуалној мрежи (енгл. *double tagging*). Код овог напада нападач код пакета који припада природном VLAN-у (енгл. *native VLAN*) уметне и ознаку припадности виртуалној мрежи у коју покушава да пошаље пакет. Комутатор приликом прослеђивања пакета одбацује прву ознаку (идентификатор *native VLAN*-а) и прослеђује пакет даље. Међутим, следећи комутатор при пријему пакета види другу ознаку те сматра да пакет припада виртуалној мрежи коју та ознака садржи.

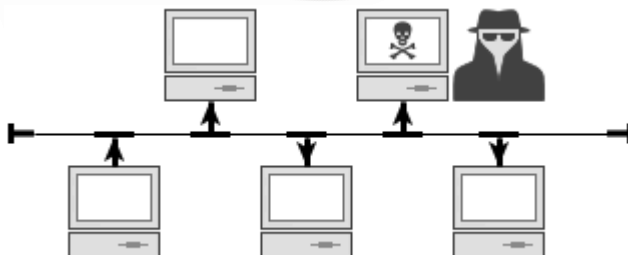
## 4.2. Безбедност у бежичним рачунарским мрежама

Бежичне приватне рачунарске мреже најчешће су засноване на 802.11 групи спецификација. Ове спецификације дефинишу физички слој и слој везе података *OSI* комуникационог модела. Са безбедносног аспекта 802.11 бежичне мреже се могу поредити са првим генерацијама Етернет мрежа (слика 1), односно са Етернет мрежама заснованим на коаксијалним кабловима или на разводницима (енгл. *hub*). Код таквих Етернет мрежа разводници су примљене пакете прослеђивали свим члановима мреже, који су их, затим, обрађивали или одбацивали, у зависности од тога да ли су пакети били адресовани на њих или не.



Слика 4.2-1 - Логичка еквиваленција  
Етернет разводника и 802.11 приступне тачке

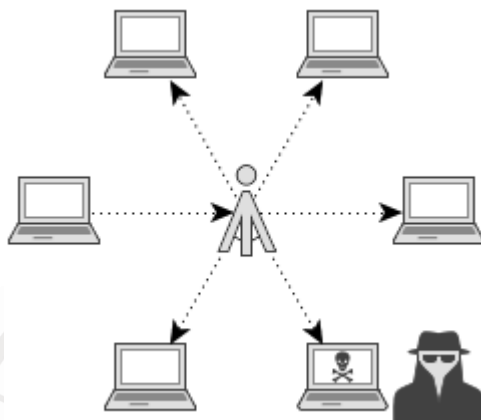
Овакав приступ је проблематичан, пре свега гледано са безбедносног, односно приватносног аспекта. Наиме, у таквом сценарију је нападачу било довољно да постане члан мреже да би могао да надгледа саобраћај свих комуникација.



Слика 4.2-2 - Прислушкивање у раним Етернет мрежама



Оно што бежичне мреже чини безбедносно још осетљивијим је могућност нападача да лако приступи мрежи. Наиме, док је код Етернет технологије за приступ мрежи потребно пронаћи слободну утичницу комутатора или разводника, код бежичних мрежа је довољно да се нападач нађе у подручју које покрива бежична приступна тачка.



Слика 4.2-3 - Прислушкивање у незаштићеним бежичним мрежама је лако

Постоји више приступа којима се бежичне приватне рачунарске мреже штите и нападају. У основи се заштита бежичних рачунарских мрежа заснива на контроли приступа (ко има право да постане члан мреже) и шифровању саобраћаја.

#### 4.2.1. Контрола приступа и прислушкивање

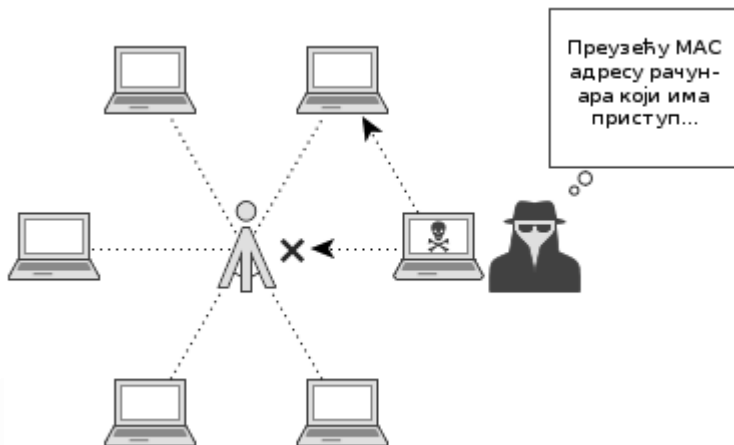
Један од често коришћених начина за спречавање приступа бежичној мрежи је искључивање емитовања њеног назива (енгл. *Service Set Identification, SSID*). Тиме се онемогућава проналажење мреже путем стандардних алата за управљање мрежним везама. Ово, међутим, представља веома низак ниво заштите јер иоле озбиљнији нападачи могу да, коришћењем алата као што су *Kismet*<sup>4</sup> или *inSSIDer*<sup>5</sup>, веома једноставно открију на овај начин сакривене бежичне мреже.

Следећи начин за спречавање приступа приватној бежичној мрежи је филтрирање физичких адреса, односно омогућавање повезивања само клијентима чије су физичке адресе унете у базу бежичне приступне тачке.

4 <http://www.kismetwireless.net>

5 <http://www.inssider.com>

Међутим, ни ово не представља озбиљну заштиту бежичне мреже из тог разлога што нападач може веома лако открити која је физичка адреса неког од клијената којима је приступ бежичној мрежи дозвољен, а затим свом рачунару доделити ту адресу. На тај начин је и ова безбедносна мера заобиђена.



Слика 4.2.1-1 - Заобилажење заштите филтрирањем физичких адреса

Најозбиљније решење за контролу приступа је коришћење друге верзије заштићеног бежичног приступа (енгл. *WiFi Protected Access, WPA*), који се може користити у два режима:

1. у комбинацији са *802.1X* протоколом и
2. са коришћењем раније размењеног кључа.

Заштићени бежични приступ у комбинацији са *802.1X* протоколом је безбеднија варијанта и заснива се на коришћењу дигиталних сертификата и *RADIUS* сервера за одобравање приступа мрежи. Често се назива и *WPA* за предузећа (енгл. *WPA-Enterprise*). Предност овог решења је отпорност на разбијање лозинки, што је евентуално могуће код *WPA* у комбинацији са раније размењеним кључем. Међутим, основна мана овог решења је сложено постављање система и управљање базом корисника којима је дозвољен приступ.

Заштићени бежични приступ заснован на раније размењеном кључу подразумева дефинисање приступне лозинке (кључа) на бежичним приступним тачкама и његово дистрибуирање корисницима безбедним каналом (на пример, вербална размена). Овај кључ може бити у хексадецималном или *ASCII* облику и његова дужина је од 8 до 64 бајта.

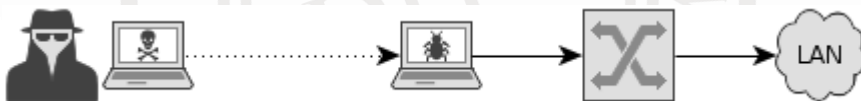
Претходници друге верзије заштићеног бежичног приступа били су његова прва

верзија, као и протокол за приватност једнаку приватности у жичним мрежама (енгл. *Wired Equivalent Privacy*, *WEP*). Као што му и име говори, овај протокол је развијен са циљем да пружи исти ниво безбедности у бежичним мрежама као у мрежама заснованим на Етернет технологији. Међутим, показало се да је ниво безбедности коју овај протокол нуди веома низак, односно да је веома лако разбити заштиту коју он пружа. Из тог разлога се његова употреба више не препоручује.

Осим што нуде контролу приступа бежичној мрежи, протоколи *WEP/WPA/WPA2* подразумевају и шифровање саобраћаја, тако да је њиховом употребом знатно отежано прислушкивање. *WEP* протокол користи RC4 алгоритам за шифровање са кључевима дужине од 40 до 128 бита. Прва верзија *WPA* протокола такође користи RC4 алгоритам са кључевима дужине 128 бита, док се код друге верзије користи AES алгоритам са истом дужином кључева.

#### 4.2.2. Проблем вишеструко умрежених рачунара

Један од великих безбедносних проблема код свих приватних рачунарских мрежа чине вишеструко умрежени рачунари. Под вишеструко умреженим рачунарима се сматрају рачунари који поседују два или више мрежна интерфејса којима су повезани са различитим рачунарским мрежама.



Слика 4.2.2-1 - Упад у приватну Етернет мрежу преко члана са WiFi везом

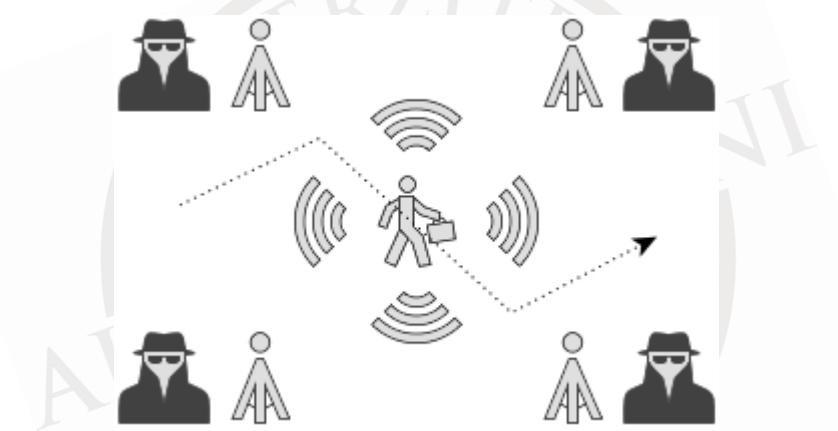
Пример безбедносног проблема који може да изазове вишеструко умрежени рачунар је преносиви рачунар који је путем Етернет технологије повезан са приватном рачунарском мрежом, док је на њему активан и 802.11 порт путем кога је повезан са рачунаром нападача (на пример, као последица заражавања тројанцем). У таквој конфигурацији нападач из даљине (и до пар стотина метара) има могућност да приступа приватној рачунарској мрежи у улози исправног корисника.

Бежичне везе су код вишеструко умрежених рачунара посебно проблематичне јер је веома тешко открити њихово присуство. За одбрану од оваквих напада се код професионалних система користе ометачи радио-веза.

### 4.3. Безбедност Bluetooth мрежа

Технологија под назвиом *Bluetooth* је данас подразумевана технологија за формирање личних рачунарских мрежа (енгл. *Personal Area Network, PAN*), односно мрежа које повезују личне уређаје одређеног корисника (мобилне телефоне, *hands-free* сетове, преносиве рачунаре, тастатуре, мишеве и слично). Треба имати у виду да, поред евентуалног разбијања заштите и читања података, нападачи имају и друге могућности за напад коришћењем ове технологије.

Метод под називом *Bluetoothing* користи се за „узимање отиска прста“ уређаја који има подршку за *Bluetooth* умрежавање, односно за његову идентификацију путем физичке адресе. На основу ове идентификације нападачи могу да прате кретање власника уређаја.



Слика 4.3-1 - Праћење кретања корисника путем *Bluetooth* уређаја

Такође, треба имати у виду и могућност ускраћивања коришћења уређаја и сервиса који функционишу преко *Bluetooth* комуникационих канала. Овакав тип напада заснива се на ограниченој пропусној моћи таквих канала, а нападач код њих затрпава уређај великом количином саобраћаја, онемогућавајући га на тај начин да комуницира са осталим уређајима.

Америчка државна безбедносна агенција дала је својим корисницима следеће препоруке<sup>6</sup> код коришћења *Bluetooth* технологије са циљем подизања нивоа безбедности:

- Никад не користити стандардне комерцијалне *Bluetooth* хедсетове.

6 Bluetooth Security, Systems and Network Analysis Center Information Assurance Directorate, National Security Agency

- *Bluetooth* функционалност укључивати само када је неопходно.
- Користити само уређаје са *Bluetooth* примопредајницима ниске снаге, 2. или 3. класе.
- Уређаје који користе *Bluetooth* држати физички што ближе једне другима током коришћења.
- Независно надгледати уређаје и везе због неауторизованих активности.
- Омогућити видљивост уређаја приликом претраге само уколико је потпуно неопходно.
- Дозволити уређајима да се други уређаји повезују на њих само уколико је потпуно неопходно и само док има потребе за везом.
- Упаривати *Bluetooth* уређаје само у безбедним зонама и уз коришћење дугих, случајно генерисаних кључева. Никада не уносити кључеве уколико се неочекивано појави захтев за уносом.
- Одржавати физичку контролу уређаја све време. Уређаје који су украдени или изгубљени уклонити са листе упаривих уређаја.
- Користити филтере пакета, редовно инсталирати нове верзије софтвера и одржавати анти-вирусни софтвер ажурним.

Такође, када су програмери у питању исти документ садржи следеће препоруке:

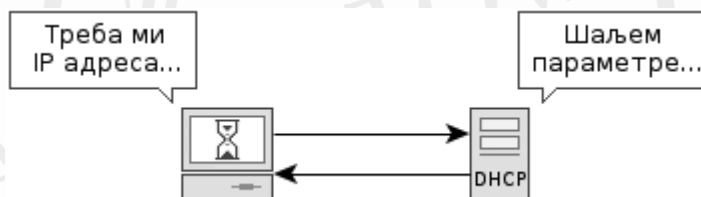
- Искључити подршку за хедсетове и хендс-фри профиле уколико везе нису адекватно заштићене применом одговарајућих техника.
- Кључеви морају бити дужине од најмање 8 знакова а период важења мора бити ограничен.
- Користити индикаторе за промену конфигурације и активност линкова, као што су *LED* диоде или десктоп иконице.
- Користити описе *Bluetooth* уређаја који нису информативни и идентификовати све упарене и повезане уређаје преко физичке адресе.
- Захтевати ауторизацију од стране корисника за све долазеће захтеве за повезивањем и не прихватати повезивање, фајлове и друге објекте од непроверених извора.
- Програмирати сваки уређај да иницијализује *Bluetooth* аутентификацију одмах по успостављању везе (познато и као *Security Mode 3, Link Level Security*).
- Програмирати сваки уређај да укључи 128-битно шифровање одмах након међусобне аутентификације.

- Искључити могућност да корисник контролише *Bluetooth* подешавања која би потенцијално могла да угрозе безбедност.
- Укључивати *Bluetooth* сервисе само када је неопходно. Трајно искључити или уклонити све *Bluetooth* сервисе који нису неопходни.
- Дигитално потписивати сав фирмвер, драјвере и апликативни софтвер. Потврдити да софтвер који није ауторизован не може да користи *Bluetooth API*.

Наведене препоруке јасно илуструју колико је пажње потребно за безбедно коришћење *Bluetooth* технологије, односно колико опасно може бити коришћење без свести о безбедносном аспектима.

#### 4.4. Напади на DHCP сервис

Сервис за динамичко подешавање чланова мреже (енгл. *Dynamic Host Configuration Protocol*) је инфраструктурни сервис којим администратори могу аутоматизовано да додељују мрежне параметре (адресу, маску, подразумевани мрежни пролаз, DNS сервере и слично). Овај сервис је незаменљив у мрежама са великим бројем чланова који се често мењају



Слика 4.4-1 - Принцип рада DHCP сервиса

Постоји неколико типичних напада на *DHCP* сервис, а њима се угрожавају и поверљивост, и веродостојност, и доступност мрежних телекомуникација. Оно што отежава заштиту *DHCP* сервиса је то да:

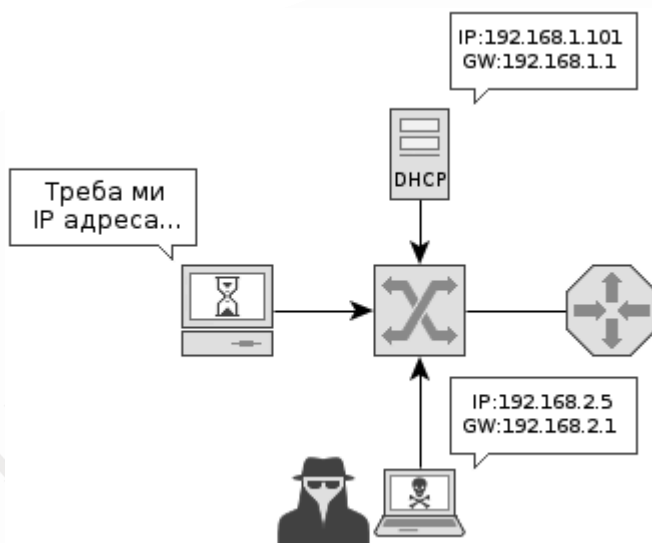
- комуникација са *DHCP* сервером је најчешће прва комуникација коју нови чланови мреже остварују унутар ње,
- комуникација са *DHCP* сервером се одвија „у позадини“, односно нема експлицитних показатеља да се користи неки мрежни сервис,
- корисници веома ретко имају и знања и интересовања да проверавају податке добијене од *DHCP* сервера и
- напади на *DHCP* сервис, посебно када су у питању напади на поверљивост и веродостојност, најчешће не блокирају рад осталих

сервиса, тако да се са аспекта корисника све одвија нормално.

У наставку су дати основни типови напада на *DHCP* сервис.

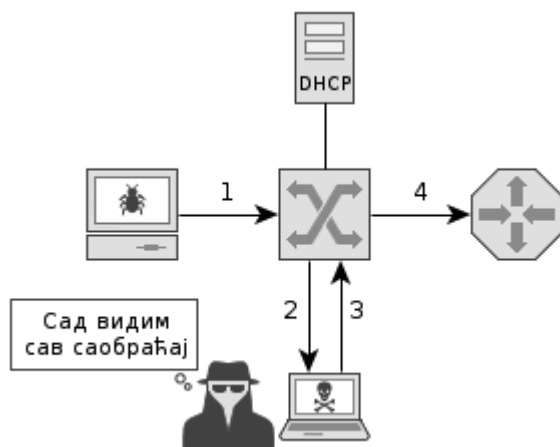
#### 4.4.1. Лажни DHCP сервер

Постављање лажног *DHCP* сервера је обично први корак у извршавању напада типа „човек у средини“ на чланове приватне рачунарске мреже. Код овог напада нападач са мрежом повезује свој рачунар на коме су активиране функције *DHCP* сервера и рутирања. Након тога, рачунар нападача одговара на *DHCP* упите нових чланова мреже са циљем да на њих одговори **пре** правог *DHCP* сервера и да им пошаље погрешне мрежне параметре.



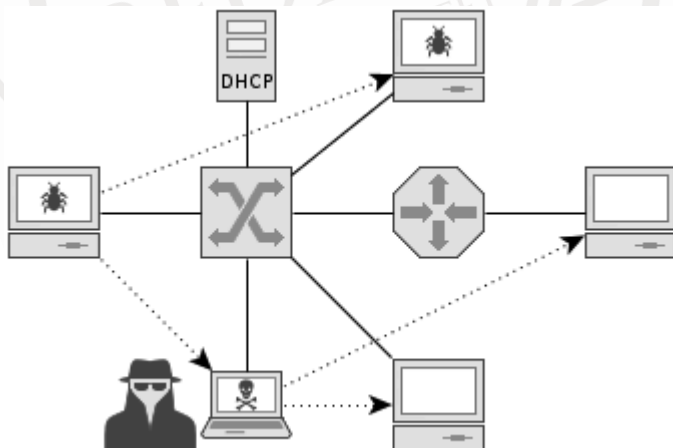
Слика 4.4.1-1 - Нападачев рачунар шаље погрешне параметре клијентима

Првенствена сврха слања погрешних мрежних параметара је то да се рачунари којима су они достављени адресују из посебног мрежног опсега (на пример, из опсега 192.168.2.0/24 уместо из опсега 192.168.1.0/24 који се користи у мрежи), као и да им се рачунар нападача постави као подразумевани мрежни пролаз. На тај начин нападач има увид у саобраћај између нападнутих рачунара и остатка мреже.



Слика 4.4.1-2 - Нападач има увид у комуникације погрешно подешених рачунара

Треба имати у виду и да се основном, претходно описаном, поставком напада нападачу омогућава увид само у садржај комуникација између чланова приватне мреже и других мрежа, као и у саобраћај између чланова који су погрешно подешени и оних који то нису. Међутим, комуникација између два погрешно подешена рачунара ће се у тој поставци одвијати директно, пошто су адресовани из истог мрежног опсега (192.168.2.0/24), односно сматрају да су у истој мрежи.



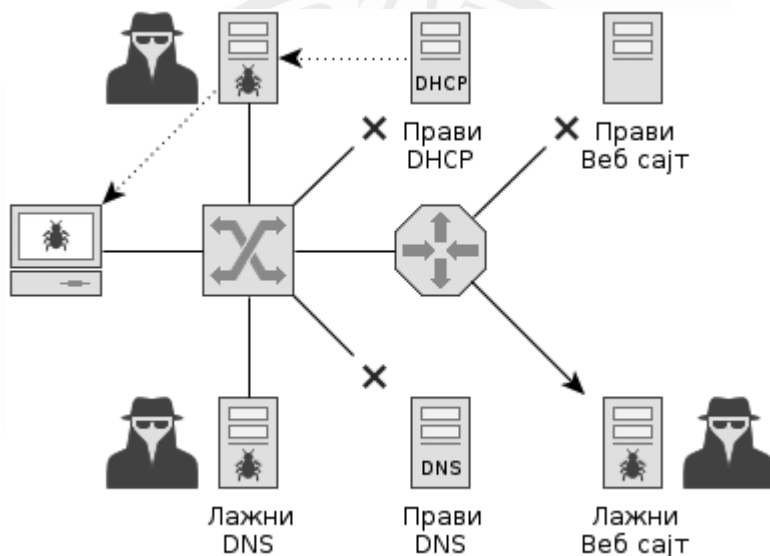
Слика 4.4.1-3 - Директне и посредне комуникације код овог типа напада

Да би се ово превазишло, нападач може сваком рачунару давати посебан мрежни опсег (на пример, 192.168.2.0/30, 192.168.2.4/30, итд) чији су једини чланови тај рачунар и рачунар нападача у улози подразумеваног мрежног



пролаза. На тај начин ће се све мрежне комуникације одвијати посредством рачунара нападача. Међутим, таква поставка може довести до престанка рада одређених мрежних сервиса који су подешени само за рад у једној приватној рачунарској мрежи.

Треба имати у виду и то да је овакав тип напада тешко открити пошто код правилно изведеног напада сви сервиси настављају нормално да раде, односно корисници немају индикаторе да се било шта променило у мрежи. Међутим, оно што може довести до сумње корисника јесте евентуалан пад перформанси уколико рачунар нападача постане „уско грло“ у комуникацији, што се може десити уколико се превелики број рачунара преусмери на њега. Да би се то спречило могуће је ограничити скуп рачунара који се нападају филтрирањем *DHCP* захтева на које се одговара путем физичких адреса.



Слика 4.4.1-4 - Замена DNS сервера у DHCP одговору

Посебну варијанту овог напада чини напад на веродостојност путем подметања лажног *DNS* сервера. Код овог типа напада нападач од *DHCP* сервера изнајмљује одређену *IP* адресу коју касније прослеђује клијенту, у неизмењеном облику, као и подразумевани мрежни пролаз. Међутим, оно што се мења, односно лажно шаље клијенту је адреса *DNS* сервера. На тај начин клијент наставља да нормално комуницира са члановима своје и других мрежа, али за разрешавање мрежних имена користи нападачев *DNS* сервер. Овакав напад је обично увертира у обмањивање корисника подметањем лажног Веб сајта, са циљем крађе корисничких података (корисничког имена и лозинке, података кредитне

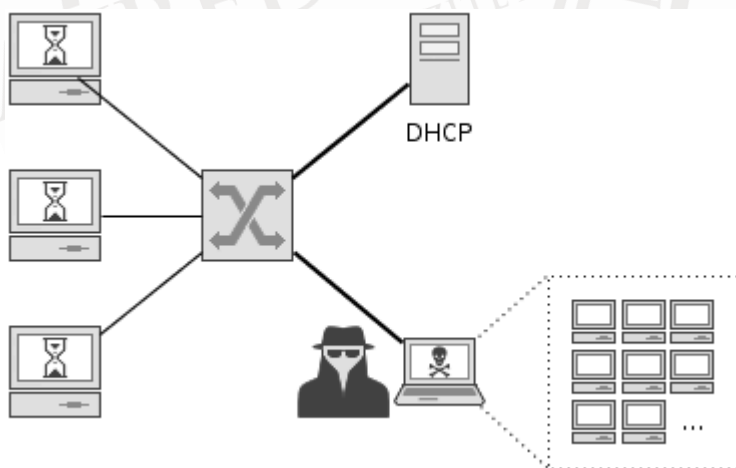
картице и слично).

Дакле, напад путем лажног *DHCP* сервера је релативно једноставан напад којим се могу угрозити поверљивост и веродостојност код комуникација у којима учествују чланови приватне рачунарске мреже. Сама доступност обично није сврха овог напада јер се нападом на њу алармирају мрежни администратори који би даљом истрагом открили присуство нападача у приватној рачунарској мрежи.

Једна од техника за одбрану од оваквих типова напада је *DHCP Snooping* која је доступна на Циско Каталист комутаторима. Овом техником се портови на комутаторима деле на оне у које се има поверења (енгл. *trusted*) и оне са којима то није случај (*untrusted*). На тај начин се само портovima у које се има поверења дозвољава да шаљу *DHCP* одговоре, док је портovima у које се нема поверења дозвољено само да шаљу *DHCP* захтеве за добијање адресе.

#### 4.4.2. Исцрпљивање адреса

Један од популарних напада на *DHCP* сервис је напад са циљем исцрпљивања адреса (енгл. *starvation attack*) којим се напада доступност сервиса у приватној рачунарској мрежи. Суштина овог напада је заузимање свих адреса приватне рачунарске мреже од стране нападача, тако да се рачунари регуларних корисника не могу адресовати.



Слика 4.4.2-1 - Напад исцрпљивања адреса на *DHCP* серверу

За потребе овог напада нападач може користи једноставан програм (скрипт) чија је логика илустрована следећим псеудо-кодом:

док год DHCP сервер враћа IP адресу:

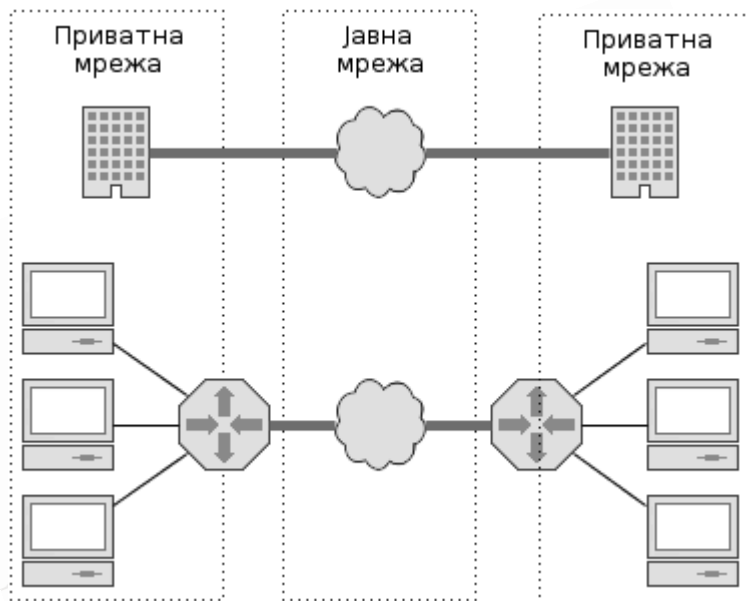
```
{  
    преузми адресу од DHCP сервера  
    повећај MAC адресу мрежног интерфејса за 1  
}
```

Оно што треба имати у виду је да се овим нападом онемогућавају будући клијенти *DHCP* сервера, односно рачунари који ће покушати да од *DHCP* сервера добију мрежне параметре након извршеног напада. Из тог разлога се овакви напади у пословним организацијама често извршавају неколико минута пре почетка радног времена, односно пре него што крене укључивање клијентских рачунара.

Одбрана од овог типа напада је готово немогућа на самом *DHCP* серверу, пошто са његовог аспекта захтеви за адресама стижу од различитих рачунара (са различитих физичких адреса). Међутим, уређај који има могућност да открије овакав тип напада је мрежни комутатор. Мрежни комутатор има могућност да примети да са једног клијентског порта стиже више порука са различитим физичким адресама, као и да тај порт искључи. На пример, комутатори које производи компанија Циско имају тзв. *Port Security* функционалност која аутоматски искључује клијентски порт у случају појављивања више различитих физичких адреса на њему.

## 5. Заштита међумрежних комуникација

Заштита комплетних рачунарских мрежа, као што је на пример заштита приватне Етернет мреже употребом 802.1q и 802.1x технологија, подразумева власништво над читавом мрежном инфраструктуром. Са друге стране, заштита појединачних сервиса, на пример Веб сервиса, заснива се на *end-to-end* заштити на нивоу ентитета који комуницирају. Посебан изазов у заштити рачунарских телекомуникација представља заштита међумрежних комуникација.

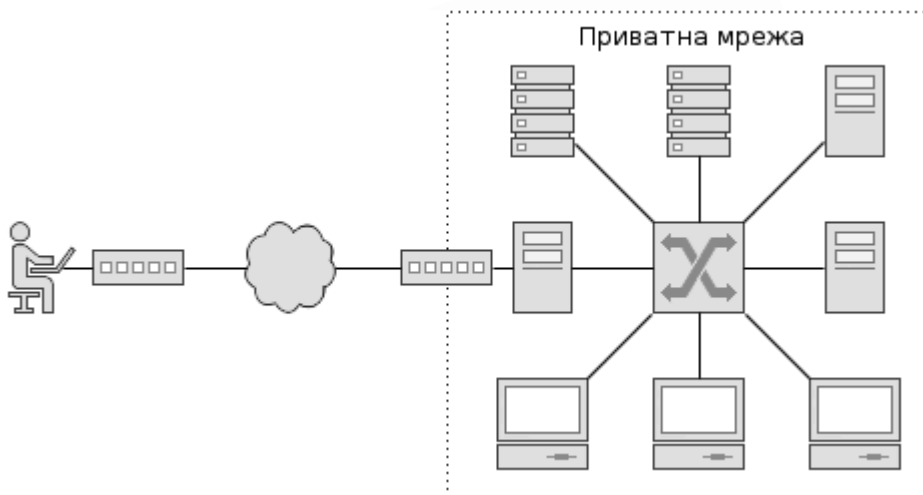


Слика 5-1 - Комуникација између две удаљене приватне мреже

Под заштитом међумрежних комуникација подразумева се безбедно повезивање две несуседне рачунарске мреже преко небезбедних комуникационих канала који припадају једној или већем броју мрежа у туђем власништву. При том је за кориснике овако повезаних мрежа потребно обезбедити исти ниво услуге као и код приватних мрежа, како у функционалном тако и у безбедносном смислу. Другим речима, потребно је обезбедити исте или приближне перформансе, поверљивост, веродостојност и доступност на вези између приватних мрежа као што је то случај код канала локалних приватних мрежа.

Када је у питању величина мрежа које се повезују, оне могу бити исте или различите величине. Могуће је повезивање приватних мрежа које садрже по

више хиљада чланова, али и повезивање појединачних корисника са мрежама од свега пар чланова. Чест пример повезивања великих приватних мрежа је повезивање удаљених локација истог пословног система. Са друге стране, када је у питању повезивање појединачних или малобројних спољних корисника са приватном мрежом (тзв. удаљени приступ мрежи), то могу бити филијале или банкомати и централна мрежа банке, запослени који раде од куће, теренски радници, фирме ангажоване на пословима пружања одређених услуга и друго. Другим речима, једну приватну мрежу је у пракси могуће повезати са великим приватним мрежама које садржа стотине или хиљаде рачунара, малим мрежама од пар рачунара, удаљеним корисницима који користе стоне или преносиве рачунаре, или чак мобилне телефоне.

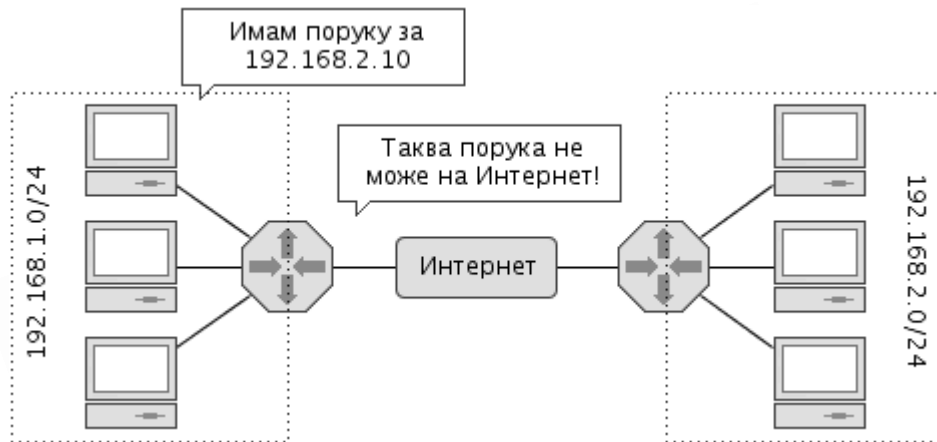


Слика 5-2 - Удаљен приступ приватној мрежи

Раније се за повезивање удаљених приватних мрежа користило више различитих технологија и инфраструктура - фрејм-релеј мрежа, изнајмљене линије и друго. Данас се најчешће користи Интернет мрежа, како због своје глобалне доступности, тако и због ниже цене у односу на алтернативе. Са друге стране, канали Интернет мреже се не сматрају безбедним а само повезивање са овом мрежом теоретски омогућава нападе од стране више милијарди уређаја везаних на ту мрежу. Из тог разлога је потребно посебну пажњу обратити на безбедносне проблеме који могу проистећи из повезивања са Интернет мрежом и коришћења њених канала. Међутим, данас постоји више проверених приступа и решења за безбедно коришћење канала Интернет мреже за потребе повезивања удаљених приватних мрежа и приступа њима од стране удаљених корисника.

## 5.1. Повезивање удаљених приватних мрежа

Повезивање удаљених приватних мрежа, односно омогућавање комуникације између њихових чланова представља, како безбедносни, тако и функционални изазов. Приватне мреже се адресују посебним адресним опсезима Интернет протокола - 10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16 - који се не могу користити на Интернет мрежи. Другим речима, није могуће рутирати пакете адресоване приватним адресама кроз Интернет мрежу (слика 1).

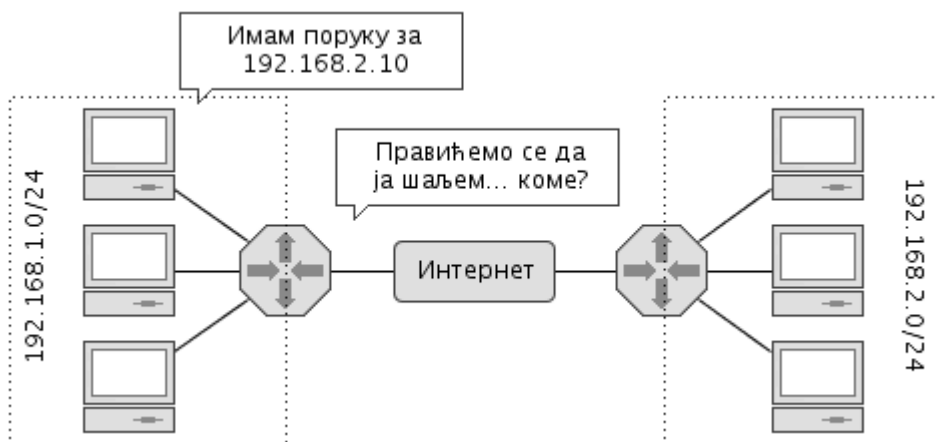


Слика 5.1-1 - Приватне адресе се не могу користити на Интернету

Делимично решење се на први поглед налази у превођењу мрежних адреса (енгл. *Source Network Address Translation, SNAT*), односно у замени изворишне и одредишне адресе јавним адресама рутера који приватне мреже повезују са Интернет мрежом. Међутим, проблем који се у том случају јавља јесте немогућност прецизног одређивања примаоца у одредишној приватној мрежи (слика 2). Да би се овај проблем решио могућа су два решења:

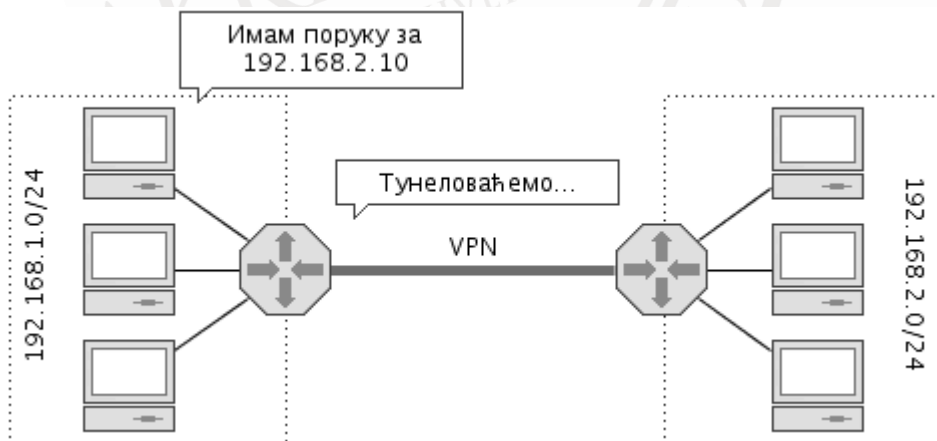
1. закуп по једне јавне адресе за сваког члана приватне мреже и
2. одређивање прималаца путем прослеђивања портова.

Недостатак првог решења је управо потреба да се закупи велики број јавних адреса на Интернет мрежи, што није јефтино а често ни могуће решење. Додатно, таквим решењем се јавља и потреба за ручним подешавањем прослеђивања и ажурирањем подешавања код сваке промене у приватној мрежи.



Слика 5.1-2 - Превођење мрежних адреса и проблем адресовања примаоца

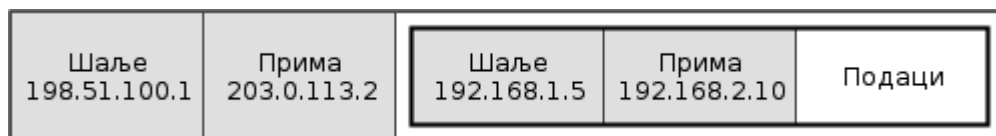
Друго решење, коришћење прослеђивања портова (енгл. *Port Forwarding*) не захтева закуп већег броја јавних Интернет адреса (мада се и за тим може јавити потреба у случају великог броја чланова у мрежи) али захтева да се ручно мапирају сви сервиси (портови) који се користе у приватној мрежи. Такав приступ, осим што представља огроман терет за администраторе мреже, често онемогућава или знатно отежава употребу сложенијих мрежних сервиса. На пример, у таквом сценарију није могуће користити сервисе који динамички отварају различите портове, као ни директно видети са које адресе из удаљене приватне мреже стиже одређени захтев.



Слика 5.1-3 - Тунеловањем се омогућава комуникација између удаљених мрежа

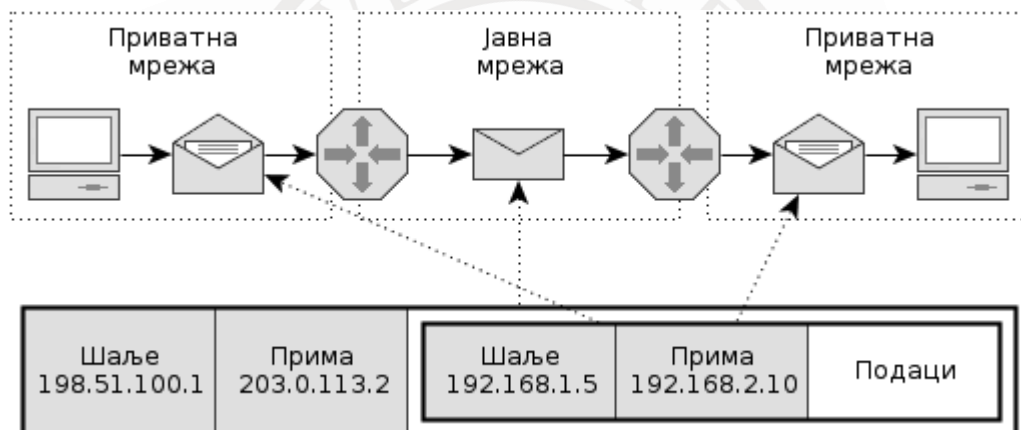
Еlegantно решење за претходно наведене проблеме је такозвано **тунеловање**. Тунеловање је приступ којим се омогућава комуникација између две приватне

мреже путем посредујуће јавне мреже, или мреже засноване на технологији која се разликује у односу на ону која се користи у приватним мрежама. Осим што омогућава повезивање удаљених приватних мрежа, тунеловање нуди и могућност заштите комуникација путем шифровања. Основна идеја код тунеловања је уметање (инкапсулирање) пакета у нове пакете чији је задатак да изворне пакете пренесу дуж одређене деонице пута (слика 4).



Слика 5.1-4 - Тунеловање путем инкапсулације IPv4 пакета

На слици 5 је приказан модел комуникације између чланова две удаљене приватне мреже коришћењем тунеловања. Илустрација се заснива на 4. верзији Интернет протокола али је концепт суштински исти и за остале протоколе.



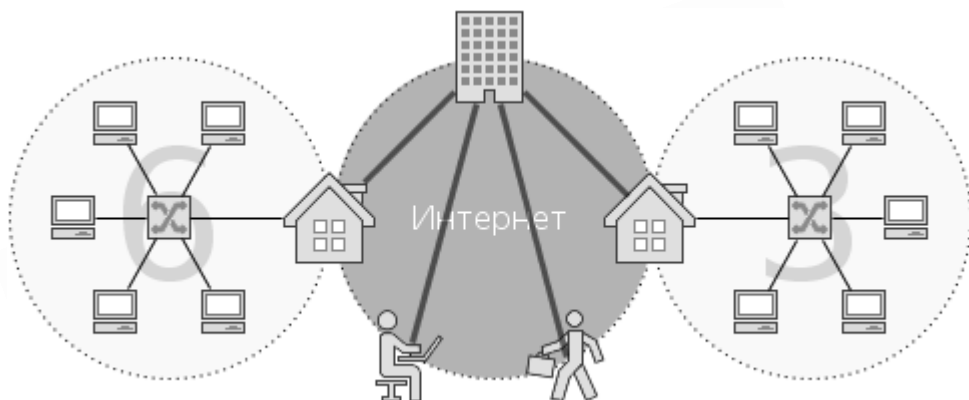
Слика 5.1-5 - Путање изворних и носећих пакета

Пошиљалац пакета, који се налази у првој приватној мрежи, шаље пакет до локалног рутера који га повезује са јавном мрежом. Рутер, након пријема пакета, закључује да је одредиште у мрежи ка којој постоји тунел, те формира нов пакет који адресује на рутер који одредишну приватну мрежу повезује са јавном мрежом. Садржај тог пакета чини комплетан пакет адресован на приватну мрежу. Након пријема пакета, одредишни рутер из његовог садржаја преузима изворни пакет и прослеђује га рачунару у сопственој приватној мрежи.



## 5.2. Виртуалне приватне мреже

Виртуалне приватне мреже (енгл. *Virtual Private Network, VPN*) су основни модел безбедног повезивања удаљених приватних мрежа коришћењем јавних, небезбедних комуникационих канала. Могу се користити за повезивање приватних мрежа са удаљеним мрежама или појединачним корисницима. Често се користе за повезивање удаљених локација истог пословног система, остваривање веза са пословним партнерима, омогућавање запосленима да са терена или од куће приступе ресурсима унутар компаније и томе слично.



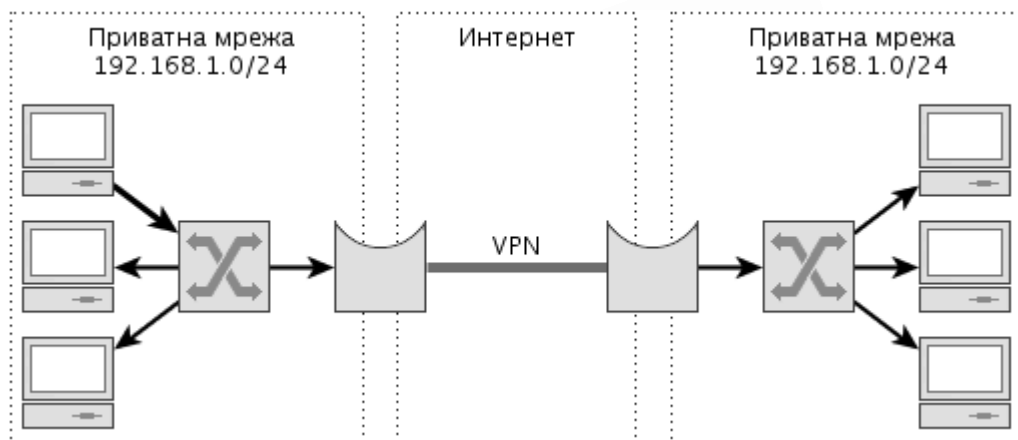
Слика 5.2-1 - Виртуалне приватне мреже и удаљени приступ

Виртуалне мреже омогућавају поверљивост и веродостојност (целовитост и аутентичност) података који се њима преносе, затим контролу приступа и одбрану од напада поновљеним слањем истих података. Ниво заштите коју виртуалне приватне мреже нуде зависи од укључених функционалности, изабраних шифарских алгоритама, конфигурационих параметара и спољних параметара система у којима се врши *VPN* терминација.

Постоји више техничких решења и велики број готових производа за прављење виртуалних приватних мрежа. Рутери већих произвођача мрежне опреме (Циско, Џунипер и други) поседују већ уграђене функције за лако креирање виртуалних приватних мрежа. Линукс оперативни систем је у своје језгро интегрисао подршку за *IPsec* систем заштите који омогућава креирање виртуалних приватних мрежа. Новији серверски оперативни системи компаније Мајкрософт такође имају подршку за прављење виртуалних приватних мрежа. Додатно, оне се могу формирати и коришћењем различитих софтверских пакета (на пример *OpenVPN*) или виртуалних машина.

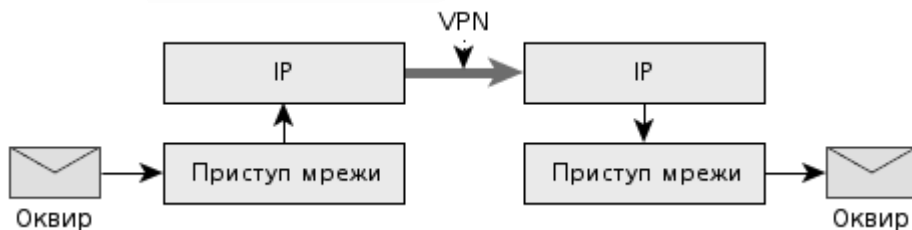
### 5.2.1. Нивои рада виртуалних приватних мрежа

У основи, виртуалне приватне рачунарске мреже могу се поделити на оне које функционишу као мрежни мостови (на слоју везе података) и оне које функционишу као рутери (на мрежном слоју). Са овог аспекта слој функционисања се односи на то на ком слоју оне пружају услугу приватним мрежама које повезују. Предност коришћења VPN-ова који раде као мостови је могућност коришћења мрежних сервиса који захтевају да се чланови који комуницирају налазе у истој мрежи, док је предност рутерског режима рада смањена количина саобраћаја која путује кроз VPN канал.



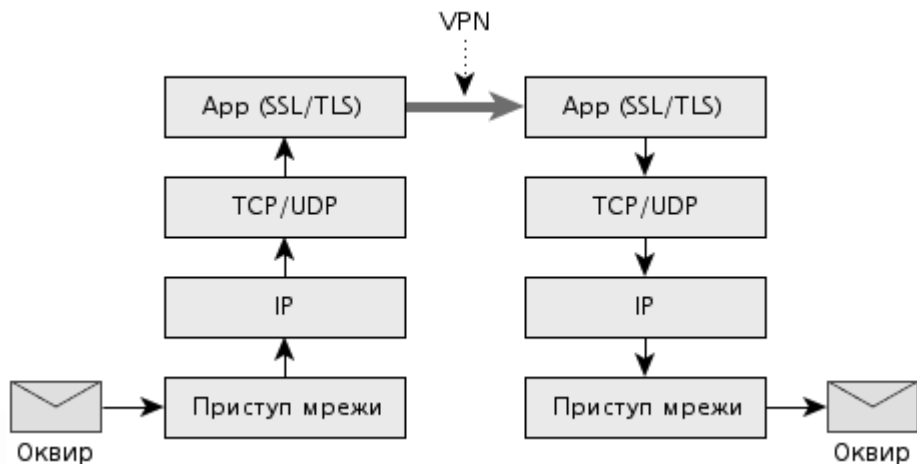
Слика 5.2.1-1 - Виртуална приватна мрежа са мрежним мостовима

Виртуалне приватне мреже које функционишу као мрежни мостови прихватају податке из приватних мрежа које повезују на нивоу оквира, односно пакета протокола на слоју везе података. Оквири се инкапсулирају у пакете виртуалне приватне мреже и прослеђују до одредишне мреже у коју се преносе. На овај начин је омогућено формирање једне велике приватне мреже чији су сегменти међусобно повезани виртуалном приватном мрежом.



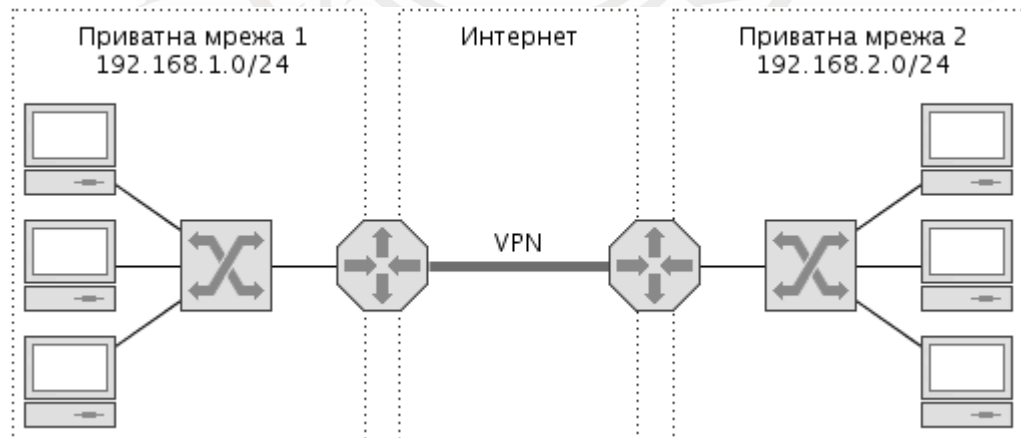
Слика 5.2.1-2 - Виртуална приватна мрежа као мост на мрежном слоју

Подаци се кроз виртуалну приватну мрежу могу преносити инкапсулирани на нивоу мрежног слоја или слоја апликације. У суштини нема значајнијих разлика у томе на ком се нивоу преносе подаци, с обзиром да се шифрују у оба случаја (на пример, *IPsec* на мрежном слоју и *SSL/TLS* као подршка слоју апликације).



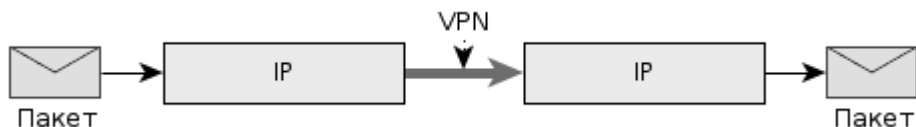
Слика 5.2.1-3 - Виртуална приватна мрежа као мост на слоју апликације

Виртуалне приватне мреже које функционишу као рутери прихватају податке из приватних мрежа које повезују на нивоу пакета мрежног слоја (*IP* датаграма). Ови датаграми се инкапсулирају у пакете виртуалне приватне мреже и прослеђују до одредишне мреже у коју се преносе. На овај начин су приватне мреже које се повезују *VPN*-ом одвојене, односно користе различите опсеге мрежних адреса.

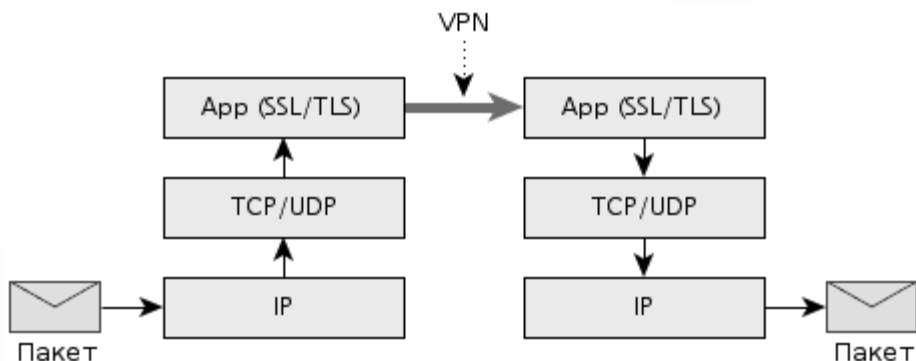


Слика 5.2.1-4 - Виртуална приватна мрежа са рутерима

Са друге стране, датаграми који се преносе кроз *VPN* канал могу бити инкапсулирани у пакете мрежног слоја:



Слика 5.2.1-5 - Виртуална мрежа на мрежном слоју која функционише као рутер или слоја апликације:

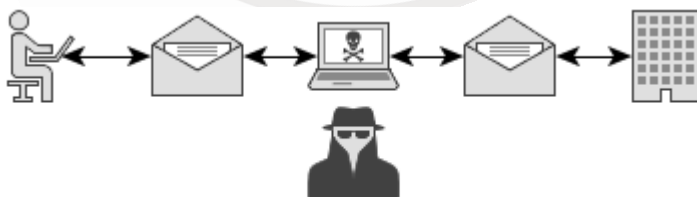


Слика 5.2.1-6 - Виртуална мрежа на слоју апликације у улози рутера

као што је то случај и код *VPN*-ова који се понашају као мостови између приватних мрежа.

## 5.2.2. Безбедносне функције виртуалних приватних мрежа

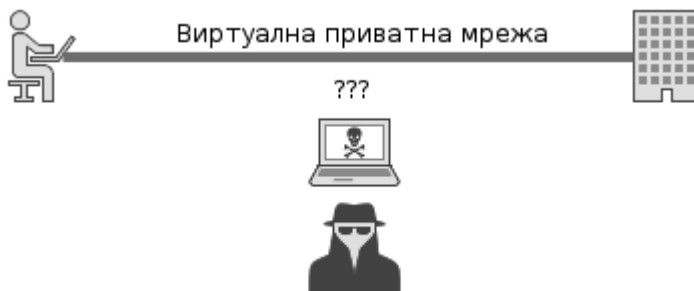
Виртуалне приватне мреже, пре свега, омогућавају поверљивост и веродостојност података који се њима преносе. Ово се постиже шифровањем и дигиталним потписивањем, са коришћењем јавних или посебних алгоритама.



Слика 5.2.2-1 - Подаци који путују јавним каналима омогућавају прислушкивање

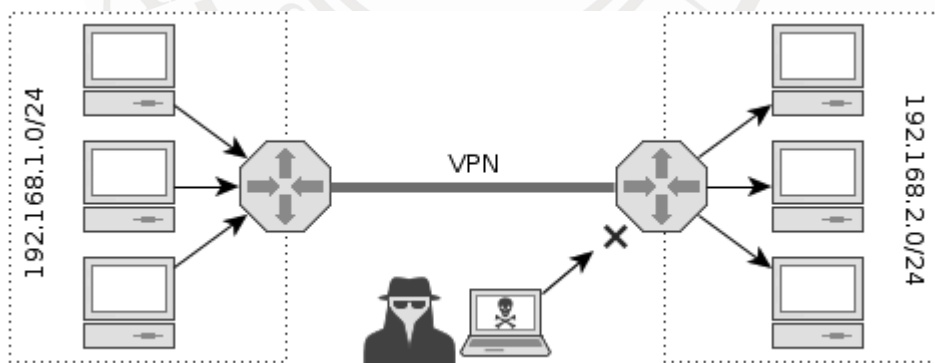
Уколико се подаци кроз јавне комуникационе канале шаљу у изворном облику, нападачи имају могућност да послати садржај читају и мењају. Додатно,

нападаци могу и да пошаљу сопствене пакете лажно се представљајући као једна од регуларних страна у комуникацији. За идентификовање страна које комуницирају путем виртуалне приватне мреже користе се дигитални сертификати, кључеви и парови корисничких имена и лозинки.



Слика 5.2.2-2 - Нападач нема могућност читања и измене података у VPN-у

Следећа значајна безбедносна функција коју виртуалне приватне мреже нуде је контрола приступа. На пример, одређена организација може приступ својим системима у приватној мрежи омогућити само путем виртуалне приватне мреже. То значи да нападач, да би успео да приступи системима у приватној мрежи, мора познавати све параметре за успостављање VPN везе са VPN рутером на граници приватне мреже. Нападач до тих параметара у правилно заштићеним системима који креирају виртуалну приватну мрежу не може доћи. Ово, међутим, са друге стране указује на важност правилне заштите система који се користе за VPN, односно на потенцијалан начин напада на овај вид заштите.



Слика 5.2.2-3 - Само аутентификовани учесници имају улаз у приватну мрежу

Још један тип заштите коју нуде виртуалне приватне мреже је и заштита од напада поновљеним слањем истих пакета.

## 6. Безбедност система доменских имена

Систем доменских имена (енгл. *Domain Name System*) један је од најважнијих инфраструктурних сервиса Интернет мреже. Иако, теоријски гледано, овај сервис није неопходан за сам рад Интернет мреже, он омогућава корисницима да приступе жељеним ресурсима, а неки сервиси, као што је електронска пошта, не могу да функционишу без ослањања на систем доменских имена. Такође, честа је пракса да се више сајтова (са различитим доменима) чува на једном Веб серверу, односно на једној *IP* адреси. Без коришћења система доменских имена ни приступ таквим сајтовима није могућ.

Подаци из базе података са којом ради *DNS* сервер углавном нису поверљиви (мада се у одређеним случајевима могу користити код напада извиђања). Међутим, напади на *DNS* сервис су најчешће усмерени на веродостојност ових података, као и на доступност самог сервиса. Нападом на веродостојност података корисници се могу довести у заблуду да приступају ресурсима на Интернету којима они желе, а да то у ствари буду ресурси подметнути од стране нападача.

Доступност ресурса на Интернету у првом кораку зависи од *DNS* сервиса, јер корисници морају прво да путем овог сервиса сазнају где се сервис налази да би му приступили. Уколико тај корак не да исправне резултате, цео даљи процес приступања је онемогућен.

Постоји и други начин за нападе на доступност путем *DNS* сервиса. На пример, нападач може у базу *DNS* сервера који користи велики број клијената унети лажну адресу на којој се сервер налази за домен коме ти корисници често приступају. У том случају ће даљи захтеви обманутих клијената бити упућени на погрешан сервер, који се може срушити под неочекиваним оптерећењем. Замислите, на пример, да неко нападом на *DNS* сервере компаније Гугл све захтеве корисника њихових услуга преусмери на један сервер неке мање компаније.

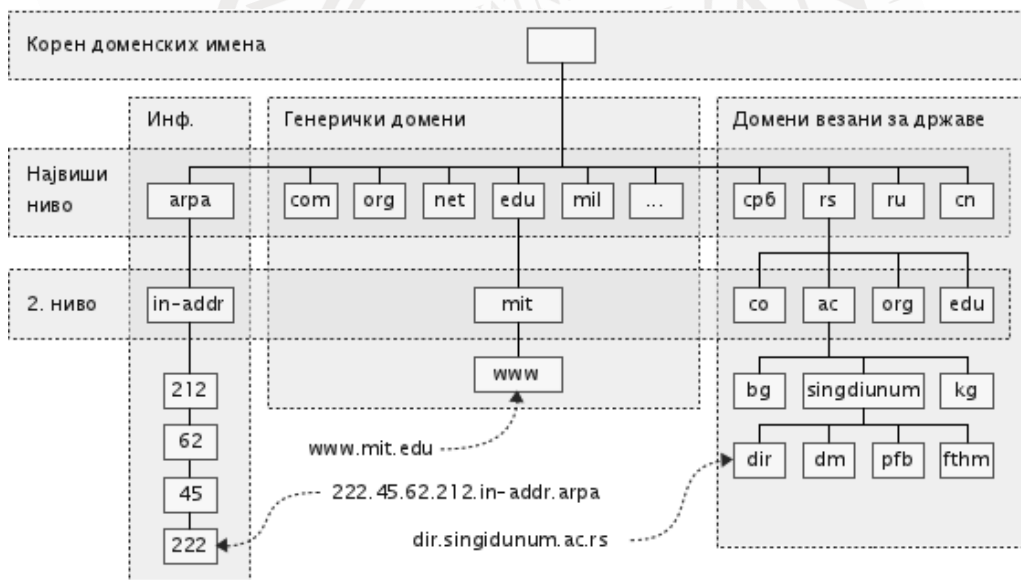
### 6.1. Основни принципи рада система доменских имена

Простор доменских имена је стабло за чији сваки чвор постоји запис у *DNS*-у надлежном за ту зону. У надлежном *DNS* серверу (енгл. *authoritative DNS nameserver*) могуће је за одређену зону декларисати подзоне путем декларисања одговарајућих *DNS* под-сервера.

За разумевање система доменских имена и начина његовог функционисања потребно је разумети саму структуру имена домена (енгл. *domain name*). Назив домена се састоји од два или више делова раздвојених тачкама. Узмимо за пример домен *dir.singidunum.ac.rs*:

- Прва ознака са десне стране представља домен највишег нивоа (енгл. *top level domain, TLD*), у овом случају *rs*.
- Свака наредна ознака гледано са десне стране - *ac*, *singidunum*, и *dir* - представља поддомен. Максималан број поддомена је 127 а сваки од чланова може имати максималну дужину од 63 знакова, с тим да целокупна дужина назива (укључујући све поддомене и тачке којима су раздвојени) не сме прећи 255 знакова.

Домен може имати дефинисано једно или више имена домаћина (енгл. *hostname*) којима су придружене адресе Интернет протокола. У наведеном случају, домен је *dir.singidunum.ac.rs* а име домаћина би могло да буде *www.dir.singidunum.ac.rs* са одговарајућом адресом Интернет протокола 212.62.45.222.



Слика 6.1-1 - Хијерархијска организација система доменских имена

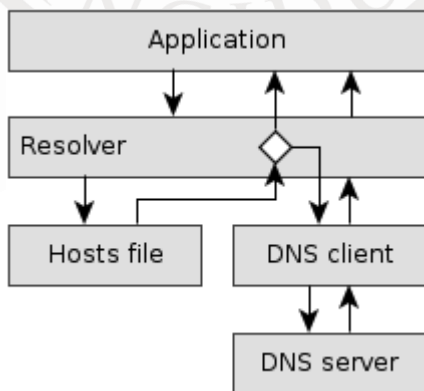
Сервис система доменских имена чине хијерархијски повезани сервери. За сваки од домена мора да постоји декларисан један или више надлежних *DNS* сервера који су задужени за чување и давање информација о њему. Један *DNS*

сервер може бити задужен и за већи број потпуно независних домена. У корену стабла постоје специјални *DNS* сервери који се зову корени сервери (енгл. *root servers*) и они су задужени за домене највишег нивоа - домене на самом корену стабла. Без поменутих корених сервера рад Интернета не би био могућ јер они чине основу сваког доменског именовања на њему. Тренутно постоји 13 корених сервера и њихова имена су [A-M]. *root-servers.net*.

Домен највишег нивоа је прва с десна ознака у сваком имену домена - у домену *dir.singidunum.ac.rs* домен највишег нивоа је *rs*. Постоје три категорије домена највишег нивоа:

- домени највишег нивоа везани за државе - домени дужине два слова везани за земљу или одређени географски простор: *rs* - Република Србија, *ru* - Руска Федерација, *cn* - Народна Република Кина и тако даље;
- генерички домени највишег нивоа - домени који се користе за одређену класу организација: *com* - комерцијални системи, *org* - непрофитне организације, *edu* - образовне установе и тако даље;
- инфраструктурни домени највишег нивоа - једини у овој групи је *arpa* домен.

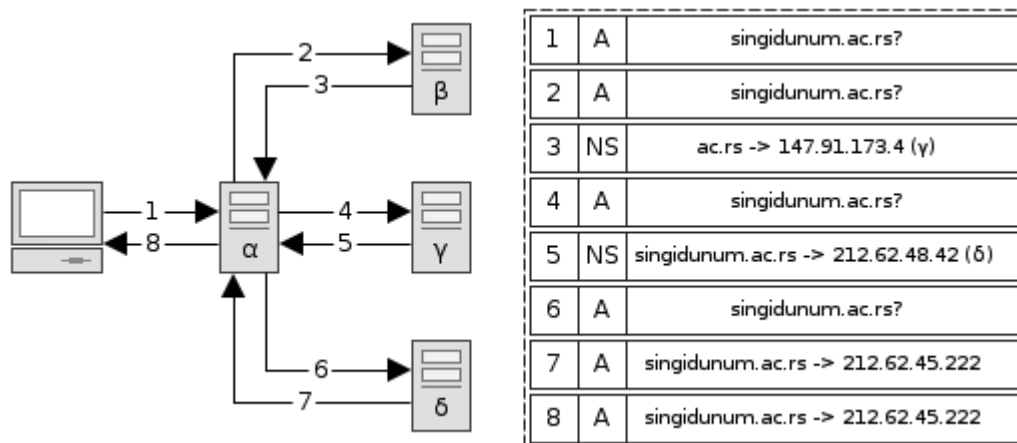
За нашу државу, као и за све државе чије писмо садржи и друге знакове сем енглеског алфабета, значајна је од скора доступна могућност коришћења и међународних знакова у називу домена. У складу са њом, дефинисано је и више домена највишег домена за наведени тип држава. Када су у питању ћирилични домени највишег нивоа, Република Србија је, првенствено захваљујући професионалном и ефикасном раду РНИДС-а, обезбедила сrb домен највишег нивоа, одмах након Руске федерације. Јавна употреба овог домена започела је 2011. године.



Слика 6.1-2 - Процес разрешавања доменског имена



Клијентска компонента *DNS* система назива се разрешивач (енгл. *resolver*). Ова компонента се обраћа *DNS* серверу да би од њега добила адресу Интернет протокола за задато име домена. Разрешивач је системска компонента која се користи посредно, односно путем програма којима је ова услуга потребна. Разрешивачи доменских имена користе системске мрежне параметре који најчешће садрже логичку адресу једног или два *DNS* сервера.



Слика 6.1-3 - Пример потпуног рекурзивног разрешавања домена

Пример коришћења *DNS* услуге:

1. Апликација (на пример, Веб браузер) добија униформни локатор ресурса <http://www.dir.singidunum.ac.rs/index.php> од стране корисника и рашчлањује га на протокол (*http*), име домаћина ([www.dir.singidunum.ac.rs](http://www.dir.singidunum.ac.rs)) и локалну адресу ресурса (*/index.php*).
2. Апликација се обраћа разрешивачу доменских имена у циљу добијања адреса Интернет протокола за тражено име домаћина.
3. Разрешивач доменских имена се обраћа *DNS* серверу из мрежне конфигурације рачунара питањем: "Да ли знаш која је адреса Интернет протокола доменског имена [www.dir.singidunum.ac.rs](http://www.dir.singidunum.ac.rs)?"
4. Уколико *DNS* коме се разрешивач обратио није надлежан за домен у коме се тражени домаћин налази ([dir.singidunum.ac.rs](http://dir.singidunum.ac.rs)) он се обраћа једном од корених *DNS* сервера питањем: "Који је *DNS* сервер надлежан за *rs* домен?"
5. Корени сервер враћа одговор: "147.91.8.6".
6. *DNS* сервер се обраћа серверу 147.91.8.6 питањем: "Који је *DNS* сервер надлежан за *ac.rs* домен?".

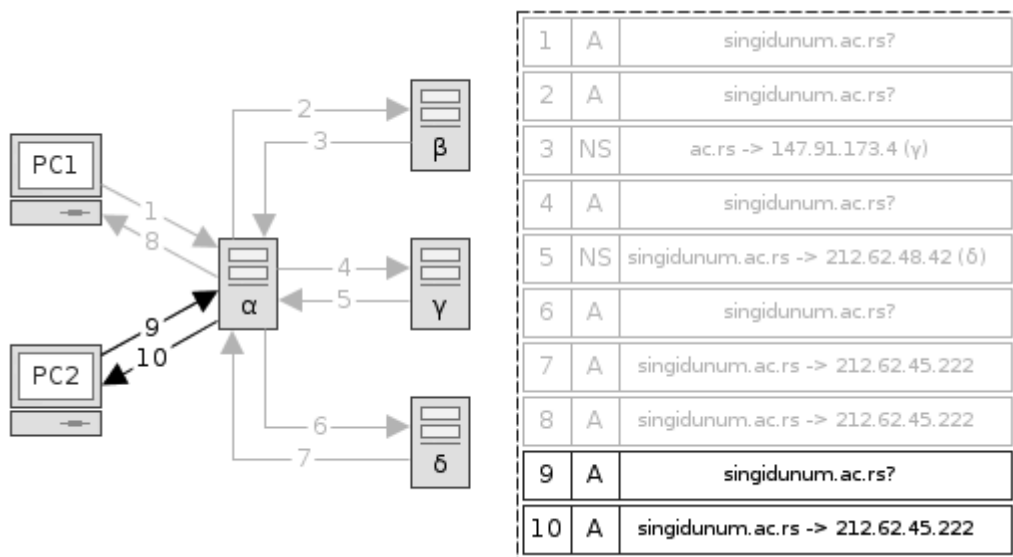
7. *DNS* сервер на адреси 147.91.8.6 враћа одговор: "147.91.8.21".
8. *DNS* сервер се обраћа серверу на адреси 147.91.8.21 питањем: "Који је *DNS* сервер надлежан за домен *singidunum.ac.rs*?"
9. Сервер на адреси 147.91.8.21 враћа одговор: "212.62.48.42".
10. *DNS* сервер се обраћа серверу на адреси 212.62.48.42 са питањем: "Који је *DNS* сервер надлежан за домен *dir.singidunum.ac.rs*?"
11. Сервер на адреси 212.62.48.42 враћа одговор: "212.62.45.222".
12. *DNS* сервер се обраћа серверу на адреси 212.62.45.222 питањем: "Која је адреса домаћина *www.dir.singidunum.ac.rs*?"
13. Сервер на адреси 212.62.48.222 враћа одговор: "212.62.45.222"
14. *DNS* сервер враћа одговор клијенту чији му се разрешивач обратио: "адреса Интернет протокола за домаћина *www.dir.singidunum.ac.rs* је 212.62.45.222".

На овај начин апликација на клијентском рачунару добија адресу Интернет протокола Веб сервера и путем те адресе прослеђује захтев за Веб страницом */index.php*. Овај пример објашњава рекурзију у раду *DNS*-а. Треба имати у виду да један *DNS* сервер може чувати информације о више различитих домена као и да више *DNS* сервера могу пружати информацију о једном домену.

### **6.1.1. Кеширање код система доменских имена**

На примеру потпуног рекурзивног *DNS* разрешавања приказан је скуп корака који је теоретски неопходно проћи да би клијент добио информацију од сервера. У пракси би, међутим, овакав начин рада код сваког *DNS* упита за сваки Интернет домен створио огромно оптерећење свих *DNS* сервера који учествују у разрешавању одређеног домена. Ово се пре свега односи на корене *DNS* сервере и *DNS* сервере којима клијент директно приступа. Да би се избегло поменуто оптерећење уведено је кеширање резултата.

Кеширање код *DNS*-а има за циљ да омогући сваком од *DNS* сервера смањење броја упита које он поставља осталим *DNS* серверима при разрешавању упита везаног за домене из зоне за које он није надлежан. Уколико је кеширање укључено на *DNS* серверу, он у својој интерној бази чува резултате свих успешно обављених разрешавања тако да, уколико се исти упит понови, сервер не мора да тражи поново све информације од осталих *DNS* сервера већ користи потојећу информацију из базе.



Слика 6.1.1-1 - Разрешавање домена са кеширањем информација

Овакав начин рада штеди процесорске ресурсе и комуникационе канале DNS сервера али отвара и ново питање - уколико се информација о домену на за њега надлежном DNS серверу измени, како ће се то одразити на клијенте који врше упит за тај домен посредством других сервера који у својој бази имају забележену претходну информацију? Одговор на ово питање лежи у ограничењу периода важења (енгл. *Time To Live, TTL*) информација које је надлежни сервер дао. Овај параметар одређује након ког времена ће посреднички DNS сервери обновити информацију у својој бази везану за тај домен.

Период важења се изражава у секундама и најчешће је постављен на 86.400 секунди, односно један дан. То у пракси значи да је максимално време (након измене домена на надлежном DNS серверу) током кога ће клијенти добијати застарелу информацију од посредничких DNS сервера један дан. Остали системски параметри сваке информације о имену домена су:

1. *Serial*: серијски број зоне који се увећава при свакој измени података, а служи осталим серверима за утврђивање да ли се информација изменила на главном серверу.
2. *Refresh*: број секунди након кога ће *slave* и *secondary* сервери освежити своје податке за зону.
3. *Retry*: број секунди након кога ће *slave* и *secondary* сервери поново покушати освежавање података са master сервера уколико претходни

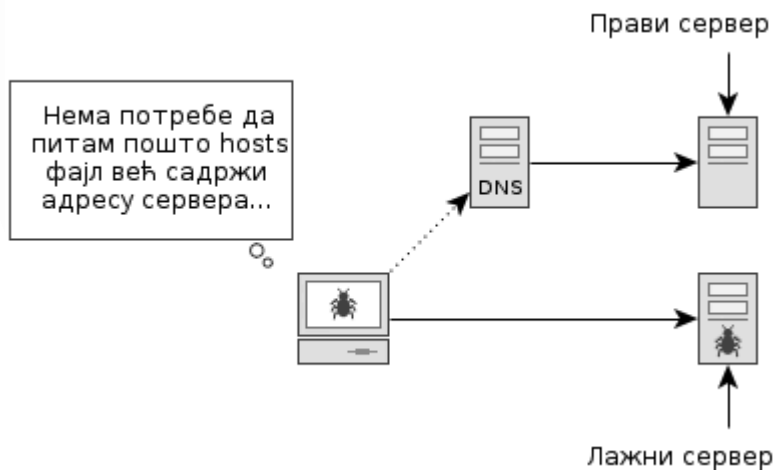
покушај не успе.

4. *Expire*: број секунди након кога ће *slave* и и *secondary* сервери одустати од покушаја да освеже своју базу са *master* сервера уколико претходни покушаји не успеју.

## 6.2. Напади путем хостс фајла

Хостс фајл има предност над коришћењем DNS сервера за превођење одређеног назива домена у IP адресу. То значи да, уколико у хостс фајлу постоји адреса коју треба разрешити, неће се користити услуге DNS сервера већ ће се користити вредност из хостс фајла. Међутим, овакав однос компонената у разрешавању адреса отвара и потенцијалне безбедносне проблеме.

Основни напад коришћењем хостс фајла има за циљ да корисника усмери на лажни сервер, који је најчешће под контролом нападача и на коме се најчешће налази реплика садржаја који корисник очекује да види. На тај начин се нападачу омогућава да жртви подметне лажне информације, као и да од ње преузме поверљиве податке, као што су корисничко име и лозинка.



Слика 6.2-1 - Преусмеравање жртве на лажни сервер нападом на хостс фајл

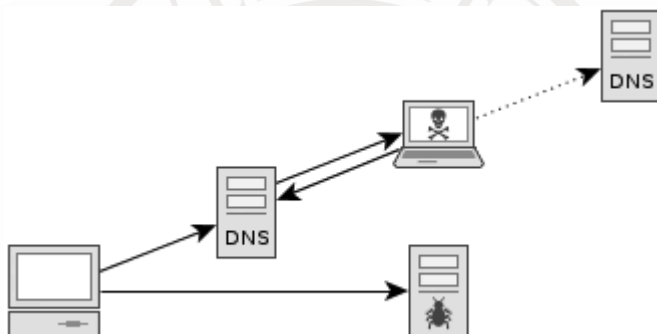
На пример, уколико нападач у хостс фајл жртве унесе следећи запис, који уместо праве IP адресе сервера води на адресу на којој се налази сервер нападача:

203.0.113.123	accounts.google.com
---------------	---------------------

рачунар жртве ће приликом посете наведеној адреси комуницирати са сервером нападача. Нападач у том случају може да на сервер постави страницу која изгледа идентично као и страница за пријављивање на Гугловим серверима, и да на тај начин од корисника украде корисничко име и лозинку.

### 6.3. Тровање кеша DNS сервера

Тровање кеша *DNS* сервера (енгл. *DNS cache poisoning*) је тип напада код кога нападач обмањује кориснике услуга *DNS* сервера тако што у његов кеш убацује погрешне податке. Ово најчешће чини уметањем између *DNS* сервера који користи жртва и *DNS* сервера који је надлежан за домен коме жртва жели да приступи. Након тога, рачунар нападача игра улогу надлежног *DNS* сервера и *DNS* серверу жртве шаље погрешну информацију. Алтернативни начин је пресретање и измена одговора који је послао надлежни *DNS* сервер.

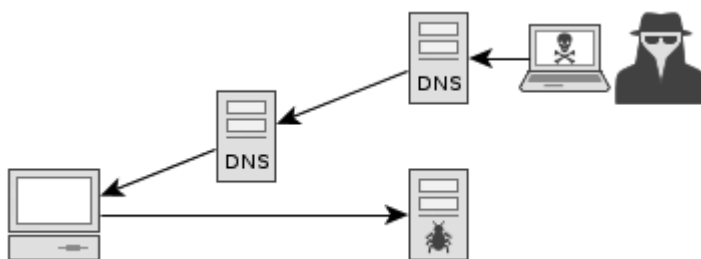


Слика 6.3-1 - Тровање посредовањем

Да би се спречио овакав тип напада потребно је омогућити аутентификацију *DNS* сервера, односно дигитално потписивање информација које они шаљу. На тај начин је нападач спречен и у лажном представљању, и у измени пресретнутих података.

### 6.4. Измена информација на серверу

Један од начина да нападач клијентима подметне лажну информацију о неком домену је да изврши напад директно на надлежне сервере, односно да у њихове базе унесе погрешне информације. То је могуће учинити нападом на сам сервер и преузимањем контроле над њим, а затим ручном изменом података. Ово је, међутим, у већини случајева веома тешко или немогуће извести.



Слика 6.4-1 - Подметање лажних информација на надлежни DNS сервер

Алтернатива претходном начину измене података је коришћење динамичког ажурирања. На пример, DNS сервер може бити подешен да дозвољава да се измене на њему врше аутоматски, слањем новог стања са одређене IP адресе:

```
zone "singidunum.ac.rs" {
    type master;
    file "db.singidunum.ac.rs";
    allow-update { 192.168.1.100; };
};
```

У датом примеру сервер ће прихватити измене које стижу са адресе 192.168.1.100. Међутим, уколико нападач успе да преотме ову адресу (енгл. *IP address hijacking*), он ће моћи и да у базу надлежног сервера унесе нетачне информације. Да би се овакав тип напада спречио потребно је користити DNSSEC безбедносно проширење система доменских имена које омогућава аутентификацију сервера.

## 6.5. Напади извиђања и DNS сервис

Информације које пружају DNS сервери су најчешће јавне. Међутим, базе на основу којих се те информације генеришу могу садржати информације које су поверљиве, пре свега због тога што садрже инфраструктурне информације које могу послужити за напад на сервере и сам сервис. Из тог разлога нападачи код напада извиђања покушавају да дођу и до оваквих информација.

Један од најједноставнијих приступа за преузимање зонских фајлова, односно фајлова на основу којих DNS сервери генеришу одговоре на упите клијената, је лажно представљање нападача као слејв DNS сервер и захтевање од примарног сервера да пошаље своју базу, односно зонски фајл. Оваква размена фајлова је регуларан начин за динамичко ажурирање информација на серверима, али у

овом случају оно чини основу за напад.



Слика 6.5-1 - Преузимање базе представљањем као слејв сервер

Да би се овакав тип напада спречио потребно је користити *DNSSEC* безбедносно проширење система доменских имена. Тиме се могућност трансфера базе ограничава само на аутентификоване сервере.

### 6.5.1. DNS Cache Snooping

Напад под називом „забадање носа у кеш *DNS* сервера“ (енгл. *DNS Cache Snooping*) нема за циљ да крајњем кориснику *DNS* сервиса протури лажне податке, већ да утврди којим доменима је он приступао. То се врши анализом кеша на *DNS* серверу, а она се може извести на следећа три начина:

1. остваривањем приступа фај-систему *DNS* сервера на коме се налази кеш,
2. слањем упита за одређене домене са мерењем времена одзива и
3. слањем упита код којих је *RD (Recursion Desired)* индикатор искључен.



Слика 6.5.1-1 - Провера којим сајтовима су клијенти приступали

Предност последња два приступа је могућност добијања информације без пробоја у *DNS* сервер, док је њихова мана ограниченост на потврђивање само оних домена за које нападач пошаље упит. Другим речима, њима није могуће откривање којим све доменима је корисник приступао, већ само да ли је приступао доменима из листе нападача.

## 6.6. Безбедносна проширења система доменских имена

Безбедносна проширења система доменских имена (енгл. *Domain Name System Security Extensions, DNSSEC*) чине допуну *DNS* сервиса којом се његовим корисницима нуди виши ниво безбедности. С обзиром на то да основна поставка система доменских имена не нуди никакав систем заштите од измене порука или лажног представљања, основни задатак *DNSSEC*-а је да одговори на тај недостатак. Другим речима, поменута проширења омогућавају веродостојност информација које се дистрибуирају *DNS* сервисом. Она се остварује коришћењем асиметричне криптографије, односно дигиталним потписивањем порука.





## 7. Безбедност Веб сервиса

Веб сервис је постао најкоришћенији сервис Интернет мреже, пре свега захваљујући преузимању инфраструктурне улоге, односно омогућавању коришћења других сервиса као својих подсервиса. На пример, све већи број корисника електронске поште прелази са традиционалних десктоп клијената на Веб клијенте. У последње време се, чак, појавило више покушаја да се комплетан графички кориснички интерфејс и радно окружење персоналних рачунара замени Веб браузером.

Оваквим развојем Веб сервиса и његових технологија јавили су се, очекивано, и нови безбедносни проблеми. У суштини, могући су напади на све компоненте у испоруци Веб сервиса: Веб клијента, протоколе и Веб сервер. У наставку су дати основни безбедносни проблеми савременог Веба.

### 7.1. Напади на Веб корисничке агенте

Веб кориснички агенти или Веб браузерери су прешли огроман развојни пут - започевши као једноставни конзолни интерпретери пар десетина *HTML* ознака они су данас постали сложени готово као и оперативни системи. У прилог томе говори и чињеница да се некадашња битка око доминације на пољу оперативних система данас све више измешта управо на поље Веб браузера.

#### 7.1.1. Преотимање сесије

Преотимање сесије (енгл. *session hijacking*) је поступак код кога нападач преузима улогу корисника који се регуларно аутентификовао за рад у Веб апликацији. На пример, корисник се може пријавити на Веб сервер коришћењем свог корисничког имена и лозинке, док му нападач након тога може преузети сесију, односно наставити да ради у Веб апликацији у улози пријављеног корисника. При том, исправан корисник нема информацију о томе шта се догодило, односно он наставља несметано да ради у Веб апликацији, паралелно са нападачем.

Преотимање сесије се заснива на концепту да корисник не шаље своје корисничко име и лозинку при сваком захтеву серверу већ то чини само једном, а након тога добија **идентификатор сесије** (енгл. *session id*) чијим достављањем му се одобравају даљи захтеви. Идентификатор сесије је у суштини сесијски кључ који је случајно генерисан на страни сервера и потребно је обезбедити

његову тајност. Међутим, код коришћења незаштићеног *HTTP* протокола нападач може веома лако сазнати вредност овог кључа, уколико има приступ комуникационом каналу.

За одбрану од оваквог типа напада постоји више приступа. На пример, раније је било популарно интегрисање вредности *IP* адресе у сесијски идентификатор, тако да би нападач, да би могао да приступи Веб апликацији, морао и да преотме *IP* адресу регуларног корисника. Овакав приступ се све више напушта због мобилности клијената, односно због потребе да се настави несметан рад у Веб апликацији и приликом промене *IP* адресе, условљене променом приступне тачке Интернету током кретања. У суштини, најбоља заштита од овог типа напада је коришћење *HTTPS* уместо *HTTP* протокола, односно онемогућавање приступа нападача сесијском идентификатору путем шифровања.

### 7.1.2. Приступ програмима и подацима на клијенту

Веб браузерери имају задатак да заштите рачунар клијента од малициозног садржаја на који наиђу. Под малициозним садржајем се може сматрати било шта, од активних компонената (*JavaScript*, *Java*, *Flash*...) па до обичног *HTML*-а. Узмимо за пример следећи *HTML* код:

```
<input type="file" value="C:\Users\AJ\zrm.pdf" style="display: none" />
```

Уколико Веб браузер не би имао забрану предефинисања вредности контроле за унос фајлова, коришћењем оваквог кода нападач би могао да са рачунара посетиоца преузме било који фајл. Додатно, пошто је у стилу приказа контроле дефинисано да се она не приказује, посетилац уопште не би био свестан да је до слања фајлова дошло.

Када су у питању активне компоненте, ситуација постаје још сложенија. На пример, Јава аплети од корисника траже једноставну потврду да би се извршили, а након тога имају готово исте привилегије као и програми које је корисник инсталирао на рачунар. Или, у савременим Веб браузерима *JavaScript* кодовима није дозвољен приступ клипборд меморији. У противном, нападач би могао да ишчитава садржај ове меморије код посетилаца, а у њој се могу налазити поверљиви подаци.

## 7.2. Напади на Веб сервере

Веб сервери су, као и Веб клијенти, доживели огроман развој у циљу

одговарања на нове захтеве који су постављани пред Веб сервис. Улога првих Веб сервера је била да са спољне меморије учитају тражени фајл и проследи његов садржај клијенту. Данас, Веб сервери представљају огромне софтверске конструкције, са огромним бројем уграђених функција и додатних модула. Од Веб сервера се данас тражи подршка за извршавање програма на страни сервера (*PHP, ASP, JSP* и друго), могућност контроле приступа, преписивања захтева клијената, рад у прокси-режиму, отпорност на *DoS* нападе и много тога другог.

Напади на Веб сервер се могу извршити на много начина. Основни напади на Веб сервере су засновани на пропустима у њиховој реализацији. Међутим, напади се исто тако могу извршавати и путем пропуста у оперативном систему на коме се Веб сервер извршава, пропуста у подешавањима и пропуста у компонентама за програмирање на страни сервера.

Основни савет за заштиту Веб сервера је његово инсталирање на заштићеном оперативном систему, искључивање свих функција које нису неопходне, правилно подешавање коришћених функција и држање верзије Веб сервера и укључених компонената ажурним. Додатно, потребно је периодично користити и аутоматизоване безбедносне скенере за проверу да ли су се у међувремену појавили пропусти.

Још једна напомена везана за ажурирање верзија Веб сервера и додатних компонената је та да приликом инсталирања нове верзије треба искључити сервер са мреже, а након инсталирања пажљиво проверити да ли су сва подешавања правилно прихваћена на новој верзији. Аутор овог текста је више пута био сведок успешног извршавања напада на Веб сервере као последице аутоматизованог ажурирања приликом чега је дошло до замене конфигурационих параметара подразумевањем.

### 7.2.1. Фајл *robots.txt*

Фајл *robots.txt* се користи да би информисао Веб претраживаче о томе које садржаје треба да индексују и колико често, као и које садржаје треба да искључе из индексовања. На пример, уколико желимо да роботима свих претраживача забранимо приступ свим садржајима на сајту, у овај фајл бисмо унели следећи садржај:

```
User-agent: *  
Disallow: /
```

Обично се овим фајловима претраживачима забрањује да индексују и објављују информације о секцијама у којима се налазе фајлови, слике, административне функције и слично, односно секцијама у којима се налазе осетљиви садржаји којима корисници не би смели директно да приступају. Међутим, с обзиром да је приступ фајлу *robots.txt* јавно доступан, нападачи могу искористити ове фајлове управо за сазнавање на којим локацијама се налазе осетљиви садржаји. Дакле, приликом креирања овог фајла потребно је водити рачуна о томе да се у њему не изложе било које информације о потенцијално безбедносно осетљивим деловима сајта.

### 7.3. Напади на Веб апликације

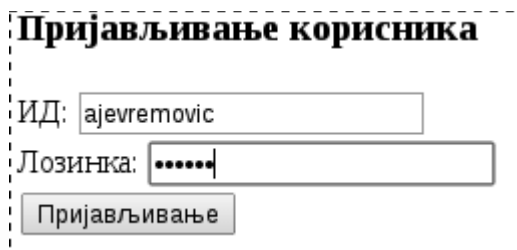
Веб апликације представљају посебан подсистем унутар постојећег телекомуникационог система рачунарских мрежа и Веб сервера у њима. Заправо, Веб апликације данас постају најчешћа форма реализације софтвера, па самим тиме постају и све сложеније, као и технологије на којима се заснивају. Из тог разлога се јављају и нови безбедносни пропусти који се заснивају на Веб апликацијама. У основне безбедносне проблеме Веб апликација спадају:

- проблеми са аутентификацијом,
- проблеми са ауторизацијом,
- проблеми са управљањем сесијом,
- проблеми са уметањем кода и података,
- проблеми са путањама на фајл-систему,
- проблеми са логичким пропустима,
- проблеми са безбедношћу на страни клијента,
- проблеми са нежељеним откривањем информација,
- проблеми са ескалацијом привилегија,
- проблеми са активним компонентама у Веб страницама и
- проблеми изазвани рањивостима у Веб серверу и оперативном систему.

Заштита Веб апликација је сложено питање, управо из разлога што су и саме Веб апликације сложени системи који функционишу унутар сложених система (Веб сервера и рачунарских мрежа уопште). Додатни отежавајући фактор чини и то што су Веб апликације често јавно доступне путем Интернета. Са друге стране, постоји велики број концепата, алата и решења за подизање нивоа безбедности Веб апликација и сервера на којима се оне извршавају.

### 7.3.1. Напади уметањем SQL наредби

Један од најопаснијих типова напада на Веб апликације је уметање *SQL* наредби. Овај напад се реализује тако што корисник посебним уносима мења *SQL* наредбе које слој логике шаље серверу базе података и на тај начин врши произвољне акције у бази података. За илустрацију овог типа напада ће бити коришћен једноставан пример код кога се корисници пријављују за рад у Веб апликацији путем *HTML* формулара:



Пријављивање корисника

ИД:

Лозинка:

Слика 7.3.1-1 - *HTML* формулар за пријављивање корисника

Корисничка имена и лозинке се чувају у бази података а улазни подаци (из приказаног формулара) се, у односу на податке из базе података, проверавају на страници са следећим *PHP* кодом:

```
<?php
$id      = $_POST['id'];
$lozinka = $_POST['lozinka'];
$upit = "select * from Korisnik where id='$id' and lozinka='$lozinka'";
$resultat = mysqli_query($link, $upit);
if (mysqli_num_rows($result) == 1)
{
    // Приступ апликацији је одобрен
    ...
}
```

Уколико корисник за ИД унесе вредност *ajevremovic* а као лозинку *abc123*, упит који ће се извршити у бази изгледаће овако:

```
select * from Korisnik where id='ajevremovic' and lozinka='abc123'
```

Међутим, уколико корисник за вредност лозинке унесе:

```
' or true; --
```

целокупан упит ће изгледати овако:

```
select * from Korisnik where id='ajevremovic' and lozinka='' or true; --'
```

Овакав упит ће учитати ред са корисником чији је ИД *ajevremovic*, без обзира на то што корисник не зна која је лозинка за тај налог. Другим речима, с обзиром на наведени код за аутентификацију корисника, нападач ће моћи да се пријави за рад у Веб апликацији за било који налог, без познавања лозинке.

У неким ситуацијама би још штетнији ефекат имао следећи приступ, у коме нападач као лозинку уноси:

```
'; drop table Korisnik; --
```

У случају уноса ове вредности коначан упит би изгледао:

```
select * from Korisnik where id='ajevremovic' and lozinka=''; drop table Korisnik; --'
```

Дакле, уносом ове вредности нападач би могао да натера систем да комплетно уклони табелу *Korisnik* из базе података.

За одбрану од приказаног типа напада потребно је **нормализовати** податке који пристижу од корисника, а који ће се користити за формирање упита ка бази података. Под нормализацијом се подразумева довођење унетих вредности у предвиђени опсег. У наставку је дат код који ће искористити уграђене функције *mysql* проширења за избегавање овог типа напада:

```
$id      = mysqli_real_escape_string($link, $_POST['id']);  
$lozinka = mysqli_real_escape_string($link, $_POST['lozinka']);
```

Применом ове функције би претходно наведени улазни текст нападача:

```
' or true; --
```

био преведен у:

```
\' or true; --
```

тако да би коначан упит изгледао овако:

```
select * from Korisnik where id='ajevremovic' and lozinka='' or true; --'
```

односно био би третиран као обичан текст и не би променио логику упита.

Нормализација улазних података може се вршити и кроз инсистирање на типу података или применом функција чији је резултат безбедан скуп вредности:

```
$starost = (int) $_POST['starost']; // цео број  
$visina = (double) $_POST['visina']; // децималан број  
$zaposlen = (bool) $_POST['zaposlen']; // логичка вредност  
$lozinka = md5($_POST['lozinka']); // хеш функција
```

Дакле, да би Веб апликација била отпорна на нападе путем уметања SQL кода, потребно је увек проверавати и нормализовати улазне податке који долазе из непроверених извора (корисничког уноса, фајлова и слично).

### 7.3.2. Унакрсно скриптовање

Унакрсно скриптовање (енгл. *Cross Site Scripting, XSS*) је тип напада специфичан за динамичке Веб апликације. Код овог типа напада нападач прави посебан захтев ка Веб серверу који је тако формиран да у садржај изворног одговора умеће произвољан садржај. На тај начин се посетилац доводи у заблуду да садржај потиче од стварног аутора странице.

Да би се напади унакрсним скриптовањем извршили неопходно је да странице које се нападају садрже одређени тип пропуста. Овај пропуст се огледа у томе да се кориснички унос директно репродукује у одговору, без обраде у циљу избегавања недовољених вредности. На пример, уколико одређена страница садржи следећи код:

```
<input value="<?php echo $_REQUEST['unos']; ?>">
```

а нападач као вредност уноса зада:

```
"><a href="http://maliciozni.host/virus.exe">Инсталирајте овај  
програма</a><input type="hidden" value="
```

или регуралног корисника усмери на адресу:



```
http://napadnuti.host/stranica-sa-propustom.php?unos=%E2%80%9C%3E%3Ca+href%3D%E2%80%9Dhttp%3A%2F%2Fmaliciozni.host%2Fvirus.exe%E2%80%9D%3E%D0%98%D0%BD%D1%81%D1%82%D0%B0%D0%BB%D0%B8%D1%80%D0%B0%D1%98%D1%82%D0%B5+%D0%BE%D0%B2%D0%B0%D1%98+%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%3C%2Fa%3E%3Cinput+type%3D%E2%80%9Dhidden%E2%80%9D+value%3D%E2%80%9D
```

коначан резултат у виду *HTML* кода ће изгледати овако:

```
<input value="">
<a href="http://maliciozni.host/virus.exe">Инсталирајте овај програм</a>
<input type="hidden" value="">
```

Другим речима, посетиоцу ће се појавити хипер-линк ка преузимању малициозног софтвера, а он ће сматрати да му инсталацију тог софтвера предлаже аутор странице коју је посетио.

### 7.3.3. Откривање верзије Веб апликација

Пример једне активности у оквиру напада извиђањем представља утврђивање верзије Веб апликације са отвореним кодом, а која се користи на серверу који се напада. Оваква активност се назива „узимање отисака прстију“ (енгл. *fingerprinting*) а заснива се на анализи карактеристика апликације које су видљиве споља.

За споља видљиве особине апликације најчешће се користе статични *JavaScript* и *CSS* фајлови с обзиром да су они углавном доступни кроз изоловане захтеве, без потребе да се приступалац аутентификује и ауторизује. Оно што нападач испитује на серверу јесте постојање одређених фајлова, као и специфичности у њиховом садржају.

Да би нападач на основу постојања поменутих *JavaScript* и *CSS* фајлова и њихових садржаја могао да утврди о којој верзији софтвера је реч, он претходно мора да утврди специфичности свих постојећих верзија. У наставку је дат пример откривања верзије *WordPress*-а, популарног решења за прављење блогова.

Први корак у откривању верзије *WordPress*-а која се користи на неком серверу је прављење локалне базе специфичности различитих верзија, односно *JavaScript* и *CSS* фајлова који су специфични за одређену верзију. За ово је потребно скинути све претходне верзије овог софтвера, или макар оне верзије из опсега за који се претпоставља погодак. С обзиром на то да је *WordPress* бесплатан



софтвер са јавно доступним изворним кодом, све верзије су слободно доступне на адреси: <http://wordpress.org/download/release-archive/>. На овој адреси стоји обавештење да архива садржи све објављене верзије софтвера, као и да није безбедно користити верзије раније од последње. За потребе овог примера преуземамо последње четири верзије софтвера:

```
$ wget http://wordpress.org/wordpress-3.6.tar.gz
$ wget http://wordpress.org/wordpress-3.7.tar.gz
$ wget http://wordpress.org/wordpress-3.8.tar.gz
$ wget http://wordpress.org/wordpress-3.9.tar.gz
```

Након преузимања софтвера извршићемо распакивање архива у директоријуме чији називи одговарају верзијама:

```
$ tar xzf wordpress-3.6.tar.gz && mv wordpress wordpress-3.6
$ tar xzf wordpress-3.7.tar.gz && mv wordpress wordpress-3.7
$ tar xzf wordpress-3.8.tar.gz && mv wordpress wordpress-3.8
$ tar xzf wordpress-3.9.tar.gz && mv wordpress wordpress-3.9
```

Након распакивања архива креираћемо текстуални фајл *razlike.txt* у који ћемо сместити резултате анализе за утврђивање разлика. Утврђивање разлика ћемо извршити путем *diff* алата, поређењем по две суседне верзије, а уз укључивање само *JavaScript* и *CSS* фајлова:

```
$ touch razlike.txt
$ diff -r -q wordpress-3.6 wordpress-3.7 | \
  grep -E '^.*\.(js|css)' >> razlike.txt
$ diff -r -q wordpress-3.7 wordpress-3.8 | \
  grep -E '^.*\.(js|css)' >> razlike.txt
$ diff -r -q wordpress-3.8 wordpress-3.9 | \
  grep -E '^.*\.(js|css)' >> razlike.txt
```

Након извршавања наведених наредби у фајлу *razlike.txt* налазе се информације о томе који фајлови постоје само у одређеној верзији, као и између којих верзија постоје разлике у одређеним фајловима:

```
...
Files wordpress-3.6/wp-admin/css/wp-admin.min.css and wordpress-3.7/wp-
admin/css/wp-admin.min.css differ
```

```
Only in wordpress-3.7/wp-admin/js: about.js
Only in wordpress-3.7/wp-admin/js: about.min.js
Files wordpress-3.6/wp-admin/js/accordion.min.js and wordpress-3.7/wp-
admin/js/accordion.min.js differ
...
```

На основу добијених резултата можемо даље испитивати у чему се огледају разлике између различитих верзија фајлова:

```
$ diff wordpress-3.8/wp-includes/js/wp-lists.js \
    wordpress-3.9/wp-includes/js/wp-lists.js
312c312,316
<  e = $( $.trim(e) ); // Trim leading whitespaces
---
>  if ( 'string' == typeof e ) {
>    e = $( $.trim( e ) ); // Trim leading whitespaces
>  } else {
>    e = $( e );
>  }
```

Дакле, можемо утврдити да су аутори у верзији 3.9 проширили код додавањем провере типа променљиве, што можемо користити или за евентуални напад на претходне верзије (с тим да овде за то не постоји озбиљна основа), или за откривање верзије софтвера која се на неком серверу користи.

Уместо наведеног појединачног залажења у разлике, једноставнији начин чини израчунавање хеш вредности за сваку верзију фајла:

```
$ md5sum wordpress-3.*/wp-admin/css/color-picker.min.css
3b55aee012fbb85df3b3fd5d0f85a60c  wordpress-3.6/wp-admin/css/color-
picker.min.css
9d746a565ddc0e7ad9f9521644289b90  wordpress-3.7/wp-admin/css/color-
picker.min.css
37cbd34c7179a2f7445918849718d8fb  wordpress-3.8/wp-admin/css/color-
picker.min.css
c976ab722a8e8698ca56a38df290504b  wordpress-3.9/wp-admin/css/color-
picker.min.css
```

На основу овакве базе је прилично лако утврдити коју верзију *WordPress-a*

користи одређени сајт. На пример, уколико желимо да утврдимо која се верзија софтвера користи на званичном сајту на коме се промовише сама поменута апликација, преузећемо један од наведених фајлова и израчунати његову хеш вредност:

```
$ wget http://wordpress.org/wp-admin/css/color-picker.min.css
$ md5sum color-picker.min.css
c976ab722a8e8698ca56a38df290504b color-picker.min.css
```

Упоредивањем резултата и базе можемо закључити да се на сајту *wordpress.org* користи (очекивано) последња верзија *WordPress*-а која је доступна у тренутку писања овог примера. У случају да је добијен другачији резултат, односно да је утврђено да се не користи последња верзија софтвера, следећи корак би био проналажење откривених пропуста за ту верзију и њихово искоришћавање за постизање малициозног циља.

Јасно да се цео описани процес може једноставно аутоматизовати, што и јесте честа пракса нападача. Аутоматизација у овом погледу обухвата и прављење посебних софтверских агената (енгл. *Web crawler*) који претражују сајтове на Интернету и праве локалну базу сајтова на којима се користи одређени софтвер, уз забележавање верзије. У тренутку када се за одређену верзију пронађе безбедносни пропуст, нападачи користе аутоматизоване софтверске агенте који сајтовима из базе приступају и користе поменути рањивост за преузимање поверљивих података из базе, постављање сопственог садржаја, уметање малициозног садржаја за напад на посетиоце и слично.

#### 7.3.4. Откривање рањивости у претходним верзијама

На сличан начин на који нападачи могу путем *CSS* и *JavaScript* фајлова утврдити верзију јавно доступног софтвера који се користи на серверу - упоређивањем разлика у верзијама - они могу утврдити и који безбедносни пропусти постоје у претходним верзијама софтвера. У принципу, проналажење безбедносних пропуста је могуће и без упоређивања верзија, пролажењем кроз цео изворни код (што је иначе и једини начин за проналажење пропуста у последњој верзији софтвера), али такав приступ захтева доста времена и напора.

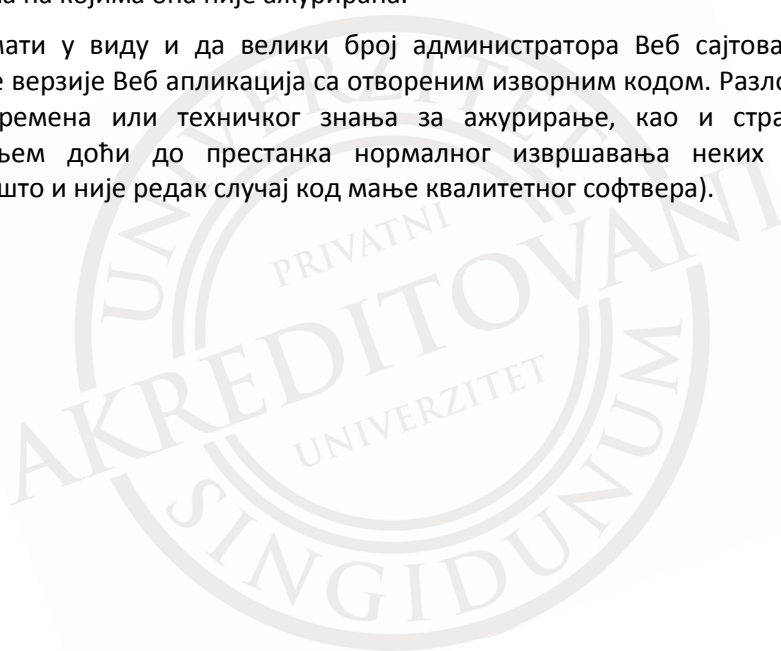
```
$ diff v1/login.php v2/login.php
53c53
< if ($ POST['user'] != 'administrator' and $ POST['password'] != 'abc123')
```

```
---
```

```
> if ($_POST['user'] != 'administrator' or $_POST['password'] != 'abc123')
```

У претходном листингу је дат пример проналажења безбедносног пропуста у новој верзији фиктивне Веб апликације упоређивањем измена у *PHP* фајловима. Као резултат поређења може се видети да је „тврдо“ кодован део за проверу корисничког имена и лозинке логички измењен, односно да је у претходној верзији рађено погрешно поређење које је дозвољавало приступ погађањем само једног параметра, корисничког имена или лозинке. На основу овог закључка нападач може несметано да приступи првој верзији апликације на серверима на којима она није ажурирана.

Треба имати у виду и да велики број администратора Веб сајтова користи застареле верзије Веб апликација са отвореним изворним кодом. Разлог томе је мањак времена или техничког знања за ажурирање, као и страх да ће ажурирањем доћи до престанка нормалног извршавања неких функција система (што и није редак случај код мање квалитетног софтвера).



## 8. Напади са циљем ускраћивања услуге

Напади са циљем ускраћивања услуге (енгл. *Denial of Service, DoS*) су усмерени на ускраћивање **доступности** одређене услуге путем напада на одређени критични ресурс који је потребан за њену испоруку. У основне критичне ресурсе за испоруку највећег дела услуга у рачунарским мрежама спадају:

1. комуникациони канал,
2. централни процесор и
3. радна меморија,

а критичне ресурсе чини и било шта друго што може довести до обустављања испоруке сервиса - број дозвољених паралелних веза, простор спољне меморије и друго. Уколико би рачунарски системи имали неограничене и увек доступне ресурсе, напади са циљем ускраћивања услуге не би били могући.

Нападаци могу нападати критичне ресурсе на два начина:

1. онемогућавањем/уништавањем и
2. исцрпљивањем.

За онемогућавање или уништавање ресурса потребно је да нападач има физички приступ ресурсу (на пример, пресецање мрежног кабла на слици 1) или да сам ресурс садржи пропуст у дизајну који се може искористити из даљине (на пример, пропусти са препуњавањем бафера код којих специјално форматизовани захтеви могу довести до блокаде оперативног система).



Слика 8-1 - Напад на доступност уништавањем критичног ресурса

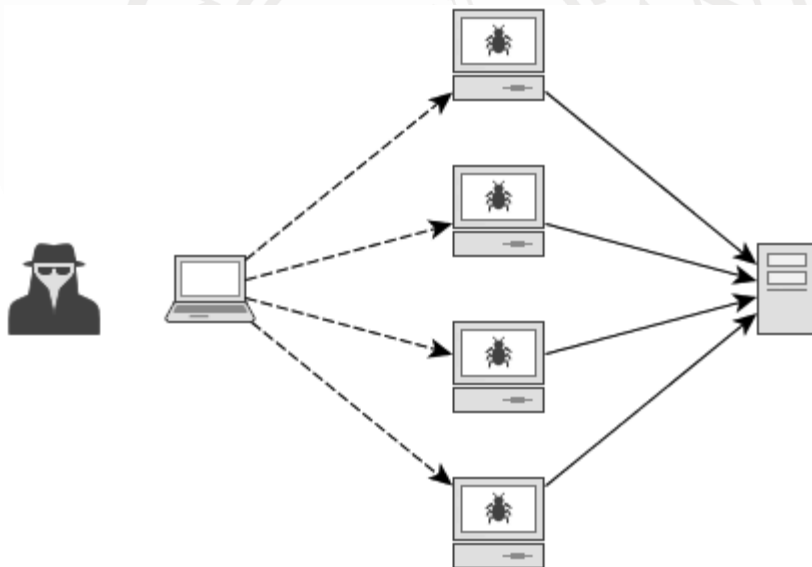
С обзиром на то да нападачи ретко имају физички приступ критичним ресурсима система, као и на то да су пропусти који се могу искористити ради онемогућавања одређеног ресурса веома ретки, нападачи са циљем ускраћивања услуге углавном користе технику **исцрпљивања** критичних ресурса. Исцрпљивање комуникационог канала своди се углавном на слање огромне количине података серверу, тако да није могуће проследити регуларне

захтеве. Ицрпљивање централног процесора и меморије сервера врши се углавном путем слања велике количине захтева или захтева који су тако формирани да изискују велико време за обраду.



Слика 8-2 - Ицрпљивање ресурса сервера огромним бројем захтева

У пракси, нападачи често нису у могућности да самостално исцрпу одређени ресурс, већ им је за то потребна помоћ већег броја рачунара. Напади у којима учествује већи број рачунара називају се **дистрибуирани** напади са циљем ускраћивања услуге (енгл. *Distributed Denial-of-Service, DDoS*). С обзиром на то да нападачи најчешће немају сопствене кластере рачунара за овакве нападе, за њих се користе регуларни рачунари инфицирани софтвером који омогућава нападачу да их контролише из даљине (тзв. зомби рачунари).

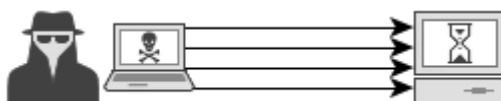


Слика 8-3 - Модел дистрибуираног напада на доступност сервиса (DDoS)

Број различитих начина за извођење напада са циљем ускраћивања услуге је велики а повремено се откривају и потпуно нови начини или нове варијанте већ познатих. У наставку су дати описи и начини извођења карактеристичних и популарних напада са циљем ускраћивања услуге.

## 8.1. Пинг поплаве

Пинг поплава (енгл. *ping flood*) је тип напада за чије се извршавање користи *ICMP* протокол. Напад се извршава тако што нападач великим интензитетом „пингује“ жртву, односно шаље велики број захтева типа *Echo request* у кратком временском периоду. Циљна последица овога је преоптерећивање комуникационог канала до жртве или њен евенуталан слом заузећем ресурса.



Слика 8.1-1 - Нападач шаље велики број „пинг“ захтева

У наставку је дат пример покушаја извршавања напада овог типа у Линукс оперативном систему:

```
bash-4.2$ ping -f 192.168.60.60
PING 192.168.60.60 (192.168.60.60) 56(84) bytes of data.
ping: cannot flood; minimal interval, allowed for user, is 200ms
```

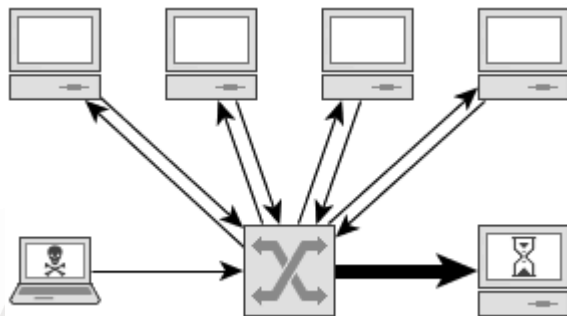
Пошто се систем брани од употребе за извршавање оваквих напада, најмање дозвољено чекање између слања два захтева је 200 милисекунди. Да би се то време скратило потребно је имати администраторске привилегије:

```
bash-4.2$ su - c "ping -f 192.168.60.60"
PING 192.168.60.60 (192.168.60.60) 56(84) bytes of data.
^C
--- 192.168.60.60 ping statistics ---
141250 packets transmitted, 141250 received, 0% packet loss, time 4211ms
rtt min/avg/max/mdev = 0.005/0.006/0.065/0.002 ms, ipg/ewma 0.029/0.007 ms
bash-4.2#
```

На примеру се може видети да је рачунар са кога се врши напад послао 141.250 пинг захтева за нешто више од 4 секунде. С обзиром да је рачунар коме су захтеви послати успешно одговорио на њих све, може се закључити да је напад делимично успешан, односно да није успело потпуно заузеће централног процесора жртве или комуникационог канала до ње. Са друге стране, множењем броја захтева са њиховом просечном величином може се закључити да оптерећење комуникационог канала током напада није занемарљиво.

## 8.2. Смурф напад

Смурф напад се заснива на могућности напада да „пингује“ (шаље *ICMP* захтеве) емисиону адресу мреже, односно све рачунаре у локалној мрежи, као и да у тим захтевима лажира изворишну адресу, тј. да као извор захтева наведе рачунар на који врши напад. Слањем таквог захтева нападач ће иницирати слање великог броја одговора рачунару који напада, са циљем загушења његовог комуникационог канала.



Слика 8.2-1 - Архитектура извођења смурф напада

У наставку је дат пример „пинговања“ емисионе адресе мреже, без лажирања изворишне адресе:

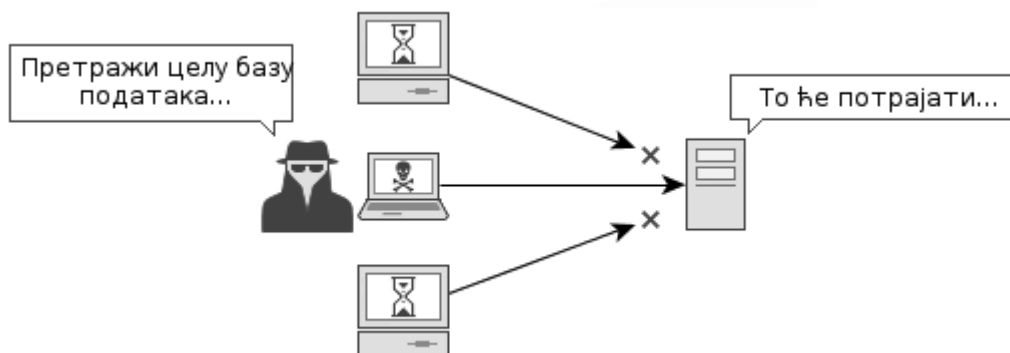
```
bash-4.2# ping -b 192.168.1.255
WARNING: pinging broadcast address
PING 192.168.1.255 (192.168.1.255) 56(84) bytes of data.
64 bytes from 192.168.1.62: icmp_req=1 ttl=64 time=0.248 ms
64 bytes from 192.168.1.199: icmp_req=1 ttl=64 time=0.273 ms (DUP!)
64 bytes from 192.168.1.17: icmp_req=1 ttl=255 time=0.481 ms (DUP!)
64 bytes from 192.168.1.198: icmp_req=1 ttl=255 time=0.626 ms (DUP!)
64 bytes from 192.168.1.198: icmp_req=1 ttl=255 time=0.612 ms (DUP!)
64 bytes from 192.168.1.198: icmp_req=1 ttl=255 time=0.691 ms (DUP!)
...
```

Лажирање изворишне (физичке и логичке) адресе може се извршити њеним једноставним задавањем административним алатима или заменом на нивоу подршке за протокол.



### 8.3. Напади на нивоу апликације

Напади на нивоу апликације углавном се користе за онеспособљавање удаљеног рачунара заузимањем његовог централног процесора и меморије, пре него комуникационог канала. На пример, нападач може пронаћи тип захтева за чије је решавање и слање одговора потребно доста процесорског времена - претрага базе Веб сајта - и послати велики број таквих паралелних захтева серверу. На тај начин је нападач успео да са минималном количином саобраћаја (такви *HTTP* захтеви могу бити и мањи од 1KB) привремено или у потпуности онеспособи Веб сервер.



Слика 8.3-1 - DoS напад на нивоу апликације

За одбрану од оваквих типова напада потребно је радити профилисање апликација које се извршавају на серверу и идентификовати захтеве за чије је извршавање потребна већа количина процесорског времена и меморије. У циљу одбране сервера потребно је такве захтеве ограничити (позивање само од ауторизованих корисника, временска учесталост и слично).

Напади на нивоу апликације су посебно значајни за разматрање када је у питању Веб окружење јер оно представља специфичну **отворену** платформу за дистрибуирано рачунарство. На Вебу је могуће једним захтевом минималне величине, креираним минималним ангажовањем ресурса на клијенту, иницирати огромно ангажовање ресурса и комуникационог канала серверске стране.

## 8.4. Напади спорим пријемом/слањем

### *Извод из описа алата SlowHTTPTest*

*Slowloris and Slow HTTP POST DoS attacks rely on the fact that the HTTP protocol, by design, requires requests to be completely received by the server before they are processed. If an HTTP request is not complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this creates a denial of service. This tool is sending partial HTTP requests, trying to get denial of service from target HTTP server.*

<http://code.google.com/p/slowhttpptest/>

Напади спорим пријемом или слањем података заснивају се на чињеници да сервери за сваку везу са клијентима резервишу одређену количину ресурса и држе је резервисану све док траје веза. Да би се одбранили од превеликог заузећа ресурса сервери често имају ограничен број паралелних активних веза, што нападачи такође могу искористити за онемогућавање регуларних захтева. Овакви напади се најчешће врше на нивоу транспортног протокола, заузећем сокета, или на нивоу апликативног протокола заснивајући се на његовим специфичностима.



Слика 8.4-1 - Прокси сервер у улози „храњења на кашичицу“

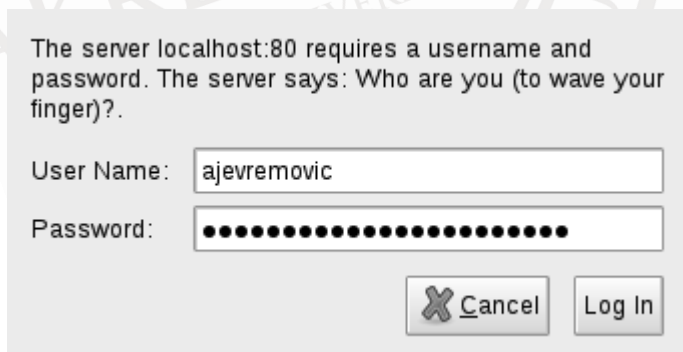
У циљу одбране сервера од напада спорим пријемом или слањем података развијен је модел тзв. „храњења на кашичицу“ (енгл. *spoon feeding*). У овом моделу (слика 1) користи се прокси сервер који је задужен да прими комплетан захтев од клијента пре прослеђивања серверу, односно да од сервера прими комплетан одговор а да га затим проследи клијенту. На тај начин се спречава заузимање ресурса на серверу - отворених сокета, веза, меморије и слично.

## 9. Шпијунирање рачунарских система

Један од честих типова напада у рачунарским мрежама је и шпијунирање рачунарских система и мрежних уређаја, односно њихових корисника. Код овог типа напада нападач првенствено крши тајност (приватност), а евентуално и веродостојност и доступност. На пример, нападач који инсталира *back-door* малициозни софтвер на рачунар жртве, првенствено има могућност да неовлашћено приступа различитим подацима на рачунару (на екстерној и у радној меморији, изгледу екрана, подацима са улазних уређаја и слично) а по потреби може и да мења податке, да их уклони или на други начин да онеспособи рачунар.

### 9.1. Снимање уноса са тастатуре

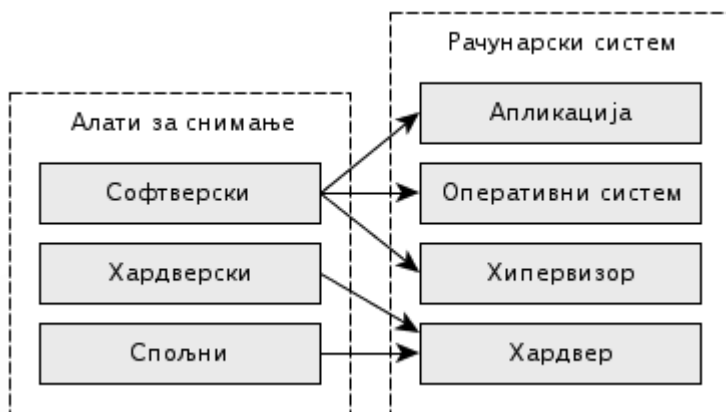
Подаци које корисници уносе путем тастатуре честа су мета нападача. Већину података корисници уносе у отвореном облику (репродукују се на екрану) док се за неке податке (на пример лозинке) користе специјализоване контроле које прикривају унос. У сваком случају, чак и када корисник уноси текст у отвореном облику, на пример приликом слања електронске поште, он најчешће не жели да ти подаци буду доступни лицима којима нису намењени. Међутим, постоји више начина да нападачи дођу до текста који је корисник унео путем тастатуре, било да је унос био у отвореном или скривеном облику.



Слика 9.1-1 - Унос лозинки са тастатуре је заштићен од визуалног надгледања

Један од основних алата за снимање текста унетог преко тастатуре су тзв. бележници тастера (енгл. *key logger*). Ови алати бележе све притиснуте тастере на тастатури у одговарајуће лог фајлове, а затим их на неки начин шаљу нападачу. Могу бити реализовани на више нивоа, у зависности од чега се деле

на софтверске, хардверске и спољне. Софтверски алати за снимање текста се реализују у виду независних апликација/сервиса, драјвера за тастатуру на нивоу оперативног система или на нивоу хипервизора код система за виртуализацију. У сваком случају, њихова улога је да пресретну све унете карактере и сачувају их у одговарајући фајл. Најчешће понашање је слање ухваћеног текста нападачу (путем мејла, *FTP* или *HTTP* протокола и слично) периодично или након одређеног броја снимљених уноса.



Слика 9.1-2 - Нивои рада алата за снимање уноса са тастатуре

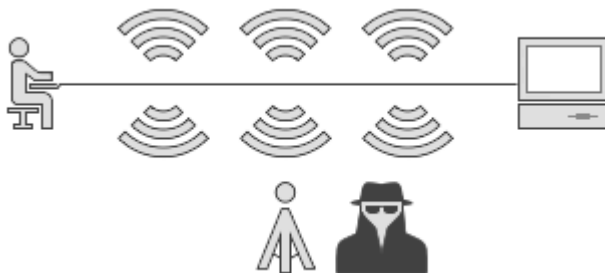
Хардверски алати за снимање уноса са тастатуре се физички постављају између тастатуре и рачунара. У питању су мали уређаји које је тешко визуално приметити (посебно што се постављају на задњу страну рачунара) и који имају сопствену меморију у коју снимају све унете карактере. Меморија на овим уређајима је углавном довољно велика да сачува све што би једна особа могла да откуца на тастатури током целог живота а овакви уређаји се понекад и наменски користе као копија у случају отказивања хард-диска рачунара.



Слика 9.1-3 - Додаци за снимање уноса са PS/2 и USB тастатура

Да би приступио снимљеним подацима на оваквим уређајима нападач најчешће мора физички да их преузме, с тим да постоје и уређаји који имају могућност да се повежу на бежичне приступне тачке у окружењу и пошаљу податке путем Интернета. Додатно, нападач може да приђе на неколико десетина метара уређају и постави своју приступну тачку коју ће уређај искористи за достављање података.

Спољни алати за снимање уноса са тастатуре не захтевају никакав контакт нападача са хардвером нити софтвером рачунара жртве, већ они за прикупљање података користе електромагнетно значење, звукове тастера и слично. На пример, истраживачи са универзитета Беркли су успели да снимањем и анализом звукова које је изазвало притискање тастера на тастатури успешно реконструишу 96 процената унетог текста<sup>7</sup>.



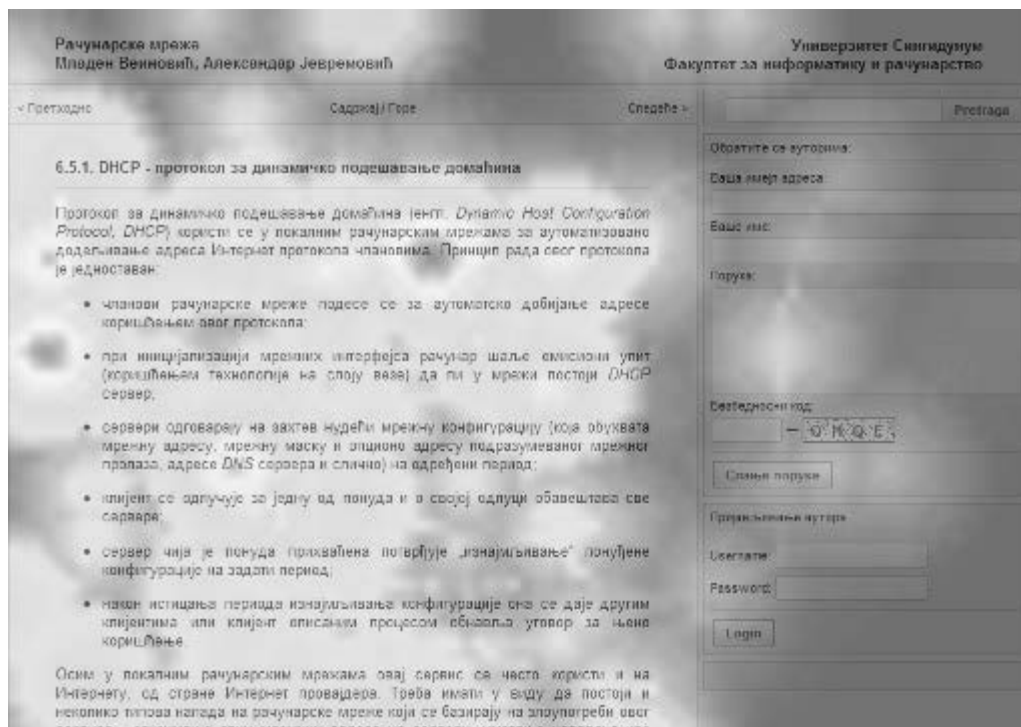
*Слика 9.1-4 - ЕМ зрачење кабла тастатуре може се злоупотребити*

Корисници често сматрају да су жичне тастатуре безбедније од бежичних. Ово је, међутим, погрешно пошто се подаци између бежичне тастатуре и пријемника на рачунару **шифрују**, док се пренос сигнала код жичних тастатура обавља без шифровања. Дакле, иако се сигнални бежичних тастатура могу примити на већој удаљености, њихово дешифровање је практично немогуће. Са друге стране, електромагнетно зрачење каблова жичних тастатура се може детектовати, а на основу њега утврдити и садржај и на удаљеностима од неколико десетина метара.

<sup>7</sup> [http://berkeley.edu/news/media/releases/2005/09/14\\_key.shtml](http://berkeley.edu/news/media/releases/2005/09/14_key.shtml)

## 9.2. Надгледање активности миша

Под надгледањем активности миша углавном се подразумева бележење  $x$  и  $y$  координата и тренутка у коме се курсор на њима налазио, као и да ли је том приликом био притиснут неки од тастера. Овакво надгледање је ефикасно због мале количине меморије потребне за бележење, као и могућности коришћења на Веб сајтовима од стране аутора.



Слика 9.2-1 - Мапа пажње корисника на основу кретања курсора миша

Иако кретање курсора миша само по себи не даје посебно вредну информацију нападачу, оно се може практично искористити у комбинацији са неким другим информацијама или уз додатну анализу. На пример, праћење курсора миша посетилаца Веб сајта може дизајнерима дати значајну информацију о томе колико је њихов дизајн ефикасан односно шта треба унапредити<sup>8</sup>. Или, постоји више радова у којима се кретање курсора миша користи за аутентификацију корисника рачунара.

8 Александар Јевремовић, Саша Адамовић, Младен Веиновић, „Праћење курсора миша посетилаца као евалуација ефикасности дизајна Веб сајта“, 57. конференција за електронику, телекомуникације, рачунарство, аутоматику и нуклеарну технику, Етран, јун 2013.

### 9.3. Шпијунски софтвер

Шпијунски софтвер (енгл. *Spyware*) је посебан тип софтвера чија је основна или споредна намена да нападачу дугорочно омогући приступ одређеним подацима и функцијама на рачунару на коме је инсталиран. Подаци које шпијунски софтвер шаље нападачу крећу се од периодичних извештаја о коришћењу неке апликације или уређаја, па до потпуног приступа спољној и радној меморији, са функцијом читања, измене и уклањања. Када је у питању приступ функцијама рачунара, шпијунски софтвер овог типа углавном омогућава коришћење свих функција које су доступне и регуларном кориснику.

Израстањем Веба у нову платформу за извршавање софтвера и комуникацију појавили су се и различити облици шпијунског софтвера који функционишу у том окружењу. На пример, Веб колачићи и различити сервиси који се интегришу у Веб сајтове омогућавају ауторима да прате кретање корисника кроз Веб и врше његово профилисање. Додатно, мање искусни корисници често прихватају понуде сајтова да у њихове Веб читаче инсталирају додатне траке са алатима, као и да промене подразумеване сајтове за претрагу. Не треба потцењивати ни везу између савремених Веб читача и он-лајн сервиса њихових аутора. На пример, Гугл Хром Веб читач подразумевано шаље серверу текст који корисник уноси у поље за адресу како би предложио термин за претрагу.

Аутори регуларног корисничког софтвера понекад намерно постављају недокументоване могућности које им омогућавају да касније приступе рачунарима на којима је такав софтвер инсталиран. Чак, одређени новинари указују на то да владе неких земаља сарађују са највећим произвођачима софтвера са циљем да се њиховим безбедносним службама омогући неовлашћен приступ рачунарима корисника<sup>9 10</sup>.

Шпијунски софтвер подразумевано јесте злонамеран али га не треба мешати са вирусима или црвима, већ углавном спада у групу „тројанских коња“. Такође, понекад се регуларно користи у организацијама да би менаџменту и администраторима омогућио увид у то шта запослени раде на својим рачунарима.

---

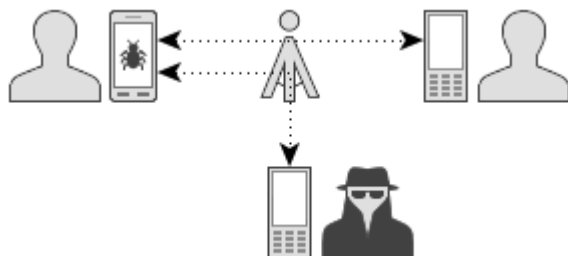
9 <http://www.wnd.com/2013/06/nsa-has-total-access-via-microsoft-windows/>

10 <http://www.computerworld.com/article/2521809/government-it/nsa-helped-with-windows-7-development.html>



## 9.4. Шпијунирање мобилних телефона

Све чешће жртве шпијунирања постају корисници мобилних телефона. Савремени мобилни телефони имају велике процесорске, меморијске и комуникационе капацитете, као и могућност лаке инсталације новог софтвера.



Слика 9.4-1 - Прислушкивање разговора путем малициозног софтвера

У складу са тим, нападачи могу мобилним телефонима да додају софтверске пакете који омогућавају детаљно надгледање дешавања на мобилном телефону (садржај активних разговора, изгледа екрана и слично), приступ подацима у меморији (поруке, фајлови...), географској локацији уређаја и улазним уређајима (звучник, камера).



Слика 9.4-2 - Препознавање везе која није шифрована на Криптофону

Још један начин да нападачи прислушкују разговоре је постављање лажних приступних станица са којима ће се телефон жртве повезати. На пример, истраживачи запослени у *ESD America* компанији су у августу 2014. године на територији САД коришћењем свог заштићеног телефона открили 500 лажних приступних тачака за мобилну телефонију<sup>11</sup>.

11 <http://finance.yahoo.com/news/mysterious-fake-cellphone-towers-intercepting-162645809.html>



## 10. Анализа безбедности система и мрежа

У три основна метода за процену безбедности одређеног рачунарског система или рачунарске мреже спадају безбедносна провера, испитивање и интервјуисање. **Безбедносном провером** система упоређује се стварно понашање система током напада са очекиваним. Овом методом се директно анализирају циљни системи у мрежи методама скенирања и пробојног тестирања са циљем да се добију информације о евентуалним безбедносним пропустима. Иако ова метода пружа квалитетне и непосредне информације о безбедности система, њена основна мана је утицај на системе који се анализирају (утицај се, у зависности од коришћених техника, може манифестовати падом перформанси система, али и његовим потпуним сломом).

Постоји више различитих типова безбедносне провере. Са аспекта рачунарске мреже провера може бити **унутрашња и спољашња**. Код унутрашње провере испитивање се врши из унутрашњости рачунарске мреже, са претпоставком привилегија регуларног корисника. Насупрот томе, код спољне провере испитивање се врши споља, на пример са Интернет мреже. У том случају се испитује и ефикасност система који рачунарску мрежу организације штите од спољних напада.

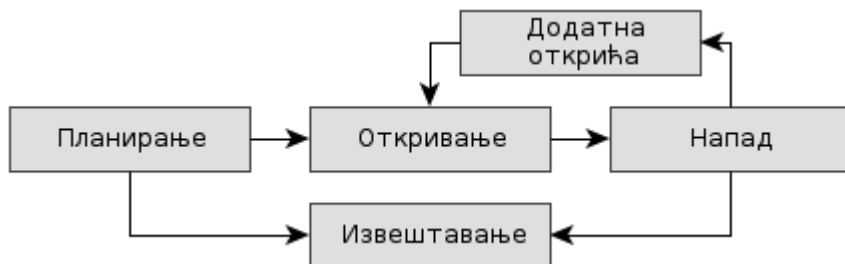
Са аспекта тајности безбедносне провере, она може бити **јавна и тајна**. Код јавне провере је ИТ особље организације упознато са тим да је у току безбедносна провера рачунарске мреже и система које оно одржава. Насупрот томе, тајна безбедносна провера одвија се од стране спољног тима којег је ангажовало руководство организације, а без знања ИТ особља организације.

**Испитивањем безбедности** анализирају се релевантни чиниоци безбедности - врши се ревизија докумената као што су безбедносне полисе, процедуре, планови, захтеви, стандардне оперативне процедуре, дијаграми са архитектуром, инжењерска документација, системска конфигурација, лог фајлови и друго. На основу ових активности добија се увид у ниво безбедности система који се испитује. Предност овог метода у односу на тестирање огледа се у томе што он углавном нема утицаја на нормално функционисање система већ се цео процес одвија паралелно.

Метода **интервјуисања** обухвата вођење разговора са појединцима и одељењима организације са циљем основног прикупљања релевантних безбедносних информација. Подаци добијени овом методом често су основа за примену осталих метода.

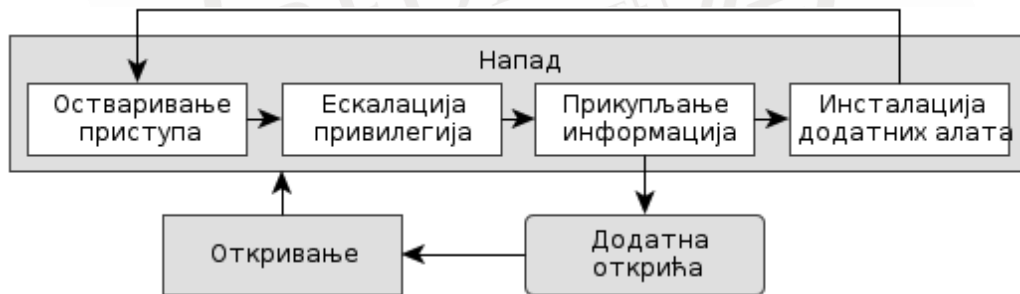
## 10.1. Провера пробојности

Провера пробојности (енгл. *Penetration Testing*) је врста напада на рачунарске мреже који има за циљ да провери у каквом су стању инсталирани безбедносни системи, односно да ли су у стању да успешно обављају своју улогу или их је могуће пробити. Резултат ове провере може бити потврда ефикасности инсталираних безбедносних система или листа пропуста које треба отклонити.



Слика 10.1-1 - Стања код провере пробојности

Провере пробојности се не разликују много од класичних напада на рачунарске мреже, с тим што њихов циљ није да нанесу штету систему који се проверава, а усто и да се сви резултати напада евидентирају и на крају стављају у форму извештаја (слика 1). Иначе, процедура напада (слика 2) и код провере пробојности се поклапа са процедуром коју би стваран нападач примењивао.



Слика 10.1-2 - Кораци у оквиру фазе напада

За проверу пробојности користе се различити специјализовани алати. Један од најпопуларнијих алата овог типа је Кали<sup>12</sup> дистрибуција Линукса, наследник популарног БекТрек пројекта<sup>13</sup>.

12 <http://www.kali.org/>

13 <http://www.backtrack-linux.org/>

## 10.2. Концепт ћупа са медом

Командант снага НАТО-а у Европи и борбених операција у Југославији генерал Весли Кларк дичио се да је уништио 93 тенка и 153 оклопна транспортера, као и да је елиминисао неколико оклопних јединица Војске Југославије. Исмеване су српске тврдње да је на Косову изгубљено свега 13 тенкова и шест транспортера, и то већином због герилских напада ОВК, а не због успешности америчких ловаца. Међутим, када је рат стао и када је започело повлачење српске војске с Косова, изронило је „најмање 220 тенкова и више од 300 оклопних транспортера“, извештавала је агенција АФП 1999. године. На целом Косову, према извештају самог НАТО-а, нађени су уништени остаци само 26 тенкова или самоходних артиљеријских оруђа. Односно, највећим делом макете – лажни циљеви за авијацију НАТО-а.

**Кобац против НАТО-а, Политика, 7. април 2014.**

Концепт ћупа са медом (енгл. *Honeypot*) подразумева постављање замке нападачу којом би се, са једне стране, онемогућило његово деструктивно деловање и, са друге стране, омогућило праћење његовог понашања. Један модел реализације овог концепта приказан је на слици 1.



Слика 10.2-1 - Принцип рада концепта „ћуп са медом“

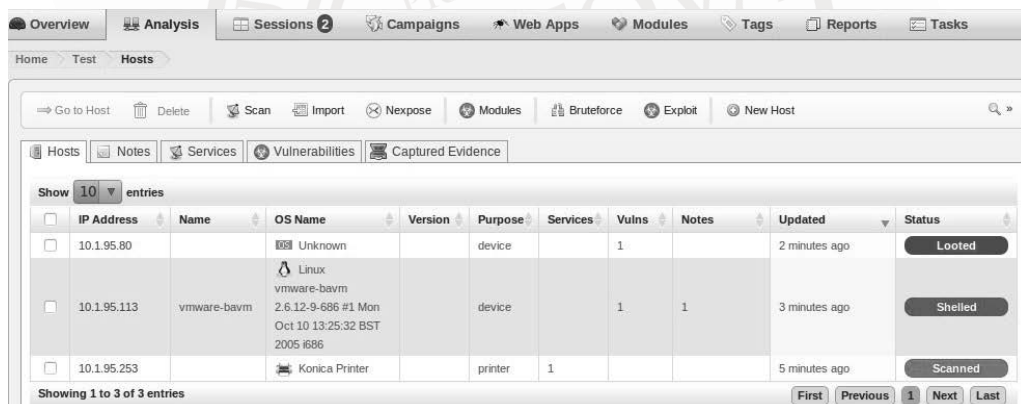
Идеја иза реализације концепта са слике 1 је да се нападач преусмери у лажно окружење, односно у окружење које је симулација реалног окружења али не садржи осетљиве податке, нити оштећење ресурса у њему представља било

какву штету организацији. Овакво окружење, на пример, може бити реализовано у виду скупа виртуалних машина и мрежа које се извршавају на једном или неколико сервера.

Додатак у симулираном окружењу је компонента која прати и бележи активности нападача, тако да се накнадном анализом може утврдити шта је био циљ напада. Такође, с обзиром да симулирано окружење представља реплику реалног окружења, пробој безбедносних система у њему представља и аларм за унапређивање њихових пандана у стварном окружењу.

### 10.3. Алати за откривање пропуста

Да би администратори имали стваран увид у безбедносно стање својих рачунарских мрежа и рачунарских система у њима, они морају редовно проверавати да ли и које рањивости постоје у њима. У потенцијалне рањивости спадају оне које претходно нису откривене, али и оне које су се јавиле као последица различитих измена у систему (додавање нових рачунара, инсталација софтвера, промена конфигурационих параметара и слично) или као последица откривања нових пропуста од стране безбедносних експерата.



Слика 10.3-1 - Изглед корисничког интерфејса Метасплит алата

Ручно проверавање великог броја система и мрежних уређаја је огроман посао који, осим што захтева велико ангажовање ИТ особља, може предуго трајати тако да евентуално рањиви системи дуго остају незаштићени. Да би се одговорило на овај проблем развијени су аутоматизовани системи за проверу рањивости рачунарских система и мрежа. Ови алати имају могућност да у минимално потребном времену провере отпорност великог броја рачунарских система и мрежних уређаја на све рањивости које су описане у њиховој бази података.

## 11. Литература

1. Karen Scarfone Murugiah Souppaya, Amanda Cody, Angela Orebaugh, „Technical Guide to Information Security Testing and Assessment“, National Institute of Standards and Technology, 2008.
2. Sheila Frankel, Karen Kent, Ryan Lewkowski , Angela D. Orebaugh, Ronald W. Ritchey, Steven R. Sharma, „Guide to IPsec VPNs“, National Institute of Standards and Technology, 2005.
3. Sheila Frankel, Paul Hoffman, Angela Orebaugh, Richard park, „Guide to SSL VPNs“, National Institute of Standards and Technology, 2008.
4. Murugiah Souppaya, Karen Scarfone, „Guidelines for Securing Wireless Local Area Networks (WLANs)“, National Institute of Standards and Technology, 2012.
5. Karen Scarfone, Derrick Dicoi, Matthew Sexton, Cyrus Tibbs, „Guide to Securing Legacy IEEE 802.11 Wireless Networks“, National Institute of Standards and Technology, 2008.
6. Karen Scarfone, Peter Mell, „Guide to Intrusion Detection and Prevention Systems (IDPS)“, National Institute of Standards and Technology, 2007.
7. Karen Scarfone, Murugiah Souppaya, Matt Sexton, „Guide to Storage Encryption Technologies for End User Devices“, National Institute of Standards and Technology, 2007.
8. Murugiah Souppaya, Karen Scarfone, „Guidelines for Managing the Security of Mobile Devices in the Enterprise“, National Institute of Standards and Technology, 2007.
9. Kevin Mitnick, The Art of Deception: Controlling the Human Element of Security, 2002, ISBN 0-471-23712-4
10. Bluetooth Security, Systems and Network Analysis Center Information Assurance Directorate, National Security Agency
11. Младен Веиновић, Александар Јевремовић, „Рачунарске мреже“, ISBN 978-86-7912-368-8; COVBIS.SR-ID, Универзитет Сингидунум, 2011;
12. Александар Јевремовић, Младен Веиновић, „Интернет технологије“, ISBN: 978-86-7912-505-7, Универзитет Сингидунум, 2013;
13. Александар Јевремовић, Саша Адамовић, Младен Веиновић, „Праћење курсора миша посетилаца као евалуација ефикасности дизајна веб сајта“, Зборник радова 57. конференције Етран, јун 2013.

CIP - Каталогизација у публикацији - Народна библиотека Србије, Београд  
004.7.056.5(075.8)

ЗАШТИТА у рачунарским мрежама / Александар Јевремовић ... [и др.]. - 2.  
изд. - Београд : Универзитет Сингидунум, 2018 (Београд : Калиграф). - 126  
стр. : илустр. ; 24 cm

На врху насл. стр.: Факултет за информатику и рачунарство. - Тираж 750. -  
Библиографија: стр. 126.

ISBN 978-86-7912-565-1

1. Јевремовић, Александар, 1983- [аутор]

а) Рачунарске мреже - Заштита

COBISS.SR-ID 265928716

© 2018.

Sva prava zadržana. Nijedan deo ove publikacije ne može biti reprodukovан u bilo kom vidu i putem  
bilo kog medija, u delovima ili celini bez prethodne pismene saglasnosti izdavača.



Александар Јевремовић  
Младен Веиновић  
Марко Шарац  
Горан Шимић

# ЗАШТИТА У РАЧУНАРСКИМ МРЕЖАМА

Основна идеја овог уџбеника је да студентима пружи квалитетну основу за проучавање домена заштите у савременим рачунарским мрежама и оспособи их за суочавање са практичним изазовима из ове области. У уџбеник су уграђена општа знања из ове области, али и искуства аутора стечена дугогодишњим бављењем овом облашћу у пракси.