# Zastrupljanje protokolov za razreševanje imen na lokalnih omrežjih

Urban Dopudja, Matevž Pesek
Univerza v Ljubljani, Fakulteta za Računalništvo in Informatiko, Večna pot 113, 1000 Ljubljana
ud74172@student.uni-lj.si, matevz.pesek@fri.uni-lj.si

## Izvleček

V kontekstu povezovanja različnih informacijskih sistemov je razreševanje domenskih naslovov ključni proces identifikacije deležnikov v širšem okolju IT infrastrukture, ki ob pomanjkljivi konfiguraciji lahko predstavlja tveganje za zlorabo s strani napadalcev. Zaradi rastoče kompleksnosti infrastrukture se količina takšnih vektorjev napada na informacijske sisteme v zadnjem času povečuje. V pričujočem članku se poglobimo v delovanje protokolov za več vrstno oddajanje (angl. multicast) razreševanje imen v omrežjih ter njihovo potencialno zlorabo. Na tipičnih primerih pokažemo načine izrabe različnih orodij, s katerimi lahko relativno enostavno izvedemo takšne napade. Skladno z demonstracijo napadov nato prikažemo različne tehnike, s katerimi je mogoče prikazane napade zadostno omejiti.

**Ključne besede:** DNS zastrupljanje, LLMNR, Omrežna varnost, Šifrirni algoritmi

## LOCAL NETWORK NAME RESOLUTION POISONING

## Abstract

In the context of connecting different information systems, the resolution of domain addresses is a key process of identification of stakeholders in the wider environment of the IT infrastructure, which in the case of faulty configuration can pose a risk of abuse by attackers. Due to the growing complexity of the infrastructure, the amount of such attack vectors on information systems has been increasing recently. In this article, we delve deeper into the operation of protocols for the multicast name resolution in networks and their potential abuse. On typical examples, we show ways of using various tools that can be used to carry out such attacks relatively easily. According to the demonstration of the attacks, we then show various mitigations of the displayed attacks, with which the displayed attacks can be sufficiently limited.

**Key words:** DNS poisoning, LLMNR, Network security, Hash algorithms

## 1. UVOD

V zadnjih dveh letih smo bili priča napadom na večje ukrajinske organizacije pred začetkom ruske invazije leta 2022 [4, 13]. Ti napadi so razkrili načine razširjenega zasega zgoščenih poverilnic z namenom dešifriranja le-teh in njihove uporabe v obsežnih napadalskih kampanjah na ukrajinsko internetno in komunikacijsko infrastrukturo. Ti napadi so se izkazali za efektivne, hkrati pa je odziv nanje pokazal več enostavnih prijemov, ki so takšne napade v nadaljevanju vojne odbili ali vsaj efektivno omejili. V okoljih Windows je izkoriščanje ranljivosti, zlasti s tehnikami kot sta LLMNR in NBT-NS zastrupitev, pogosto [11]. Te tehnike se pogosto uporabljajo v omrežjih za preusmerjanje prometa in krajo poverilnic, kar ogroža varnost celotnega sistema. Manipulacija proce-

sov DNS razreševanja je ključni del teh napadov, saj omogoča napadalcem, da prestrežejo in preusmerijo omrežni promet [15]. Avtomatizacija takih napadov, ki jo omogočajo orodja, kot sta Metasploit ali Responder, povečuje tveganje za varnost predvsem sistemov v okolju Windows domen [2]. Ta orodja olajšajo napadalcem izvajanje kompleksnih napadov, ki bi sicer zahtevali več tehničnega znanja in izkušenj.

V kontekstu potencialnih ranljivosti v praksi se je nadzor nad avtentikacijskimi procesi izkazal kot ključen. Freimanis idr. so analizirali vpliv avtentikacijskih metod na splošno varnost računalniških sistemov večjih organizacij [6]. Njihove ugotovitve, ki temeljijo na več izvedenih penetracijskih testih, kažejo, da je strikten nadzor nad podprtimi avtentikacijskimi algoritmi ključnega pomena pri zagotavljanju zaupnosti in celovitosti računalniških sistemov [17]. Zlasti je to pomembno pri preprečevanju t.i. "pass-the-hash" in "pass-the-ticket" zlorab, ki so med najpogostejšimi napadi na Windows sisteme [14]. Te zlorabe omogočajo napadalcem pridobivanje dostopa do omrežnih virov brez dejanske pridobitve gesel, kar dodatno poudarja potrebo po strogih varnostnih ukrepih.

Naš cilj je poglobljeno raziskati zlorabo protokolov za razreševanje imen z večvrstnim oddajanjem, (angl. multicast) z namenom zajetja poverilnic kot delu kompleksnejšega napada na IT infrastrukturo. Demonstracija in analiza takšnih napadov nam omogoča vzpostavitev varnejše in bolj odporne infrastrukture proti tovrstnim napadom. V nadaljevanju članka najprej predstavimo tehnične potrebe za delovanje protokolov, ki so potencialno ranljivi — Link-Local Multicast Name Resolution (LLMNR), NetBIOS Name Service (NBT-NS) in Multicast DNS (mDNS). Nato obravnavamo načine zlorabe teh protokolov z različnimi orodji ter predstavimo lastno okolje za avtomatizacijo tovrstnih napadov. članek zaključimo s pregledom obrambnih mehanizmov za zaščito pred takšnimi napadi.

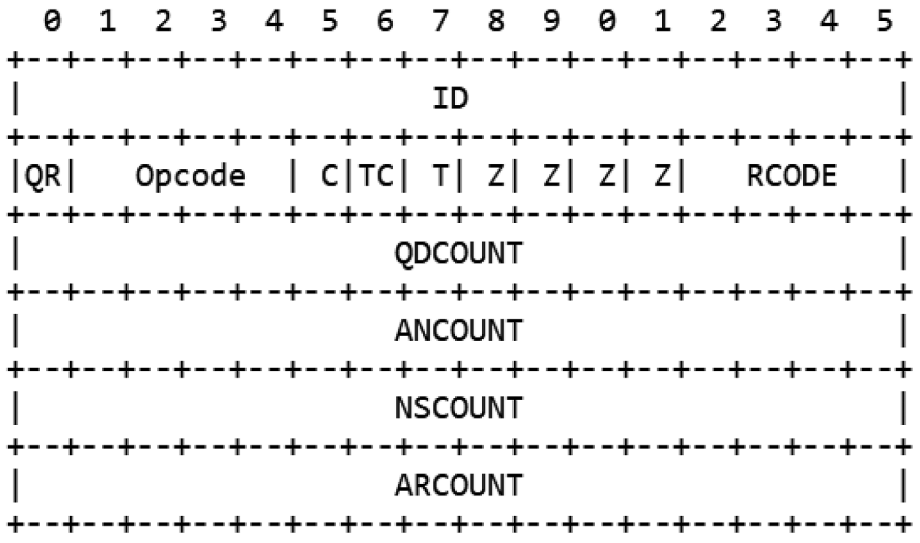
## 2 TEHNIČNE SPECIFIKACIJE RAZREŠEVALNIH PROTOKOLOV

### 2.1 Link-Local Multicast Name Resolution

Link-Local Multicast Name Resolution (LLMNR) [1] je protokol druge plasti ISO/OSI modela, ki ponuja alternativo (ali t.i. »fallback«) DNS-u za razreševanje imen v lokalnih omrežjih. LLMNR deluje decentralizirano po principu poizvedb večvrstnega oddajanja znotraj lokalnega omrežja, s katerim zagotavlja učinkovito razreševanje imen brez potrebe po centralizirani DNS infrastrukturi, vendar pa je zaradi njegove narave lahko zlorabljen v okviru kibernetskih napadov.

LLMNR deluje na vratih 5355, pri čemer so IPv4 poizvedbe poslane na naslov za večvrstno oddajanje 224.0.0.252, IPv6 poizvedbe pa na naslov FF02::1:3. V kontekstu LLMNR so gostitelji (angl. hosts) običajno konfigurirani tako kot pošiljatelji kot tudi odzivniki, lahko pa so tudi izključno pošiljatelji (vendar ne obratno), saj mora vsak gostitelj, konfiguriran kot odzivnik, delovati tudi kot pošiljatelj z namenom zagotavljanja edinstvenosti imen.

Postopek razreševanja se odvija v zaporedju, kjer pošiljatelj sproži poizvedbo, na katero nato odgovori odzivnik. Odgovor se pošlje nazaj pošiljatelju kot večvrstni ali enovrstni UDP paket, odvisno od narave poizvedbe. Format LLMNR paketa (poizvedba ter odgovor) temelji na formatu DNS-a, kateri je definiran v standardu RFC1035 - razdelek 4. Standard RFC predvideva pošiljanje UDP paketov znotraj dovoljenih velikosti z namenom izogibanja drobljenju (oz. fragmentaciji) - priporočljivo do 512 oktetov. Implementacija protokola pa lahko sprejme UDP pakete do velikosti največje enote prenosa (angl. maximum transmission unit - MTU) ali 9194 oktetov – velikost Ethernet jumbo 9 KB okvirja, z odštetimi 22 okteti za glavo ter oznaki navideznega omrežja (VLAN) in CRC kode.




Slika 1: **Format zaglavja paketa[1]**

- ID: 16-bitni identifikator, dodeljen poizvedbam, ki pošiljateljem omogoča ujemanje odgovorov. Zaradi varnosti je nastavljen na psevdonaključno vrednost.
- QR: 1-bitno polje, ki označuje, ali je sporočilo odgovor (set) ali poizvedba (clear).
- OPCODE: 4-bitno polje, ki določa vrsto poizvedbe.
- C: Označuje konflikt v poizvedbi ali edinstvenost imena v odgovoru.
- TC: Določa prirezovanje (truncation) zaradi omejitev dolžine. Če je nastavljeno v odgovoru, mora pošiljatelj ponovno poslati poizvedbo prek TCP.
- T: Označuje pogojni odgovor, če oseba, ki je odgovorila, ni preverila edinstvenosti imena.
- Z: Rezervirano za prihodnjo uporabo, trenutno nastavljeno na 0.
- RCODE: Koda odziva, nastavljena v odgovorih. V poizvedbah mora biti nič. RCODE, ki ni ničel, v odgovorih za večvrstno oddajanje vodi do poizvedbe TCP.
- QDCOUNT, ANCOUNT, NSCOUNT, ARCOUNT: 16-bitna nepredznačena (unsigned) števila, ki določajo število vnosov v različnih delih sporočila. Upoštevati mora določena pravila, da se prepreči tiho zavrženje s strani pošiljateljev ali prejemnikov.

## 2.2 Multicast DNS

Protokol mDNS[3] deluje na podoben princip in služi podoben namen kot LLMNR, le da je večinoma uporabljen v energetsko omejenih napravah/vgrajenih ter operacijskih sistemih kot so Linux ter MacOS, za razliko od LLMNR, ki je primarno uporabljen v Windowsu. mDNS se razlikuje tudi v tem, da IPv4 poizvedbe sprejema na naslov 224.0.0.251, IPv6 poizvedbe pa na FF02::fb. Kljub vsemu, novejše Windows različice za razreševanje pogosto uporabijo kar oba protokola, kot je razvidno iz spodnje slike prometa, zajetega s programom Wireshark, kjer smo izvedli poizvedbo po imenu (angl. hostname) “abc”.

| | | | |
|---|---|---|---|
| 10.0.2.15 | 224.0.0.251 | MDNS | 69 Standard query 0x0000 AAAA abc.local, "QM" question |
| 10.0.2.4 | 10.0.2.15 | MDNS | 79 Standard query response 0x0000 A 10.0.2.4 |
| fe80::d584:cf84:4b6… | ff02::fb | MDNS | 89 Standard query 0x0000 AAAA abc.local, "QM" question |
| fe80::d584:cf84:4b6… | ff02::1:3 | LLMNR | 83 Standard query 0xbf24 A abc |
| 10.0.2.15 | 224.0.0.252 | LLMNR | 63 Standard query 0xbf24 A abc |

Slika 2: **LLMNR in mDNS poizvedba ter odgovor**

Na sliki prikazujemo, kako naprava pošlje mDNS poizvedbo prek IPv4 ter IPv6, nato pa isto stori še z uporabo protokola LLMNR. V tem primeru, že na prvo povpraševanje odgovori napadalec.

### 2.3 NetBIOS Name Service

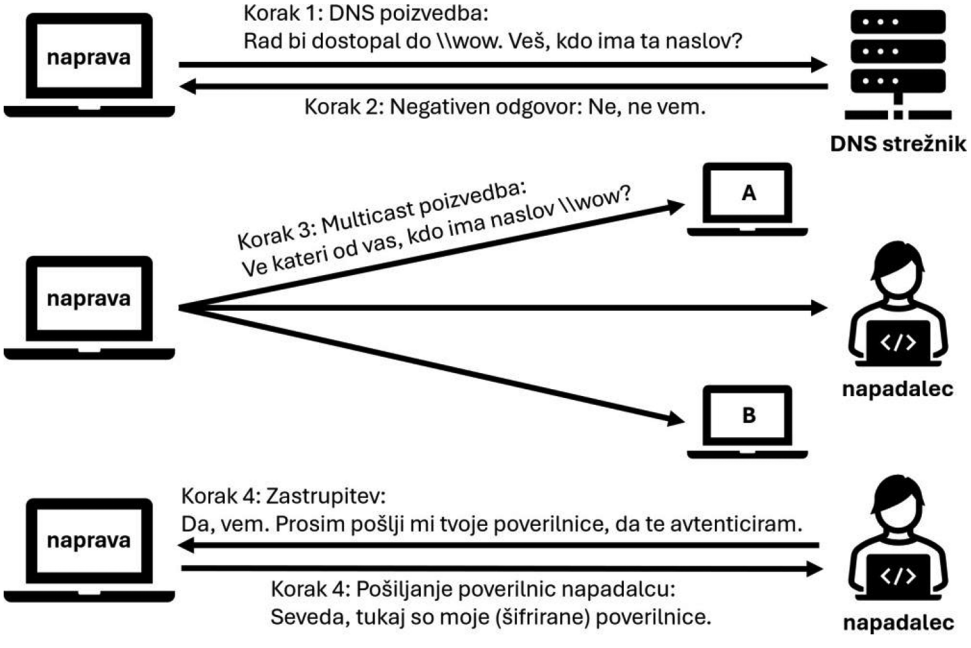
NBT-NS[10] je starejši protokol za razreševanje imen, ki je bil predvsem uporabljen v starejših Windows okoljih. Uporablja se za razrešitev NetBIOS imen v IP-naslove. Deluje preko UDP in uporablja vrata 137.

Za razliko od mDNS in LLMNR, ki sta bolj generična in delujeta na različnih operacijskih sistemih, je NBT-NS specifičen za okolja Windows. Deluje skupaj z drugimi protokoli, povezanimi z NetBIOS, kot je NetBIOS preko TCP/IP (NBT), in se uporablja predvsem zaradi povratne združljivosti v sodobnih Windows domenah/omrežjih. NBT-NS pravzaprav ne razrešuje domenska imena v IP-naslove, temveč v NetBIOS imena, ki so uporabljena za prepoznavanje Windows naprav ter storitev. Za razliko od ostalih dveh protokolov, NBT-NS podpira samo IPv4.

## 3 ZLORABA

V tem sklopu bomo najprej predstavili teoretično zlorabo omenjenih protokolov, nato pa prikazali praktični primer z uporabo orodja Responder. Kot potreben pogoj za izvedbo napada mora predhodno napadalec imeti dostop do omrežja.

Naprava, ki želi dostopati do nekega domenskega imena, katerega naslova še ne pozna, sprva pošlje poizvedbo na DNS strežnik. Če slednji ne poseduje ustreznega zapisa, naprava nato poplavi omrežje z vprašanjem po tem naslovu. Ta korak procesa predstavlja ključno ranljivost, pri kateri se lahko v komunikacijo vrinemo kot napadalec, ki na lokalnem omrežju posluša za tovrstnimi poizvedbami in ob prejeti poizvedbi sestavi "zastrupljen" odgovor, v katerem predstavlja sebe (ali drugo napravo) pod iskanim imenom. Poizvedujoča naprava nato od tarče zahteva poverilnice ali druge informacije, katere dobimo v obliki zgoščenih vrednosti ali pa celo kot golo besedilo (plain text).




Slika 3: **Diagram poteka "zastrupitve"**

### 3.1 Orodje RESPONDER

Za praktično izvedbo napadov smo uporabili orodje *Responder* [8]. Prikazali bomo šest tipov napadov z omenjenim orodjem v praktični obliki na simulacijskem okolju. Za simulacijsko okolje smo postavili tri virtualne naprave in jih povezali na isto virtualno omrežje (NAT network). Omenjene tri naprave so:

- Naprava Windows 11 s prizetimi nastavitvami — uporabljena kot tarča pri večini napadov,
- Naprava Kali Linux z orodjem Responder — uporabljena kot napadalec,
- Naprava Debian — uporabljena kot podpora napravam - na primer za Samba strežnik.

### 3.2 Zastrupljanje z uporabo SMB strežnika

Ko tarča poskusi dostopati do SMB strežnika, pošlje poizvedbo DNS strežniku, ki odgovori, da nima zapisa za to domensko ime. Tarča nato pošlje večvrstno poizvedbo po omrežju, ki jo lahko prestrežemo z uporabo orodja Responder *responder -I eth0* .

Responder tarči pošlje paket "Standard Query Response 0x0000 A", v katerem tarčo pozove, naj se avtenticira. Na Windows napravi se pojavi vpisno okno, kamor uporabnik vpiše poverilnice, in se pošljejo napadalcu. Napadalec prejme izpis v formatu:

```
Poisoned answer sent to <ip> for name
abc.local NTLMv2-SSP Client: <ip>
NTLMv2-SSP Username: <hostname>/<username>
NTLMv2-SSP Hash:
<username>::<hostname>:<hash>
```

### 3.3 Zastrupljanje z uporabo protokola WPAD

WPAD (angl. Web Proxy Auto-Discovery Protocol) je mehanizem za konfiguracijo omrežja, ki se uporablja predvsem v večjih organizacijah za samodejno odkrivanje posredniških (angl. proxy) strežnikov. Z WPAD protokolom lahko odjemalec poišče konfiguracijsko datoteko posrednika, ki se običajno nahaja na spletnem strežniku v lokalnem omrežju. Konfiguracijska datoteka vsebuje navodila o odjemalčevem dostopu do interneta, vključno s tem, kateri posredniški strežnik naj uporabi in katera vrsta prometa naj bo usmerjena

preko slednjega. Če uporabnik poskuša dostopati do neveljavnega URL naslova (npr. skozi brskalnik),

DNS strežnik ne bo vseboval imel zapisa za iskano stran. Brskalnik bo, če ima vklopljeno funkcionalnost "automatic configuration detection", poslal večvrstno povpraševanje po omrežju, v katerem povprašuje po WPAD strežniku.

To funkcionalnost brskalnika lahko zlorabimo z uporabo "-w" zastavice pri zagonu programa Responder, katera vzpostavi zlonamerni WPAD strežnik. Ko uporabnik zahteva konfiguracijsko datoteko, jo le-ta pozove za poverilnice. Tako zopet pridobimo NTLMv2(/SSP) zgoščene vrednosti poverilnic, v nekaterih primerih pa celo v golem tekstu.

### 3.4 Prisilna uporaba osnovne avtentikacije

Osnovna (angl. basic) avtentikacija uporablja golo besedilo za pošiljanje poverilnic — to je nešifrirana oblika, ki jo je mogoče neposredno prebrati. Dešifriranje lahko dolgotrajen in zahteven proces, zato bi bilo z vidika porabe časa in procesorske moči najlažje, da poverilnice izmenjujemo v goli obliki, kar pa predstavlja veliko grožnjo varnosti. Za ta primer nadaljujemo s prej opisanim napadom z strežnikom WPAD, ki mu v konfiguraciji orodja Responder dodamo zastavico "-b", ki prisili uporabnike v uporabo osnovne avtentikacije. Na tem mestu bi radi izpostavili, da to deluje v relativno redkih primerih. V primeru WPAD napada, ki se izvede skozi brskalnik, uporabnik prejme opozorilo, da se njegove poverilnice ne bodo varno prenesle, številni drugi programi in storitve pa avtomatsko zavržejo povpraševanje, če je avtentikacija nastavljena na osnovno. V primeru, da žrtev vpiše poverilnice, jih prejmemo v formatu:

```
Basic Client : <ip>
Basic Username : <username>
Basic Password : <password>
```

### 3.5 Prisilen spust šifrirnega algoritma iz NTLMv2-SSP na NTLM

Windows za šifriranje poverilnic in ostalih informacij privzeto uporablja šifrirni algoritem NTLMv2--SSP, ki je nadgradnja algoritma NTLM (angl. NT LAN Manager)[16] z dodatkom SSP (angl. Security Support Provider). Na sistemu Windows SSP predstavlja dinamično knjižnico (angl. Dynamic Link Library - DLL), ki ponuja vmesnik med operacijskim sistemom in različnimi avtentikacijskimi protokoli in tako omogoča okolju Windows razširjen nabor podprtih protokolov.

SSP je v kontekstu opisanega napada zanimiv z vidika funkcionalnosti ESS (angl. Extended Session Security), katera doda "SSP" zastavico v zgoščene NTLM vrednosti, in s tem podaljša SSP zgoščeno vrednost, zaradi katere je poverilnice težje dešifrirati.

Ta korak v več procesih poznamo pod imenom soljenje (angl. salting). V tem primeru napada orodje Responder konfiguriramo z zastavico "–disable-ess", s katero prisilimo tarčo, da poverilnice pošlje v obliki NTLMv2 zgoščene vrednosti, kar pomaga pri zmanjšanju časa, ki ga rabimo za dešifriranje.

V nekaterih primerih lahko dodatno omejimo kakovost zgoščevalnega algoritma z dodatno zastavico "–lm", s katero uporabnikovo napravo silimo v uporabo protokola NTLMv1, kar še dodatno zniža nivo varnosti. Vredno je tudi omeniti, da tovrstna prisila lahko privede do opozoril ali prekinitve seje s strani tarče, vendar je med našim testiranjem do takšnih opozoril prihajalo redko, za razliko od osnovne avtentikacije.

### 3.6 Zloraba posredovanja

Posredovanje (angl. relaying) je pogosto uporabljen način za nepooblaščen dostop do sistema. Deluje po principu posrednika, ki prejme veljavno avtentikacijo in nato to zahtevo posreduje drugemu strežniku ali sistemu ter se poskuša avtenticirati tem strežniku z uporabo prejetih poverilnic. Pred takimi napadi se lahko efektivno zavarujemo s podpisovanjem, vendar različni sistemi tega zaščitnega koraka ne uporabljajo [12], ali pa ga celo ne podpirajo. Princip takšnega napada smo testirali na Samba strežniku.

Uporabimo ukaz:

*nmap -p445 –script=smb-security-mode <IP-naslov tarče>*,

ki preišče vrata 445 (privzeta vrata za SMB) na tarči in preveri varnostno stanje konfiguracije Samba strežnika. Pridobimo odgovor, iz katerega lahko razberemo, da je podpisovanje izklopljeno.

```
Host script results:
| smb-security-mode
| account_used: guest
| authentication_level: user
| challange_response: supported
| message_signing: disabled (dangerous, but
  default)
```

Za napad nato uporabimo skripto "MultiRelay.py", ki jo lahko najdemo med seznamom orodij v orodju Responder. Skripti z zastavico "-t" nastavimo tarčo (kamor bodo poverilnice posredovane), ter z zastavico "-u" izvore, iz katerih sprejemamo poverilnice.

Vredno je tudi omeniti, da ta skripta ni bila posodobljena že od leta 2016 in je za njeno delovanje potrebna manjša prilagoditev — *thread.Daemon = True*.

Po zagonu skripte, ki je poslušala za poizvedbami na omrežju, smo uspešno izvedli posredovanje poverilnic Samba strežniku in tako pridobili dostop do strežnika.

### 3.7 DNS vrivanje v DHCP odgovoru

Če se v omrežju uporablja DHCP za identifikacijo IP-naslovov strežnikov, lahko orodje Responder v DHCP načinu tarči v odgovor podtakne lažni DNS zapis.

Orodje Responder lahko vzpostavi lažni DNS strežnik[5]. Ko žrtev poskuša dostopiti do naslova, najprej razreši ime z iskanjem DNS strežnika, kar stori s pošiljanjem DHCP zahteve. Responder odgovori na to zahtevo in v DHCP odgovor vstavi svoj IP-naslov DNS strežnika, in tako zastrupi odgovor. Ko žrtev to prejme, vidi IP-naslov lažnega DNS strežnika in z njegovo pomočjo poskuša dostopati do strežnika/storitve, vendar nevede dostopa le do napadalca.

Responder lahko zaženemo v DHCP-DNS poisoning načinu z zastavico "-D".

## 4 LLMNR AUTOMATION

Za izvedbo takšnih napadov lahko tudi avtomatiziramo opisane postopke in jih posledično tudi poenostavimo, na primer v primeru obsežnejših napadov z več tarčami. V ta namen smo ustvarili skripto LLMNRAutomation.sh in konfiguracijsko datoteko LLMNRAutomation.conf, ki se nahajata v našem javnem GitHub repozitoriju: https://github.com/Urichh/LLMNR-automation.

Skripta deluje v štirih korakih, izmed katerih dva slonita na drugih orodjih, ki sta potrebni za pravilno delovanje skripte. Ti dve orodji sta Responder (https://github.com/SpiderLabs/Responder) ter Hashcat (https://github.com/hashcat/hashcat). Obe orodji sta prosto dostopni. Omenjeni štirje koraki so:

1. Branje konfiguracijske datoteke in začetek poslušanja z želenimi nastavitvami.
2. Zajem zgoščenih poverilnic.
3. Organizacija zajetih poverilnic v logično datotečno strukturo za lažje dešifriranje.
4. Dešifriranje zgoščenih poverilnic.

V tem sklopu bomo razložili kako s pomočjo skripte avtomatizirano pridemo do dešifriranih poverilnic.

### 4.1 Konfiguracija

V repozitoriju se poleg LLMNRAutomation.sh skripte nahaja tudi konfiguracijska datoteka LLMNRAutomation.conf, v kateri so nastavitve s katerimi se nato zažene skripta. Datoteka vsebuje šest glavnih razdelkov:

- Vmesnik (angl. interface) – nastavitev vmesnika, na katerem skripta posluša in oddaja. Privzeta vrednost: eth0.
- Želeni strežniki – uporabnik vklopi ali izklopi lažne strežnike, ki jih skripta nato zažene. Privzeto so vsi vklopljeni.
- Preferirana avtentikacijska metoda - uporabnik nastavi šifrirni algoritem. Privzeta vrednost: NTLMv2-SSP.
- Lažni (angl. rogue) WPAD strežnik - uporabnik vklopi ali izklopi, če skripta zažene lažni WPAD strežnik za odgovore na DHCP poplavljanja. Ta nastavitev strežnika je ločena od drugih, ker se žrtev na WPAD strežnik ne povezuje neposredno, temveč je uporabljen v kombinaciji z drugimi. Privzeta vrednost: Off.
- DHCP-DNS vrivanje - uporabnik vklopi ali izklopi DHCP-DNS vrivanje, kot je razloženo v "DNS injection v DHCP odgovoru" razdelku poglavja o orodju responder. Privzeta vrednost: Izklopljena.
- Lažni zunanji IP-naslov - uporabnik lahko izbere lažni IP-naslov, iz katerega bo tarča prejela zastrupljene odgovore. Privzeta vrednost: None.

### 4.2 Organizacija zajetih poverilnic

Ko uporabnik ustavi skripto, se zajete zgoščene poverilnice shranijo v direktorij imenovan "hashes" znotraj direktorija v katerem se nahaja skripta. Te zgoščene vrednosti so urejene po IP-naslovih tarč, znotraj katerega so urejeni po protokolu in nazadnje, znotraj tekstovnih datotek, so ločene po uporabniških imenih, kot je prikazano v spodnjem diagramu:

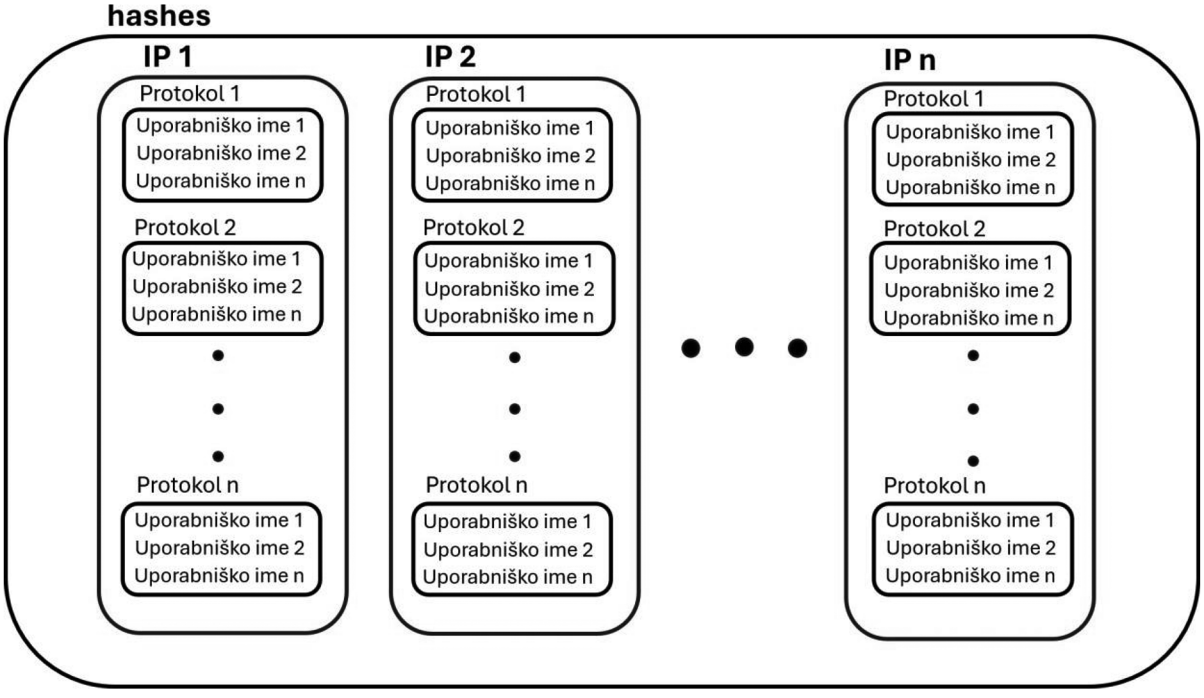
Ta datotečna struktura pomaga pri izbiri optimalnih zgoščenih vrednosti za dešifriranje, da se čim lažje prebijemo do želenih poverilnic.

### 4.3 Dešifriranje poverilnic

Ko zaključimo fazo zajema poverilnic, lahko preidemo na zadnjo fazo, ki je dešifriranje poverilnic. To storimo tako, da skripto zaženemo z zastavico "-c":

*./LLMNRAutomation.sh -c*

Če skripto zaženemo, dobimo izpis vseh zajetih poverilnic urejenih po IP-naslovu, ter nato po uporabniškem imenu. Npr:




Slika 4: **Organizacija zgoščenih poverilnic**

```
IP: <ip>
• uporabniško ime 1
• uporabniško ime 2
• ...
• uporabniško ime n
```

Nato lahko izberemo katero zgoščeno vrednost želimo dešifrirati, kar storimo z ukazom:

*./LLMNRAutomation.sh -c -i <ip> -u <uporabniško ime>*

Ta modul skripte nato dešifrira izbrane poverilnice z uporabo orodja Hashcat[7]. Za izbrano uporabniško ime, povezano z izbranim IP-naslovom, samodejno izbere najlažje zgoščene vrednosti za dešifriranje, ki si sledijo od najlažje do najtežje v slednjem zaporedju:

1. Golo besedilo (angl. plain text)
2. NTLMv1
3. NTLMv1-SSP
4. NTLMv2
5. NTLMv2-SSP

Hashcat poleg vhodne in izhodne datoteke zahteva še dodatni argument, ki predstavlja vrsto podane zgoščene vrednosti. Skripta to vrednost zazna avtomatsko, glede na spodnjo tabelo[9]:

Tabela 1: **Vrste algoritmov**

| **Algoritev** | **Vrsta** |
|---|---|
| Golo besedilo | |
| NTLMv1 | 1000 |
| NTLMv2 | 5600 |

Skripta nato začne z dešifriranjem poverilnic, vendar čas dešifriranja lahko močno variira glede na kompleksnost in dolžino gesla. Nazadnje skripta še shrani dešifrirane poverilnice v besedilno datoteko znotraj direktorija "cracked", ki je v istem direktoriju kot skripta.

## 5 DISKUSIJA IN ZAKLJUČEK

Čeprav so predstavljeni napadi lahko zelo nevarni, obstajajo številni obrambni mehanizmi, ki jih lahko preprečijo, ali pa vsaj minimizirajo posledice. Skrbniki IT okolij lahko, razen v primeru kjer narava organizacije to preprečuje, tovrstno razreševanje izklopijo, kar lahko storijo npr. kar preko upravljanja s politiko skupine (angl. group policy). V primeru, da si organizacija tega ne more privoščiti, pa lahko k varnosti pripomorejo z implementacijo omejenih dostopov do omrežja (ang. network access control) kot npr. protokol 802.1x. Poleg tega, se efektivnost takih zlorab lahko močno zmanjša s splošno dobrimi varnostnimi praksami, kot so podpisovanje zahtevkov/odgovorov na SMB strežnikih, preprečevanje uporabe zastarelih šifrirnih algoritmov, močna gesla, ki otežujejo dešifriranje gesel in ostale splošne dobre prakse infrastrukturne varnosti, kot so ločitev (segmentacija) omrežij.

V modernih računalniških sistemih je še vedno veliko vidikov, ki bodisi zaradi lahkote uporabe, povratne združljivosti ali drugih razlogov lahko predstavljajo varnostne luknje. Napadi, predstavljeni v tem članku po večini ne predstavljajo takojšnje neposredne grožnje za varnost računalniških sistemov, saj je za izvedbo takšnega napada potreben dostop do lokalnega omrežja, znotraj katerega tovrstni protokoli za razreševanje niso blokirani. Prav tako pa imajo napadalci ob uspešno izvedenem napadu pred sabo še mnogo ovir, kot so dejansko dešifriranje poverilnic, ki je ob ustrezno zahtevni kompleksnosti gesel in močnih šifrirnih algoritmih lahko zelo časovno potratno, poleg tega pa lahko zelo pomagajo tudi ostali preventivni ukrepi, kot so večstopenjska avtentikacija in podobni prijemi. Kljub vsemu pa je pomembno tem zlorabam posvetiti pozornost, saj za zagotavljanje varnosti vseeno želimo minimizirati potencialno ranljive vidike in tako zmanjšati število potencialnih vektorjev napada.

V tem članku smo predstavili delovanje protokolov za razreševanje imen v lokalnih omrežjih z uporabo večvrstnih poizvedb, ter kako lahko te protokole zlorabljajo napadalci z namenom zasega šifriranih poverilnic. Demonstrirali smo tudi delovanje orodij za tovrstne napade ter različne uporabe le-teh, kar smo nato nadgradili v lastno orodje za avtomatizacijo napadov in za konec predstavili še efektivne obrambne mehanizme. Hitra rast procesorske moči za namene dešifriranja poverilnic, neodpornost trenutnih šifrirnih algoritmov na kvantne računalnike, vedno večje kompleksnosti omrežij, večja uporaba mrežnih storitev namesto »tradicionalnih« namiznih programov in ostali dejavniki so razlogi, zaradi katerih menimo, da je zaščita pred takšnimi napadi v današnjem svetu ključnega pomena.

## LITERATURA


[1] B Aboba, D Thaler in L Esibov. *RFC 4795*. English. Jan. 2007. URL: https://www.rfc-editor.org/ rfc/rfc4795.html (pridobljeno 5. 9. 2024).

[2] Iliano Cervesato. "Empirical Study of the Impact of Metasploit-Related Attacks in 4 Years of Attack Traces". English. V: *Advances in Computer Science - ASIAN* 2007. Doha, Qatar: Springer, dec. 2007, str. 198–211. ISBN: 3-540-76927-7. URL: https://link.springer.com/chapter/10.1007/978-3- 540-76929-3_19 (pridobljeno 21. 5. 2024).

[3] S Cheshire in M Krochmal. *Multicast DNS*. English. 2013. URL: https://datatracker.ietf.org/ doc/html/rfc6762 (pridobljeno 5. 9. 2024).

[4] R Cichocki. "State-Sponsored and Organized Crime Threats to Maritime Transportation Systems in the Context of the Attack on Ukraine". English. V: *TransNav. the International Journal on Marine Navigation and Safety of Sea Transportation* 17.3 (sep. 2023), str. 5. URL: https://bibliotekanauki. pl/articles/24811512.pdf.

[5] Maven Cybertech. *Using Responder to Capture Credentials. English.* Okt. 2023. URL: https :// systemweakness . com / using - responder - to - capture - the - credentials - a9d5a1013333 (prido- bljeno 5. 9. 2023).

[6] Davis Freimanis. "Vulnerability Assessment of Authentication Methods in a Large-Scale Computer System". English. Magistrsko delo. SCHOOL OF ELECTRICAL ENGINEERING in COMPUTER SCIENCE: KTH ROYAL INSTITUTE OF TECHNOLOGY, maj 2019. URL: https : / / www . diva - portal.org/ smash/get/diva2:1358687/FULLTEXT01.pdf (pridobljeno 5. 9. 2024).

[7] Radek Hranicky` in sod. "Distributed password cracking with BOINC and hashcat". V: *Digital Investiga- tion 30* (2019). Publisher: Elsevier, str. 161–172.

[8] William Hurer-Mackay. *LLMNR and NBT-NS Poisoning Using Responder.* English. Jun. 2016. URL: https : / / www . 4armed . com / blog / llmnr - nbtns - poisoning - using - responder/ (pridobljeno 5. 9. 2024).

[9] Nicklas Mortensen Hamang. "Effective Password Cracking". English. Magistrsko delo. Faculty of ma- thematics in natural sciences: University of Oslo, 2019. URL: https://www.duo. uio.no/bitstream/ handle/10852/73247/Nicklas_M_Hamang_ Master_Thesis.pdf (pridobljeno 5. 9. 2024).

[10] *NetBIOS over TCP/IPNetbio's NBT-NS Poisoning*. English. (Pridobljeno 19. 5. 2024).

[11] Mike O'Leary. "Attacking the Windows Domain". English. V: *Cyber Operations: Building, Defending, and Attacking Modern Computer Networks*. 2nd. Apress Media LLC, feb. 2019, str. 1151. ISBN: 978- 1-4842-4294-0. URL: https:// link.springer.com/chapter/10.1007/978- 1- 4842- 4294- 0_8 (pridobljeno 5. 9. 2024).[12] Alexander Oberle in sod. "Preventing pass-the-hash and similar impersonation attacks in enterprise infrastructures". V: *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2016, str. 800–807.

[13] Konstantinos Pantazis. "An External Red Team Assessment in a Corporate Environment". English. Doktorska disertacija. Department of Information in Electronic Engineering: International Hellenic University of Greece, 2022. URL: https : / / www . researchgate . net / profile / Konstantinos - Pantazis - 8/ publication / 364958274 _ An _ External _ Red _ Team _ Assessment _ in _a_ Corporate _ Environment / links / 63610c3a8d4484154a53def7 / An - External - Red - Team - Assessment - in - a- Corporate-Environment.pdf (pridobljeno 5. 9. 2024).

[14] Abdurrahman Pektaş. "Practical Approach For Securing Windows Environment: Attack Vectors And Countermeasures". V: *International Journal of Network Security & Its Applications (IJNSA) Vol 9* (2017). URL: https://papers.ssrn.com/ sol3/papers.cfm?abstract_id=3649907.

[15] U Steinhoff, A Wiesmaier in R Araújo. *The State of the Art in DNS Spoofing*. English. 2006. URL: https : / / citeseerx . ist . psu . edu / document ? repid = rep1 & type = pdf & doi = 7fdd734e684c6eb79a61864bb418ddc93a6ac751 (pridobljeno 21. 5. 2024).

[16] Nuno Tavares. *NTLM vs KERBEROS*. English. Apr. 2018. URL: https://answers.microsoft.com/ en - us / msoffice / forum / all / ntlm - vs - kerberos / d8b139bf - 6b5a - 4a53 - 9a00 - bb75d4e219eb (pridobljeno 5. 9. 2023).

[17] Zhihao Zheng in sod. "Best Practices in Designing and Implementing Cloud Authentication Schemes". English. V: *CS & IT Conference Proceedings*. Zv. 11. Issue: 3. CS & IT Conference Proceedings, 2021, str. 10. URL: https://www.csitcp.com/ paper/11/113csit07.pdf.


▪

**Urban Dopudja** je študent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Čas posveča strokovnim izpopolnjevanjem na področju kibernetske varnosti. Njegovi raziskovalni interesi segajo na področja spletne varnosti, omrežnih protokolov in nizkonivojske analize sistemov.

▪

**Matevž Pesek** je docent in raziskovalec na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je diplomiral (2012) in doktoriral (2018). Od leta 2009 je član Laboratorija za računalniško grafiko in multimedije. Od leta 2024 izvaja predmet Varnost programov.