VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2021 Bc. Kamil Hons

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A IMPLEMENTACE POSTUPŮ PRO AUTOMATIZOVANÉ ŘEŠENÍ BEZPEČNOSTNÍCH INCIDENTŮ

PROPOSAL AND IMPLEMENTATION OF PROCEDURES FOR AUTOMATED RESPONSE OF SECURITY INCIDENTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. Kamil Hons**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Zdeněk Martinásek, Ph.D.**

SUPERVISOR

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Kamil Hons ***ID:*** 195834
Ročník: 2 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Návrh a implementace postupů pro automatizované řešení bezpečnostních incidentů

POKYNY PRO VYPRACOVÁNÍ:

Hlavním cílem práce je návrh a implementace nejméně třech automatických postupů řešení bezpečnostních incidentů ve firemním informačním systému. V teoretické části práce student analyzuje současný stav problematiky a navrhne obecné postupy pro řešení bezpečnostních incidentů v prostředí informačního systému. Tyto metody budou využívat automaticky získávané záznamy událostí. Student podrobně popíše celý postup řešení od vzniku bezpečnostního incidentu až po jednotlivé detailní kroky při jeho řešení, přičemž v určitých fázích řešení může být výhodné počítat s potvrzením nebo s rozhodnutím o dalším postupu operátora (poloautomatický proces). Navržené postupy budou zaznamenané formou vývojových diagramů. V praktické části student implementuje dle vývojových diagramů alespoň tři postupy automatizovaného řešení bezpečnostních incidentů v programovacím jazyku Python. Tyto postupy v následujícím kroku integruje do funkční technologie. Navržené a implementované postupy budou ověřeny na experimentálním pracovišti, které bude obsahovat vzorovou infrastrukturu (Active Directory, firewall, IDS, AV řešení, apod.). Dosažené výsledky budou přehledně analyzovány.

DOPORUČENÁ LITERATURA:

[1] CHUVAKIN, Anton; SCHMIDT, Kevin; PHILLIPS, Chris. Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management. Newnes, 2012.

[2] FURNELL, Steven M., et al. Preparation, detection, and analysis: the diagnostic work of IT security incident response. Information Management & Computer Security, 2010.

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 24.5.2021

Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.
Konzultant: Kamil Doležel (Service & Support spol. s r. o.)

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Tato diplomová práce se zabývá vytvořením návrhů, postupů pro řešení bezpečnostních incidentů, a to jak z teoretického, tak i z praktického hlediska. V rámci práce byly vytvořeny tři obecné scénáře v podobě grafických schémat, vytvořených v programu Inkscape, sloužící jako teoretická předloha k automatickému řešení bezpečnostních incidentů. První navržený scénář navrhuje obecný postup řešení události, ve které je příloha emailu označena jako podezřelá. Druhý scénář slouží jako návrh postupu řešení události, kdy je podezření na komunikaci nedůvěryhodné externí IP adresy s adresou lokální. Třetí scénář navrhuje postup šetření pro události, kde je třeba prošetřit podezřelý soubor na vzdáleném zařízení. Na základě těchto vytvořených scénářů byla provedena a zdokumentována praktická implementace postupů pro automatické řešení bezpečnostních incidentů v programovacím jazyce Python uvnitř prostředí Splunk Phantom. V rámci dokumentace implementace scénářů byla vytvořena dvě audiovizuální demonstrace pro ilustrování vytvořeného prostředí a funkce implementovaných scénářů za pomoci nástroje OBS a Blender. Jednotlivé implementace jsou na závěr práce testovány automatickým spuštěním nad událostmi z definovaného časového rozsahu. Výsledky jsou přehledně analyzovány formou tabulek z důvodu určení úspěšnosti těchto scénářů, jež se zakládá na prověření odlišnosti výsledků analýz od předpokladu. Na základě analýzy byly praktické implementace scénářů upraveny tak, aby jejich výstup odpovídal předpokladu. Výsledkem se tak stávají tři navržené, otestované a analyzované scénáře, které mohou dále sloužit jako základ pro specifické implementace ve firemním informačním systému. Samotná implementace teoretických scénářů byla provedena v rámci testovacího prostředí a součástí práce je popis komunikace a nastavení daného prostředí. V závěru byly popsány výsledky jednotlivých scénářů.

KLÍČOVÁ SLOVA

analýza, automatické, odpověď na událost, postup, událost

ABSTRACT

This diploma thesis deals with the development of proposals for procedures for dealing with security incidents, both from a theoretical and practical point of view. Three generic scenarios in the form of graphical diagrams, designed in Inkscape program, were created as a theoretical template for the automatic handling of security incidents. The first proposed scenario suggests a general procedure for dealing with an event in which an email attachment is marked as suspicious. The second scenario serves as a suggested procedure for handling an event, where an untrusted external IP address is suspected to be communicating with a local one. The third scenario then suggests an investigation procedure for events, where a suspicious file on a remote device needs to be investigated. Based on these created scenarios, a practical implementation of procedures for automized solving of security incidents was performed and documented in the Python programming language within the Splunk Phantom environment. As part of the documentation of the scenario implementation, two audiovisual demonstrations were created to illustrate the designed environment and the functionality of the implemented scenarios using programs such as OBS and Blender. The individual implementations are tested at the end of the thesis by running them automatically over events from a defined time range. The results are clearly analyzed in the form of tables to determine the success of these scenarios, which is based on checking how the analysis results differ from the original assumptions. Based on the analysis, the practical implementations of the scenarios have been modified to ensure that their output matches with the assumption. Thus, results are three proposed, tested and analyzed scenarios, which can further serve as a basis for specific implementations in a corporate information system. The actual implementation of the theoretical scenarios was carried out within a testing environment and the work includes a description of the communication and a setup of the environment. Finally, the results of the individual scenarios were described.

KEYWORDS

analysis, automatic, incident response, procedure, event

HONS, Kamil. *Návrh a implementace postupů pro automatizované řešení bezpečnostních incidentů*. Brno, 2021, 98 s. Diplomová práce. Vysoké učení technické v Brně, Faculty of Electrical Engineering and Communication, Department of Telecommunications. Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Návrh a implementace postupů pro automatizované řešení bezpečnostních incidentů“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu mé práce, Ing. Zdeňku Martináskovi, Ph.D. za jeho přínosné komentáře a odborné vedení práce. Dále bych rád poděkoval odbornému konzultantovi Kamilu Doleželovi za jeho praktické rady a pomoc s překonáváním technických překážek. V neposlední řadě bych také rád poděkoval své rodině a blízkým za trpělivost a podporu.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

Tato diplomová práce vznikla ve spolupráci s firmou Service and Support. Hlavním cílem bylo vytvoření funkčních scénářů pro automatické řešení bezpečnostních incidentů. Bezpečnostní incident je událost, při které došlo k narušení nebo ohrožení bezpečnosti organizace. Následkem bezpečnostního incidentu může být například ohrožení aktiv, jména nebo obchodního tajemství organizace. Pro včasnou detekci těchto událostí a incidentů slouží systémy typu *Security Information and Event Management* (SIEM), které agregují záznamy událostí napříč organizací. Jedním ze systémů tohoto typu je řešení Splunk Enterprise, které firma Service and Support implementuje. Po detekci bezpečnostní události je třeba provést šetření, které má za úkol zjistit zda se jedná o bezpečnostní incident. Pro automatizaci šetření jsou využívány systémy typu *Security Orchestration, Automation, and Response* (SOAR), které umožňují prošetřovat události v krátkém čase po jejich vzniku a jsou schopny provádět preventivní akce zabraňující škodlivým aktivitám útočníka. Na základě zájmu zákazníků o automatické řešení incidentů společnosti Service and Support bylo vybráno řešení typu SOAR jménem Splunk Phantom. Pro účely budoucího rozvoje bezpečnosti na straně zákazníků produktem Splunk Phantom bylo nutné provést vývoj v oblasti návrhů scénářů, které dále mohou sloužit pro automatické řešení bezpečnostních incidentů. Tyto scénáře mají získat podobu, která může být nasazena ve formě, ve které se nachází nebo mají sloužit jako předloha pro více přizpůsobené řešení.

Síťová infrastruktura je v dnešní době obrovským hnacím kolem nemalého počtu firem. Zavádění informačních procesů, jejich zdokonalování a rozšiřování se stalo klíčovým nejen z hlediska zlepšení komunikace mezi jednotlivými firemními útvary, ale také v otázce efektivity. Současným trendem se tak stalo rozrůstání informačních procesů pro rozvoj podnikání. S tímto nárůstem se však také razantně zvyšují objemy dat, které jsou generovány. Tato skutečnost přímo ovlivňuje také pole informační bezpečnosti, která musí na stále se zvyšující objemy dat adekvátně reagovat. Je tedy nutné spolu s informačními systémy také implementovat bezpečnostní systémy, jejichž snahou je zabraňovat, detekovat a předcházet bezpečnostním incidentům. Týmy věnující se řešení těchto incidentů není výhodné ani efektivní rozšiřovat tak, aby byly schopné řešit větší objemy bezpečnostních incidentů manuálně.

S nárůstem informačních systémů je otázka bezpečnosti důležitější než kdy dříve, a je nutné zavádět opatření, která efektivně a spolehlivě řeší bezpečnostní incidenty. Pro tento účel je ideálním nástrojem automatizace, která umožní vytíženým bezpečnostním týmům řešit incidenty rychleji. Klíčovou vlastností automatizace řešení bezpečnostních incidentů je především úspora času na jednodušších, ale také na komplexních krocích, které bezpečnostní experti pro svou práci potřebují. Řešení

komplexních bezpečnostních incidentů spočívá ve správném sestavení automatických či poloautomatických postupů, které jsou schopny nejen detekce, ale také prevence. Příklady systémů typu SOAR jsou například FortiSOAR, Cortex XSOAR nebo zvolené řešení Splunk Phantom.

Hlavním cílem diplomové práce je vytvoření automatického, komplexního postupu ve formě vývojového diagramu pro řešení bezpečnostních incidentů ve firemním informačním systému, který bude využívat automaticky získávané záznamy událostí. Tento postup bude následně implementován do prostředí technologie Splunk Phantom. V tomto prostředí bude testován na experimentálním pracovišti se vzorovou infrastrukturou.

1 Informační bezpečnost organizací

Pod tímto pojmem je skryto mnoho procesů a faktorů, které se přímo podílí na samotné informační bezpečnosti organizace. Jejím účelem je stanovit postupy, opatření a nařízení pro zabezpečení veškerých důležitých informací organizace [1]. Organizace *National Institute of Standards and Technology* (NIST) upozorňuje, že mít schopnost odpovědět na kybernetický incident je důležitá a má několik výhod. Hlavními výhodami pro organizace jsou minimalizace přerušení pracovního výkonu organizace, a tím značné ušetření nákladů. Dále možnost detekovat incidenty, a tím zvyšovat obranu organizace do budoucna na základě zjištěných informací, nebo například minimalizace škod v podobě ztráty dat [2].

Z normativního hlediska je zde několik možností, kterého se lze při návrhu a implementaci bezpečnosti organizace držet. Z hlediska českého zákona zde máme **zákon č. 181/2014 Sb. o kybernetické bezpečnosti**, který legislativně upravuje práva a povinnosti stanovených subjektů [3]. Dále je zde ISO 27000, což je soubor norem, věnující se stanovení rámce pro řešení bezpečnostních incidentů a bezpečnosti organizacím jako takovým. Potom také ITIL v4, jež se snaží vymezit procesní postupy pro správu informačních technologií [4]. A v neposlední řadě NIST a jeho normy.

Motivací pro řešení informační bezpečnosti organizace, je více:

> Digitální forenzní techniky mohou být použity pro mnoho účelů včetně:
>
> - investigace zločinu a porušení vnitřních politik,
> - rekonstrukce počítačových bezpečnostních incidentů,
> - řešení operačních problémů,
> - obnovení z nechtěných škod systému [5].

Zákon o kybernetické bezpečnosti

Jedná se o legislativní znění, upravující práva a povinnosti **provozovatele základní služby**, poskytující **základní službu**, dále orgánům a osobám spravujícím, poskytujícím nebo provozujícím **významným informačním systémem**, **významnou síť**, **informační systém základní služby**, **informační, nebo komunikační systémem** pro kritickou infrastrukturu. Základní služba je poté definována v § 2 odst. 1 písm. i) zákona č. 181/2014 Sb. o kybernetické bezpečnosti:

> ... služba, jejíž poskytování je závislé na sítích elektronických komunikací nebo informačních systémech a jejíž narušení by mohlo mít významný dopad na zabezpečení společenských nebo ekonomických činností v některém z těchto odvětví: energetika, doprava, bankovnictví, infrastruktura finančních trhů, zdravotnictví, vodní hospodářství, digitální infrastruktura, chemický průmysl.

Dále zákon v § 2 odst. 1 písm. k) vymezuje kdo je provozovatelem základní služby takto:

> ... orgán nebo osoba, která poskytuje základní službu a která je určena Národním úřadem pro kybernetickou a informační bezpečnost ...

Definice kritické informační infrastruktury je definována v § 2 odst. 1 písm. b):

> ... prvek nebo systém prvků kritické infrastruktury v odvětví komunikační a informační systémy v oblasti kybernetické bezpečnosti.

Definice významného informačního systému dle § 2 odst. 1 písm. d):

> informační systém spravovaný orgánem veřejné moci, který není kritickou informační infrastrukturou ani informačním systémem základní služby a u kterého narušení bezpečnosti informací může omezit nebo výrazně ohrozit výkon působnosti orgánu veřejné moci,

Další zajímavou oblastí jsou potom provozovatelé **digitální služby**, definice digitální služby dle § 2 odst. 1 písm. l) bod 1., 2.:

> on-line tržiště, které spotřebiteli nebo prodávajícímu umožňuje on-line uzavírat s prodávajícím podnikatelem kupní smlouvu nebo smlouvu o poskytnutí služeb, a to prostřednictvím internetové stránky on-line tržiště nebo prostřednictvím internetové stránky prodávajícího, který využívá službu poskytovanou on-line tržištěm, internetového vyhledávače, který umožňuje provádět vyhledávání v zásadě na všech internetových stránkách, a to na základě dotazu uživatele na jakékoliv téma v podobě klíčového slova, sousloví nebo jiného zadání, přičemž služba poskytuje odkazy, na nichž lze nalézt informace související s požadovaným obsahem, nebo cloud computingu, který umožňuje přístup k rozšířitelnému a přizpůsobitelnému úložišti nebo výpočetním zdrojům, které je možné sdílet.

Poté § 3 odst. 1 dále vymezuje orgány a osoby, kterým se **ukládají** povinnosti v oblasti kybernetické bezpečnosti následovně:

- a) poskytovatel služby elektronických komunikací a subjekt zajišťující síť elektronických komunikací1), pokud není orgánem nebo osobou podle písmene b),
- b) orgán nebo osoba zajišťující významnou síť, pokud nejsou správcem nebo provozovatelem komunikačního systému podle písmene d),
- c) správce a provozovatel informačního systému kritické informační infrastruktury,
- d) správce a provozovatel komunikačního systému kritické informační infrastruktury,
- e) správce a provozovatel významného informačního systému,
- f) správce a provozovatel informačního systému základní služby, pokud nejsou správcem nebo provozovatelem podle písmene c) nebo d),
- g) provozovatel základní služby, pokud není správcem nebo provozovatelem podle písmene f), a h) poskytovatel digitální služby [3].

Pro orgány definované v písmenu c) až f) je poté uložena povinnost zavádět bezpečnostní opatření v podobě souhrnu úkonů. O zavedených opatřeních je pak nutné vést bezpečnostní dokumentaci. Pří výběru dodavatele jsou tyto orgány povinné požadavky na bezpečnost zakotvit smluvně. Tyto požadavky nelze považovat za nezákonné omezení hospodářské soutěže. Poskytovatelům **digitální služby** je mimo jiné, uloženo **zvládání kybernetických bezpečnostních incidentů** v § 4 odst. 3:

> Poskytovatel digitální služby je povinen zavést a provádět vhodná a přiměřená bezpečnostní opatření pro sítě elektronických komunikací a informační systémy, které využívá v souvislosti se zajišťováním své služby, přičemž tato bezpečnostní opatření zohledňují zajištění bezpečnosti informací, zvládání kybernetických bezpečnostních incidentů, řízení kontinuity činností, monitorování, audit, testování a soulad s mezinárodními předpisy [3].

Zákon o kybernetické bezpečnosti dále definuje pojmy **nástroj pro sběr a vyhodnocení kybernetických událostí**, a je jím považován za **technické opatření**, z hlediska **organizačního opatření** pak **zvládání kybernetických bezpečnostních událostí a kybernetických bezpečnostních incidentů**. V § 11 jsou definována opatření a kdo je povinen je zavádět:

> (1) Opatřeními se rozumí úkony, jichž je třeba k ochraně informačních systémů nebo služeb a sítí elektronických komunikací před hrozbou v oblasti kybernetické bezpečnosti nebo před kybernetickým bezpečnostním incidentem anebo k řešení již nastalého kybernetického bezpečnostního incidentu.
>
> (2) Opatřeními jsou: a) varování, b) reaktivní opatření a c) ochranné opatření.
>
> (3) Reaktivní opatření jsou povinny provádět
>
> a) orgány a osoby uvedené v § 3 písm. a) a b) za stavu kybernetického nebezpečí nebo za nouzového stavu vyhlášeného na základě žádosti podle § 21 odst. 6 a
>
> b) orgány a osoby uvedené v § 3 písm. c) až f).
>
> (4) Ochranné opatření jsou povinny provádět orgány a osoby uvedené v § 3 písm. c) až f) [3].

ISO 27000

Jedná se o normy popisující možnosti pro řízení bezpečnosti informací publikované společností *Interantional Organazation for Standardization* (ISO). Jsou to placené normy a slouží jako návrhy postupů. Jednou z hlavních norem je ISO 27003, směrnice pro implementaci systému řízení bezpečnosti. Dále pak ISO 27035-2, jež je současnou verzí zaměřující se na plánování a přípravu odezvy na incidenty. Předmětem normy je mimo jiné zaměření na fázi „plánování a přípravy“, zejména pak na politiku řízení incidentů bezpečnosti informací, závazek vrcholového vedení, plán řízení incidentů bezpečnosti informací a ustavení týmu pro odezvu na incidenty *Incident Response Team* (IRT), atp. [6] [7] [4].

1.1 Bezpečnostní týmy

Pro zajištění bezpečnosti jsou sestavovány týmy na úrovni organizací, jako například firma nebo stát. Tyto týmy se věnují Forenzní činnosti při hledání důkazů o vniknutí do systému. Ta spočívá ve zpětném hledání stop a důkazů o vniknutí, které následně mohou pomoci v nápravě incidentu. Digitální forenzní proces je pak definován následovně:

1. sběr;
2. zkoumání;
3. analýza;
4. hlášení [5].

Sestavené IRT týmy jsou pak zodpovědné za vykonávání těchto procesů. Běžnou praxí je pak využívání služeb třetích stran k dohledávání jednotlivých informací v rámci bezpečnostní události, na základě kterých potom tým rozhoduje o false positive, false negative, true negative, nebo true positive situaci události. Tyto případy jsou definovány pomocí matice záměn viz tabulka 1.1 [8].

Tab. 1.1: Matice záměn binární klasifikace

	Skutečný stav	Klasifikátor	
Tvrzení klasifikátoru		Pozitivní	Negativní
Skutečnost	Pozitivní	**true positive**	**false negative**
	Negativní	**false positive**	**true negative**

V případě vyhodnocení true positive je pak možné označit událost jako incident. Šetření se potom vztahuje k informacím ohledně souborů, stavů operačních systémů, síťového provozu, nebo aplikací. Jednotlivé kroky procesu forenzní analýzy jsou tedy sběr dat, jehož účelem je najít, označit, sesbírat a uchovat data důležitá pro daný incident. Dále proces Zkoumání má za účel zjistit obsah a účel dat, které byly zajištěny, a to způsobem neporušujícím jejich integritu. V praxi se může jednat o kombinaci automatických a manuálních nástrojů, pomocí nichž lze například spustit kód nebo otevřít dokument za účelem zjištění možných nestandardních aktivit. Proces analýzy se potom soustředí na vyhodnocování výsledků získaných zkoumáním tak, aby zodpověděl otázky a domněnky vedoucí k prvotnímu označení skutečnosti jako událost. Na závěr, proces hlášení má za úkol popsat co se stalo, jaké akce byly vykonány. Výstupem šetření je navržení opatření, vyžádání interakce uživatele nebo zlepšení bezpečnosti organizace do budoucna [5].

1.2 Bezpečnostní incident

Zákon o kybernetické bezpečnosti v § 7 odst. 1 definuje **Kybernetickou bezpečnostní událost** následovně:

> ... událost, která může způsobit narušení bezpečnosti informací v informačních systémech nebo narušení bezpečnosti služeb anebo bezpečnosti a integrity sítí elektronických komunikací

Definice **kybernetického bezpečnostního incidentu** v § 7 odst. 2:

> ... je narušení bezpečnosti informací v informačních systémech nebo narušení bezpečnosti služeb anebo bezpečnosti a integrity sítí elektronických komunikací v důsledku kybernetické bezpečnostní události [3].

Z těchto ustanovení vyplývá, že **bezpečnostní událost** je prvním stupněm, který se následně může ukázat jako bezpečnostní incident. Na základě tohoto ustanovení je tedy patrné, že pro zvýšení bezpečnosti je zapotřebí implementovat **technické opatření** v podobě **nástroje pro sběr a vyhodnocení kybernetických událostí**. Pro účely zjištění co nejmenšího počtu false positive událostí je důležité implementovat nástroj, který má co možná nejvyšší množství detekčních technik.

NIST definuje incident jako porušení nebo hrozbu nastaveným politikám bezpečnosti organizace. Dále je uveden příklad incidentu v podobě útoku *Denial of Service* (DOS) jež má za úkol způsobit nedostupnost služby. V praxi má takovýto útok podobu zasílání upravených paketů krátké délky, nebo jinak upravených paketů na cílový server za účelem zahlcení části, nebo celku zařízení, a tím způsobení zamezení odbavení jiných, legitimních požadavků na server. Dalším možným způsobem je použití *Internet Control Message Protocol* (ICMP) požadavků v enormní míře, která způsobuje opět zahlcení zařízení na úrovni síťové vrstvy referenčního modelu ISO/OSI. Dále NIST uvádí příklad nebezpečného kódu. Tím mohou být například počítačoví červi, s účelem infikovat co možná nejvíce zařízení, dále také trojské koně, obsahující skrytý payload, nebo například phishing útok, kdy je zaslán falešně zkonstruovaný email v podobě blížící se legitimní zprávě jiné organizace, vyzývající například ke kliknutí na odkaz. V neposlední řadě je uveden neautorizovaný přístup, při kterém útočník může spustit kód umožňující mu například číst hesla z infikovaného stroje, nebo může dále, skrz zadní vrátka, do systému spouštět další škodlivý kód, umožňující mu například monitorování citlivé komunikace, atp. Dalšími podobami incidentu může být odcizení citlivých dat, nebo například zamezení přístupu k datům pomocí počítačového viru kategorie ransomware. Souhrnně se jedná o události, které svým následkem přímo či nepřímo ovlivňují chod organizace a dochází k významné kompromitaci systému. Důsledkem se událost mění na incident [2] [9].

1.3 Záznam událostí

Jedná se o textový záznam události, která nastala v nějakém konkrétním systému. Účel záznamu událostí (dále jen log), je například generovat zprávu pro předání statusu. Logy lze generovat na základě předem stanového času nebo na základě nastalé události, například pádu aplikace, systému, atp. Velmi rozšířeným protokolem v oblasti log managementu se stal **BSD syslog protokol**. Ten je popsán pomocí

Request for Comments (RFC) 3164. Jde o doporučení protokolu a zabývá se například řešením správy logů v rámci organizace, mimo jiné však popisuje obecné náležitosti logu jako takového, například důležitost struktury logu, tak aby obsahoval hlavní část, vypovídající o zdroji události, a také část popisující závažnost logované události. Samotná podstata logování pak má pro monitorování infrastruktury vysokou důležitost, obzvláště pak pro účely detekce bezpečnostních incidentů. Nárůst systému v informačních infrastrukturách firem, znamená také nárůst logů, které je třeba správně přeposílat na patřičné místa, tak aby je viděli patřiční lidé. Dále je nutností uchovávat integritu logů pro další zkoumání. Jedním z nástrojů, který umožňuje tyto funkce, je také nástroj Splunk [10] [11].

Například firewall mnohdy umožňuje nastavení úrovně logování, zprávy, které jsou tvořeny jsou rozděleny do kategorií:

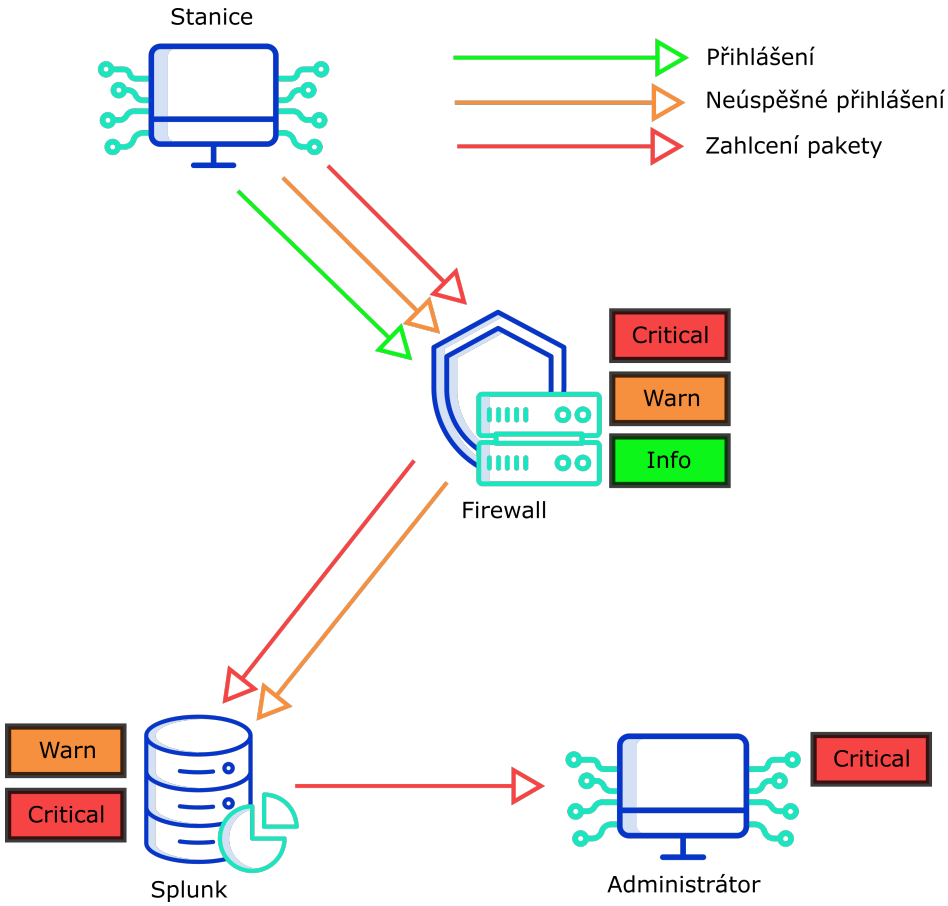
- debug-High - Detailní zprávy pro sledování co možná největšího množství událostí.
- Debug-Medium - Poměrně detailní zprávy, obsahem jsou volání pro vstupy, návratové akce, nebo výjimky v systému.
- Debug-Low - Informace jejichž podstatou není informace o specifickém podsystému, může obsahovat malé pády částí systému, nebo problémy s výkonem.
- Debug - Poměrně detailní logování používané většinou systémovými vývojáři, liší se od podsystému.
- Info - Zprávy, které mohou dávat smysl koncovým uživatelům nebo systémovým administrátorům, obsahem je postup aplikace.
- Warn - Potenciální škodlivé události, mohou zajímat koncové uživatele, nebo systémové administrátory. Obsahem jsou potenciální problémy systému.
- Error - Stavy důležité pro normální spouštění programu, avšak s tím, že aplikace stále běží.
- Fatal - Důležité chyby systému, které mohou způsobit pád systému [12].

Rozdělení do kategorií se liší systém od systému, nebo na základě výrobce. Kategorie zpráv, které pak mohou zajímat techniky v oblasti odpovědi na incidenty jsou zejména, **Warn**, **Fatal**, ale také **Info**. Dalším důležitým rozdělením je **závažnost**, ta je v rámci logu důležitým ukazatelem v ohledu upřednostňování a filtrování zpráv. Na základě tohoto pole jsou často tříděny zprávy za účely nalezení škod v systému, nebo místa průniku do systému. Přehled častých úrovní závažnosti:

- debug - většinou detailní zprávy o stavu systému,
- info - méně detailní zprávy o stavu systému,
- warning - většinou zprávy ohledně brzkého navršení limitu, atp.,
- error - zprávy o chybě systému,
- low - zprávy nízké důležitosti,
- medium - zprávy střední důležitosti,

- alert - zprávy s vyšší důležitostí,
- high - zprávy vysoké důležitosti,
- crit - zprávy nejvyšší důležitosti [13].

Proces vytvoření možných logů viz obrázek 1.1. Ve scénáři popsaném obrázkem jsou stanicí prováděny operace přihlášení označené zeleně, dále neúspěšné přihlášení oranžově a zahlcení pakety červeně. První variantou je tedy požadavek na přihlášení, ten firewall zpracuje a může o něm, pokud je úroveň logování nastavena na dostatečnou míru detekce, vytvořit log se závažností **info**. Ten je poté dočasně uložen na zařízení a nebude dále zkoumán. Dalším požadavkem je neúspěšné přihlášení na firewall. Nastavení s citlivostí na jedno přihlášení by bylo pravděpodobně moc přísné, proto předpokládejme například 3 neúspěšné pokusy v rámci pěti minut, to znamená spuštění pravidla s účelem varování a závažností **warn**. Takovýto log je pak přeposlán na Splunk server, který log uchová, a může ho dále řešit automaticky. Na závěr je zde situace zahlcení pakety firewall. taková událost by měla evokovat log se závažností **Critical**, ten je poté přesunut na Splunk server, kde proběhne vyhodnocení, a přepošle tento log administrátorovi. Příklad toho, jak může vypa-



Obr. 1.1: Příklad vzniku logů

dat log vypovídající o neúspěšném přihlášení administrátora k firewallu viz obrázek 1.2. Na obrázku lze vidět zvýrazněná pole zeleným podbarvením, prvním z nich je

```
Nov  2 14:05:27 165.85.85.42 date=2020-11-02 time=14:05:26 devname="OPR60RTK98057478"
devid="OPR60RTK98057478" logid="068542002" type="event" subtype="system" level="alert"
vd="root" eventtime=1604322326 logdesc="Admin login failed" sn="0" user="TestUser"
ui="https(195.23.45.159)" method="https" srcip=195.23.45.159 dstip=165.85.85.42
action="login" status="failed" reason="name_invalid" msg="Administrator TestUser login failed
from https(195.23.45.159) because of invalid user name"
```

Obr. 1.2: Příklad logu na firewallu

devname, jež identifikuje zařízení, ke kterému bylo přihlášení provedeno. Dalším polem je **level**, vypovídajícím o kategorii logování, následuje krátká zpráva o situaci s obsahem „Admin login failed“, poté je definován **uživatel**, který akci podnikl, jeho **srcip** tedy zdrojová IP adresa a **dstip** IP adresa cílového zařízení. Pole **status** slouží k filtraci určité množiny událostí a pole **msg** k poskytnutí člověkem čitelné informace o události. Tento log je v tomto případě kategorizován se závažností **critical**.

1.4 Data pro bezpečnostní týmy

Proces odpovědi na incident vyžaduje zdroje dat. Tato data jsou získávány zpravidla v podobě logů událostí. Pro zisk těchto logů je zapotřebí nastavení systému tak, aby prováděl monitorování sebe sama, nebo monitoroval ostatní prvky v síti. Nástroje, které umožňují základní vyhodnocení na základě signatur nebo analýzy chování pak poskytují filtrované informace, které jsou pro účely odpovědi na události vhodné. Systémem, který poskytuje data pro bezpečnostní týmy může být například *Intrusion Detection System* (IDS), *Intrusion Prevention System* (IPS), firewally, servery, ale i koncové stanice Windows či Linux. Data je možné dále získávat v podstatě z jakéhokoli zdroje v síti se schopností logování. Otázkou je pak náročnost zpracování, nebo jak významný přínos s sebou data nesou.

IDS

Jedná se o monitorovací systém, který hledá znaky vniknutí do systému či sítě. Takové znaky mohou vzniknout jako následek pokusu o kompromitaci důvěrnosti, integrity, dostupnosti, nebo bezpečnostních mechanismů. Aktivní IDS systémy také umí provádět akce za účelem zamezení, zpomalení či reportování útoku. V aktivních IDS rozeznáváme 3 typy akcí a to akci proti narušiteli, změně prostředí, nebo sběru dodatečných informací. Systémy IDS mohou být softwarové nebo hardwarové. Organizace NIST hovoří o důležitosti použití IDS systému v souvislosti, že není otázkou,

zda implementovat, ale který systém implementovat. Dále uvádí důvody k pořízení a užívání systémů IDS:

- pomáhá předcházet útokům v rámci organizace, zvýšením rizika odhalení pokusu o vniknutí.
- Detekuje útoky a jiné bezpečnostní narušení, které neodhalily jiná opatření.
- Detekují a vypořádávají se s předpoklady dalších útoků, jako například síťové sondy.
- Slouží k dokumentace současných existujících hrozeb organizaci.
- Slouží jako kontrola kvality pro současné navržené a implementované bezpečnostní opatření.
- Poskytuje užitečné informace o vniknutí [14].

IPS

Je systém monitorující síť z důvodu potenciálního nebezpečí. Dosažení nalezení hrozby za pomoci sledování škodlivých incidentů a následně sběrem relativních dat. Po detekci a sběru dat jsou data reportována systémovým administrátorům, kteří na jejich základě mohou provést bezpečnostní opatření jako doplnění tzv, „black listů“ na firewallech, nebo zamezení některým uživatelským účtům v přístupu atp. Hlavním rozdílem oproti IDS je schopnost provádět akce, dalo by se tak říct, že IPS je aktivním IDS. Hlavním úkolem IPS systémů je zabraňovat hrozbám jako:

- DOS útok,
- *Distributed Denial of Service* (DDOS) útok,
- různé ruhy exploitů,
- malware typu Worm,
- počítačové viry [15].

Firewall

Účelem firewallu je monitorovat provoz do sítě a ven ze sítě. Pokročilejší firewally jsou schopné provoz prohlížet více do hloubky a analyzovat tak typ síťového provozu. Hlavním výstupem z hlediska informační bezpečnosti organizací jsou opět logy. Hlavním nástrojem firewall k dosažení bezpečnější sítě je soustava pravidel, zakládájící se na definovaných politikách jednotlivých sítí. Na základě zdrojové IP adresy, cílové IP adresy a portu tak může rozhodnout například o blokování příchozího provozu. Taková akce je poté hlášena pomocí logu, například SIEM systému Splunk. Z toho vyplývá, že síla firewallu se zakládá na síle definovaných pravidel. Pro podporu kvality je pak vhodné implementovat automatické nástroje, které jsou schopny pravidla spolehlivě aktualizovat na základě různého sledování systému. Kdy a proč je logování na firewallu užitečné:

- pro náhled zda pravidla na firewallu opravdu fungují. Podle záměru je důležité mít správně nastavené logování.
- Pro odhalení škodlivých aktivit v síti, přesto že poskytuje pouze omezené informace.
- Pro nástavbu korelačních pravidel například jiným systémem, který rozhodne o blokování zdroje opakovaně. neúspěšných událostí.
- Odhalení vnitřních spojení pro použití při útoku na systém třetích stran [16].

Windows

Systémy společnosti Microsoft mají schopnost provádět detailní logování. Ve výchozím nastavení je logování zakázáno, ale pokud je administrátorem firmy povoleno, může poskytovat mnoho užitečných informací o uživatelově systému. Jednou z hlavních oblastí pak může být logování povolených a blokovaných spojení se systémem. Takovéto logy pak obsahují informace jako:

- čas a datum připojení,
- povolení **allow** nebo zamítnutí **drop** spojení,
- typ spojení *Transmission Control Protocol* (TCP), nebo *User Datagram Protocol* (UDP),
- zdrojovou *Internet Protocol* (IP) adresu spojení, cílovou IP adresu a port,
- informace zda se jedná o upload či download.

Společnost Exabeam na svých stránkách dále uvádí seznam podezřelých událostí na firewallu:

- povolení autentizace,
- zamítnutí provozu,
- spuštění, zastavení, restartování firewallu,
- modifikace konfigurace firewallu,
- povolení přístupu administrátorovi,
- selhání autentizace,
- administrátorovo spojení ukončeno [16].

Linux

Operační systém linux používá filtr paketů jménem **netfilter**. Ten je používán k povolení, zahození, nebo modifikaci paketů, které přichází či odchází z nebo do systému. Dalším nástrojem je **iptables**, což je firewall na softwarové úrovni, který uživateli umožňuje sestavit vlastní pravidla provozu, povolovat či zamítat IP adresy porty atp. Nástavbou nad **iptables** je **firewalld**, který schopnosti iptables navyšuje

o další funkcionalitu a zvyšuje přehlednost pravidel. Interpretace logů firewallu vyžaduje povolení logování kernelu systému. Ve výchozím nastavení jsou logy zapisovány do složky **/var/log/messages** [16].

1.5 Systémy pro automatické řešení

Tyto systémy jsou obecně označeny zkratkou SOAR a slouží zejména k usnadnění práce analytiků firem. Dále však slouží k podpoře bezpečnosti organizace, zejména pak možnosti reagovat na události a incidenty automaticky, a tím také značně rychleji než by kdy mohl uživatel. Dále zpřehledňuje vyšetřování kybernetických událostí a incidentů a poskytuje možnost provádět akce. Kosť ve svém článku uvádí, že systémy SOAR v sobě spojují tři věci:

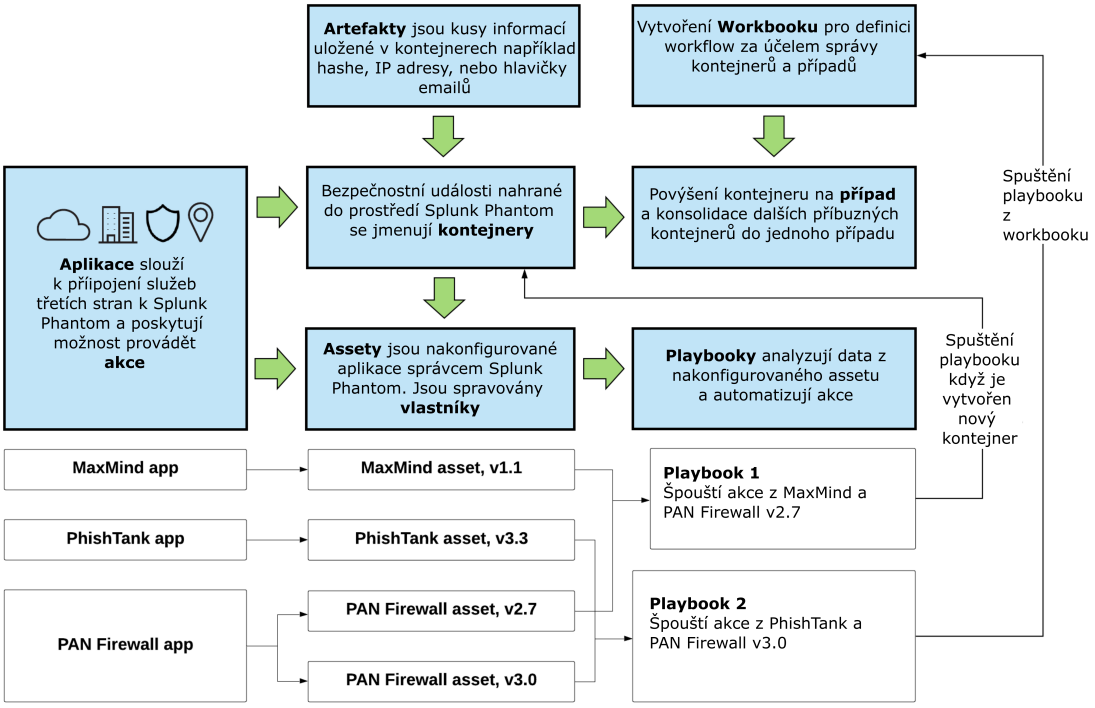
- orchestraci – integraci a propojení různých bezpečnostních a ne-bezpečnostních nástrojů a úkonů,
- automatizaci – automatizované provádění činností nad orchestrovanými nástroji,
- reakci – pomocí automatizované orchestrace provádíme jednotlivé součásti incident response procesu [17].

Systém tak slouží zejména pro podporu reakce na události, a také k měření výsledků. Další výhodou je pak možnost automatického generování hlášení, na základě zákonu o kybernetické bezpečnosti. Hlavním předpokladem dobře fungujícího systému jsou pak dobře vytvořené scénáře, které počítají, a umí pracovat s širokou oblastí dat. Mimo jiné je hlavními přednostmi úspora času, peněz, lidských zdrojů a zvýšení bezpečnosti organizace. Hlavními představiteli řešení z oblasti SOAR jsou Splunk Phantom od společnosti Splunk, FortiSOAR od společnosti fortinet, Cortex XSOAR společnosti Paloaltonetworks, SOAR společnosti Swimlane [18] [19] [20].

1.6 Splunk Phantom

Je SOAR systém kombinující orchestraci bezpečnostní infrastruktury, automatizaci scénářů a management událostí pro integraci v rámci bezpečnostního týmu. Hlavním úkolem řešení společnosti Splunk je tak pomoc při řešení bezpečnostních incidentů v rámci organizace. Popis hlavních součástí řešení Splunk Phantom viz obrázek 1.3. Dále Splunk definuje důležité prvky a pojmy takto:

- **Aplikace** přináší konektivitu k zabezpečovacím technologiím třetích stran. Spojení umožňuje Splunk Phantomu přistupovat a spouštět akce na připojených technologiích. Některé aplikace mají vizuální prvky umožňující vizualizovat data.



Obr. 1.3: Blokový diagram softwaru Splunk Phantom

- **Asset** je specifickou instancí aplikace, každý z nich reprezentuje fyzické nebo virtuální zařízení organizace jako například servery, koncové body, routery, firewally, atp. Například je možné nakonfigurovat asset pro každý firewall v síti.
- **Kontejner** je bezpečnostní událost nahraná na Splunk Phantom. V základu má kontejner popis události a slouží k shlukování podobných událostí k sobě.
- **Případ** je speciálním kontejnerem, který dokáže udržovat jiné kontejnery. Slouží tak ke konsolidaci vyšetřování.
- **Artefakt** je informace přidaná do kontejneru, jako například soubor, hash, IP adresa nebo hlavička emailu.
- **Indikátor** nebo ***Indicator of Compromise*** **(IOC) data** jako například IP adresa, jméno zařízení, nebo hash souboru. Jedná se o nejmenší jednotku dat Splunk Phantom.
- **Playbook** definuje sérii automatizovaných úloh, které je možné provést nad příchozími daty. Může být spouštěn například na základě popisu událostí.
- **Workbook** je šablona poskytující seznam běžných úloh, které analisté mohou následovat při hodnocení kontejneru nebo případu.
- **Akce** je možnost vykonat automatickou akci na zařízení třetí strany jako blokování IP adresy, uspání virtuálního zařízení, ukončení procesu atp.
- **Vlastník** je osoba zodpovědná za správu assetů v organizaci [21] [22].

2 Praktická část

2.1 Experimentální pracoviště

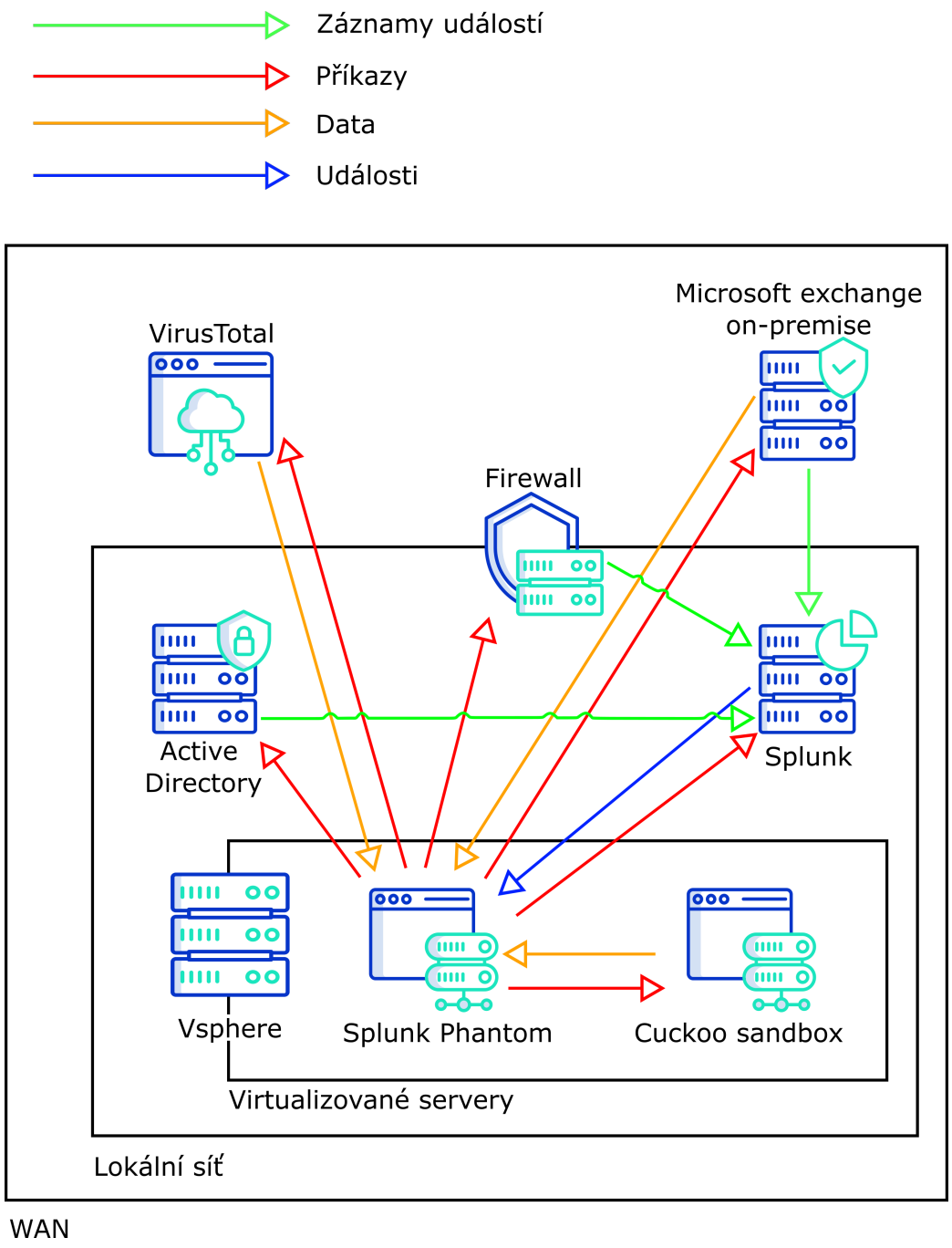
V rámci vlastní implementaci Splunk Phantom bylo vytvořeno experimentální pracoviště viz obrázek 2.1. Toto pracoviště zasahuje do tří síťových vrstev. První vrstvou je *Wide Area Network* (WAN) jejíž součástí je vzdálený server VirusTotal. Ten slouží Phantomu jako zdroj dat potřebných k **analýze** a jeho funkce je nakonfigurována příslušným assetem popsaným dále v části práce 2.2.1. Dalším prvkem spadajícím do této oblasti je Microsfot exchange on-premise. Ten slouží jako zdroj dat nejen pro Phantom, ale také pro Splunk, který se stará o vyhodnocování nebezpečných událostí aplikací Splunk Enterprise Security. Popis assetu starajícího se o funkčnost komunikace mezi Splunk phantom a exchange serverem je k nalezení v části práce 2.2.1.

Druhou vrstvou je *Local Area Network* (LAN), tedy vnitřní síť. Na rozhraní WAN a LAN leží firewall, jehož prvotním účelem je detekovat možné útoky a hlásit je Splunk serveru. Následně je možné pomocí Splunk Phantom ovládat například black listy tohoto zařízení a dynamicky reagovat na případné hrozby. Server Active Directory slouží pro odesílání záznamů událostí na server Splunk, aby zde byly vyhodnocovány potenciálně nebezpečné události jako například pokusy o přihlášení ze zvláštních lokalit, nebo opakované neúspěšné pokusy o přihlášení. Phantom je schopen ovládat Active Directory server a blokovat tak například uživatelské účty. Druhým prvkem v rámci LAN sítě je server Splunk, jehož funkcí je shromažďovat záznamy událostí a následně na základě definovaných korelačních pravidel nebo signatur označovat a hlásit události serveru Splunk Phantom. Ten je také schopen se serveru dodatečně dotazovat a požadovat například doplňující informace k události pomocí *Search Processing Language* (SPL) dotazu.

Poslední vrstva je virtualizované prostředí spravované pomocí řešení Vsphere. V rámci této vrstvy je implementován samotný Splunk Phantom a s ním také Cuckoo sandbox server. Ten je zodpovědný za primární či sekundární dodávání dat z analýz pro proces **analýzy** splunk Phantom. Popis assetu ovládajícího spojení mezi těmito virtualizovanými servery v části práce 2.2.1.

2.2 Zdroje dat

V konečném důsledku slouží pro poskytnutí událostí pro Splunk Phantom, nad kterými jsou poté prováděny šetření. Aby mohla vzniknout nová událost v prostředí Phantom, je zapotřebí nejprve za pomoci korelací detekovat podezřelé události.



Obr. 2.1: Diagram částí a komunikace experimentálního pracoviště

V případě současného stavu experimentálního pracoviště se jedná o události detekované pomocí Splunk aplikace Enterprise Security. Způsob hodnocení těchto událostí systémem Splunk Phantom je založen na již definované matici záměn viz tabulka 1.1. Pro účely této práce bude jako **klasifikátor** považován systém Splunk enterprise. Splunk Phantom poté spolu s náhledem operátora slouží jako ověření tvrzení klasifikátoru tedy **skutečnost**. Předpokladem pro správný výstup scénářů je tak

skutečnost, že události zasílané systémem Splunk Enterprise jsou označeny jako **pozitivní**. Splunk Phantom tak označuje události pouze jako **true positive** nebo **false positive**.

Nastavení komunikace

Pro správné fungování experimentálního pracoviště je nutné zajistit základní síťovou komunikaci mezi SOAR platformou Splunk Phantom a ostatními systémy. Toto nastavení je také aplikovatelné v praktické implementaci. Zásadní z hlediska nastavení potřebné komunikace je povolení potřebných portů ať na logických či fyzických firewallech. Přehled prostupů v podobě portů pro Splunk Phantom viz tabulka 2.1, pro Splunk Enterprise viz tabulka 2.2 a pro Cuckoo sandbox viz tabulka 2.3.

Tab. 2.1: Přehled portů k povolení pro Splunk Phantom

Splunk Phantom			
Port	**Protokol**	**Popis**	**Účel**
22	TCP	SSH	Vzdálený přístup k serveru
80	TCP	HTTP	Nezabezpečený webový port
443	TCP	TLS	Zabezpečený webová port
389,636	TCP,UDP	LDAP	Autentizace pomocí LDAP

Tab. 2.2: Přehled portů k povolení pro Splunk Enterprise

Splunk Enterprise			
Port	**Protokol**	**Popis**	**Účel**
8089	TCP	REST	Management port
443	TCP	SSL	Nezabezpečený webový port

Tab. 2.3: Přehled portů k povolení pro Cuckoo sandbox

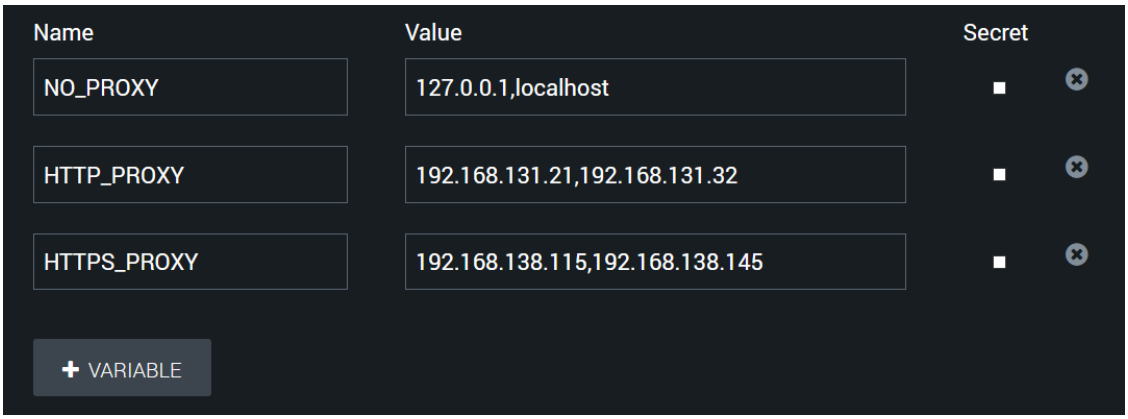
Cuckoo sandbox			
Port	**Protokol**	**Popis**	**Účel**
8080	TCP	WEB	Webová služba
8090	TCP	REST	Management port

Dále je vhodné zmínit možnost nastavení proxy konfigurace, které není součástí experimentálního pracoviště, ale pro praktickou implementaci je z hlediska komunikace toto nastavení důležité. Cesta ke globálnímu nastavení je **Administration/Administration Settings/Environment Settings**. Zde máme možnost

konfigurovat vlastní proměnné, pro účely proxy pak rozlišujeme tyto tři názvy proměnných:

- NO_PROXY
- HTTP_PROXY
- HTTPS_PROXY [23]

K jednotlivým proměnným je poté do pole **Value** nutné zapsat seznam adres oddělených čárkou: dle potřeby využití zabezpečené, nezabezpečené, nebo žádné proxy. Ukázka konfigurace globálních proměnných viz obrázek 2.2. Aby nedocházelo ke konfliktům, lze toto nastavení provést i specificky pro jednotlivé konfigurace aplikací.



Obr. 2.2: Ukázka konfigurace globálních proměnných pro účely nastavení proxy

Phantom add-on

Detekované události pomocí Splunk enterprise Seucrity jsou přeposílány do prostředí Splunk Phantom aplikací **Phantom add-On**, návod nastavení tohoto forwardingu viz dokumentace Splunk. Dle sekce *Install and Upgrade the Splunk Phantom Add-on for Splunk* této dokumentace je nejprve nakonfigurován server za pomoci autorizačního tokenu Splunk Phantom a povolen rozsah lokálních adres pro instanci Splunk [23].

Dále je třeba nastavit přeposílání jednotlivých typů událostí dle Splunk dokumentace v sekci *Use the Splunk Phantom Add-on for Splunk to Forward Events*. Nyní je nastaveno přeposílání událostí pro 3 druhy událostí detekované pomocí Splunk enterprise security. První z nich je přeposílání informací o nebezpečných souborech detekovaných na monitorovaných zařízeních. Jméno forward pravidla je **Phantom-Forward-File_Intel**, události jsou zaváděny s prametrem `label` roven

řetězci `notable`, závažnost událostí je nastavena na Low. V tomto pravidlu je namapováno 11 polí události na *Common Event Format* (CEF) pole. Toto pravidlo zasílá události k prošetření playbookem pro kontrolu souboru na vzdáleném zařízení popsaném v části práce 2.4.3. Náhled záznamu událostí s poli určenými k mapování viz obrázek 2.3 [23].

Search Fields	Value
_time	1606331411
dest	DESKTOP[redacted]local
event_id	C0F6E7BD-7D9F-4215-9C90-46B249032CD5@@notable@@26aeba768858e25e2712c2773de6dd7d
info_max_time	1606331400.000000000
info_min_time	1606327500.000000000
source	Threat - Threat List Activity - Rule
threat_collection	file_intel
threat_key	mandiant:package-190593d6-1861-4cfe-b212-c016fce1e240|Appendix_G_IOCs_No_OpenIOC.xml
threat_match_field	process
threat_match_value	wuauclt.exe

Obr. 2.3: Náhled mapovaných polí pravidla Phantom-Forward-File_Intel

Druhé pravidlo nese název **Phantom-Forward-IP_Intel** a slouží k přeposílání informací o zdroji podezřelé síťové aktivity vůči monitorovaným serverům. Pravidlo označuje události hodnotou `notable`, nastavuje závažnost na `Medium` a mapuje 13 CEF polí. Události tvořené tímto pravidlem jsou určeny ke zpracování playbookem pro kontrolu podezřelé IP adresy popsaném v části práce 2.4.2. Mapované pole k náhledu viz obrázek 2.4.

Posledním pravidlem je **Phantom-forward-email_attachment**. Jeho účelem je zaznamenat potenciálně nebezpečnou přílohu emailu. Pravidlo přeposílá události tohoto typu s označením `email` nastavuje závažnost na `Medium` a mapuje 10 CEF polí. Události tohoto pravidla jsou určeny k prošetření pomocí playbooku pro kontrolu přílohy emailu, který je popsán v části práce 2.4.1. Mapované pole viz obrázek 2.5.

2.2.1 Assety

Jedná se o konkrétní nastavení aplikace. Navigace do tohoto nastavení z hlavní nabídky, kliknutí na položku **Apps**. Zde jsou k dispozici 3 záložky: Configured Apps, Unconfigured Apps a Orphaned Assets, v každé záložce následuje závorka s číslem

Search Fields	Value
_time	1607202609
dest_dns	b[redacted]local
dest_ip	192.168.[redacted]
event_id	C0F6E7BD-7D9F-4215-9C90-46B249032CD5@@notable@@485c54121716af5c437f03bf0d7badf5
info_max_time	1607202600.000000000
info_min_time	1607198700.000000000
source	Threat - Threat List Activity - Rule
src	[redacted].236.40
threat_collection	ip_intel
threat_key	sans
threat_match_field	src
threat_match_value	[redacted]236.40

Obr. 2.4: Náhled mapovaných polí pravidla Phantom-Forward-IP_Intel

Search Fields	Value
_time	1607198439
event_id	C0F6E7BD-7D9F-4215-9C90-46B249032CD5@@notable@@27a6c8c777b6841116831c9a5fffe018
info_max_time	1607198400.000000000
info_min_time	1607196600.000000000
message_id	<CAOM9Qq1kDVK4eidq_vVaKK3GhcpCXC=ZybQhqQZoyoYhW-56gw@mail.gmail.com>
message_size	153264
message_subject	Test of label
recipient	hons@[redacted]cz
sender	[redacted]@vutbr.cz
source	Network - Suspicious Email Attachment - Rule

Obr. 2.5: Náhled mapovaných polí pravidla Phantom-forward-email_attachment

označující kolik aplikací má alespoň jeden nakonfigurovaný asset. V této implementaci je nakonfigurováno 31 aplikací převážně s jedním assetem. Ke dni publikace této práce je dalších 328 nenakonfigurovaných aplikací, což činí celkových 359 podporovaných aplikací. Kromě těchto oficiálně podporovaných aplikací lze také vytvořit vlastní využitím *Representational State Transfer* (REST) *Application Programming Interface* (API) nepodporované aplikace. Z těchto nakonfigurovaných assetů zde budou krátce popsány ty, které jsou následně využity pro vlastní implementace playbooků. Každá aplikace má vlastní rozsáhlou dokumentaci vysvětlující funkci akcí, vstupní parametry a výstupní parametry. Tato dokumentace je součástí nástroje

Splunk Phantom a nepodařilo se mi ji dohledat z jiného zdroje, proto reference na tuto dokumentaci pochází z oficiálních stránek Splunk pro registraci ke komunitní licenci [22].

Microsoft exchange on-premise EWS

Tato aplikace slouží ke komunikaci s on-premise Microsoft Ecxhange serverem, tedy emailovým serverem společnosti Microsoft. Pomocí akcí lze například dohledat emailovou schránku dle příjemce, najít a získat konkrétní email nebo smazat email. Následující seznam uvádí informace o použité verzi aplikace a aspektech nastavení assetu:

- vydavatel: Splunk,
- použitá verze: 2.0.29,
- prodejce: Microsoft.

VirusTotal

Tato aplikace slouží ke komunikaci s VirusTotal cloudem. Díky ní lze zjišťovat informace jako výsledky analýzy odeslaného souboru, stahovat soubory z VirusTotal, nebo zjišťovat reputace *Uniform Resource Locator* (URL) a souborů. Nastavení závisí pouze na vložení API klíče, který lze získat po registraci na server VirusTotal, poté jsou pod tímto registrovaným účtem prováděny akce aplikace. Následuje seznam informací o konfigurovaném assetu [24]:

- vydavatel: Splunk;
- použitá verze: 2.0.8;
- prodejce: VirusTotal.

Cuckoo

Jedná se o aplikaci umožňující komunikovat s Cuckoo sandbox serverem. Ten poté umožňuje provádět analýzy souborů a URL, jejichž detailní výsledky následně reportuje zpět. Pro konfiguraci je potřebné nastavit IP adresu serveru, port pro REST API a heslo pro přístup. Seznam informací použitého assetu v následujícím seznamu:

- vydavatel: Phantom,
- použitá verze: 1.3.17,
- prodejce: Cuckoo.

SMTP

Tato aplikace slouží k posílání emailů například z playbooku pomocí předformátované zprávy. Pro konfiguraci je nutné zadat IP adresu serveru, port a adresu odesilatele. Je zde podporována *Secure Sockets Layer* (SSL) vrstva pro zabezpečení komunikace. Základní informace o konfigurovaném assetu níže:

- vydavatel: Splunk,
- použitá verze: 2.0.14.

Splunk

Účelem této aplikace je pomocí nastaveného REST API portu komunikovat se vzdáleným serverm Splunk. Běžné využití zahrnuje především možnost dohledávání obohacujících informací v rámci Splunk Eneterprise, nebo upravování událostí v daném systému. Následuje seznam obsahující základní informace o použité verzi:

- vydavatel: Splunk,
- použitá verze: 2.1.3,
- prodejce: Splunk Inc.

VirusTotal v3

Jedná se o nově vydanou aplikaci podporující odlišný způsob práce s nahráváním vzorků pro analýzu a získávání dat zpět do Splunk Phantom. Využití této aplikace je stejné jako v případě starší aplikace popsané v části práce 2.2.1. Důvodem použití aplikace v této práci je vyzkoušení nových funkcionalit a určení, která z aplikací nabízí lepší a spolehlivější práci s daty. Následuje krátký seznam informací o použité verzi:

- vydavatel: Splunk,
- použitá verze: 1.0.2,
- prodejce: VirusTotal.

FortiGate

Tato aplikace slouží pro komunikaci s firewallem od společnosti Fortinet. Umožňuje Phantomu získávat současné politiky blokací a whitelistů na firewallu, blokovat a odblokovat konkrétní IP adresy. Dále je uveden seznam použité verze a základních informací aplikace:

- vydavatel: Splunk,
- použitá verze: 2.0.0,
- prodejce: Fortinet.

Phantom

Aplikace poskytuje možnost ovládání API rozhraní lokální či vzdálené instance Splunk Phantom. Mezi mnoha akcemi podporovanými touto aplikací lze nalézt například tvoření či export kontejnerů, přidávání položek na vytvořené seznamy, hledání nových artefaktů, nebo přidání artefaktů do existujících kontejnerů. Aplikace proto slouží jako kontrolní rozhraní Phantom instance, které je připraveno pro uživatelsky přívětivé začlenění do playbooku. Základní informace o použitém assetu v následujícím seznamu:

- vydavatel: Splunk,
- použitá verze: 3.1.2,
- prodejce: Phantom.

Windows Remote Management

Tato aplikace slouží k integraci služby Windows Remote Management a nabízí 23 akcí pro užití. Obecně jsou podporované akce reflektovány ze samotného nástroje společnosti Microsoft, který slouží především pro spolupráci hardwaru a operačních systémů různých výrobců. Z hlediska bezpečnosti jsou pak nejzajímavější akce jako získání souboru ze vzdáleného zařízení, odhlášení vzdáleného uživatele, upravení koncového firewallu stanice, vypnutí procesu, vypnutí systému atd. V seznamu níže jsou uvedeny základní informace o nakonfigurovaném assetu [25]:

- vydavatel: Splunk,
- použitá verze: 2.0.1,
- prodejce: Microsoft.

V rámci nasazení systému SOAR je nutné znát dopředu, jaké aplikace budou využity, a to zejména za účelem přípravného procesu přístupových práv k jednotlivým systémům cílové infrastruktury. Z tohoto důvodu je zde uvedena tabulka přehledu aplikací a akcí pro jednotlivé playbooky viz tabulky 2.4 kontrola souboru, 2.5 kontrola IP adresy a 2.6 kontrola souboru na vzdáleném zařízení.

Tab. 2.4: Přehled assetů a akcí playbooku pro kontrolu souboru

Kontrola souboru	
Asset	**Akce**
Microsoft exchange on-premise EWS	lookup email, get email, delete email
VirusTotal	detonate file
Cuckoo	detonate file
SMTP	send email

Tab. 2.5: Přehled assetů a akcí playbooku pro kontrolu IP adresy

Kontrola IP adresy	
Asset	**Akce**
Splunk	update event
VirusTotal v3	ip reputation
FortiGate	block ip

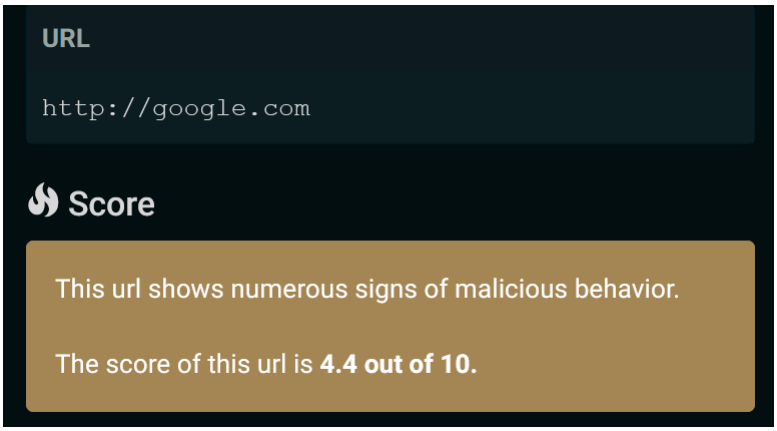
Tab. 2.6: Přehled assetů a akcí playbooku pro kontrolu souboru na vzdáleném zařízení

Kontrola souboru na vzdáleném zařízení	
Asset	**Akce**
Phantom	find artifact, update artifacts
Splunk	run query, update event
VirusTotal v3	file reputation
Windows remote management	get file, delete file
Cuckoo	detonate file

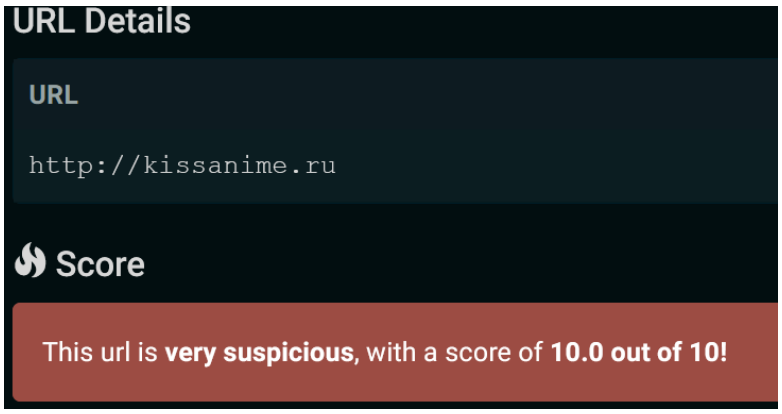
2.2.2 Cuckoo sandbox

Hlavním účelem nástroje Cuckoo sandbox je v této implementaci poskytnout primární, případně sekundární, zdroj podkladů pro automatické **vyhodnocení** událostí. Nástroj Cuckoo sandbox je instalován jako on-premise ve virtualizovaném prostředí. Hlavní výhodou pro použití s nástrojem Splunk Phantom je oficiální podpora z hlediska výměny informací konfigurovaným assetem pomocí REST rozhraní. Důraz při instalaci sandboxu byl kladen především na široké spektrum nástrojů podporující analýzu. Proto byl na serveru společně se samotným sandboxem nainstalován také open source nástroj Elastic search, který dokáže například přehledně reprezentovat síťová data analýzy, a poskytuje tak rychlý náhled na komunikaci v průběhu analýzy. Analýzy prováděné v sandboxu tvoří vysoké hodnoty score i pří neškodných souborech. Důvodem je současné nesoustředění se na odladění hlášení o potenciálně nebezpečných akcích běžných aplikací, komunikace výchozích procesů operačního systému windows, atp. Z pozorování lze určit hranici cuckoo score, které se pohybuje okolo hodnoty 5 viz obrázek 2.6, na kterém je dosaženo skóre 4.4 [26].

Dále můžeme pozorovat patrné zvýšení detekčního skóre, například při analýze stránky http://kissanime.ru, která dosahuje hodnoty 10 po kliknutí na odkaz z úvodní stránky viz obrázek 2.7. Na základě pozorování lze poté určit spodní hranici pro použití v rámci Splunk Phantom playbooků pro konkrétní typy analýz. Analýzy jsou prováděny na operačním systému Windows 7 SP1 pro simulaci nejhoršího pří-



Obr. 2.6: Skóre analýzy google.com



Obr. 2.7: Skóre analýzy kissanime.ru

padu použitého systému.

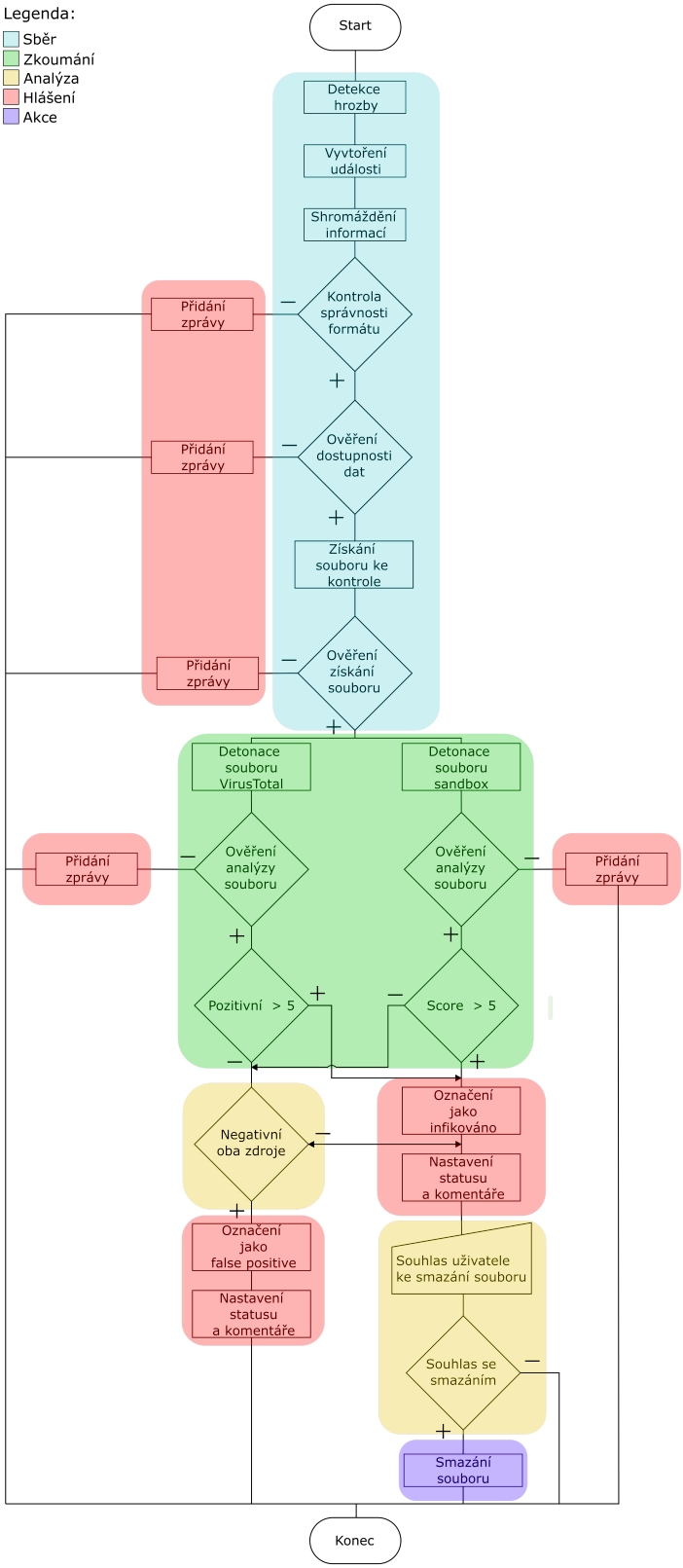
2.3 Scénáře

Hlavním cílem práce je vlastní návrh postupů řešení bezpečnostních událostí a incidentů. V této části práce jsou detailně popsány tyto vlastní návrhy postupů, které poté budou sloužit jako teoretický předpoklad k následně implementovaným formám postupů v prostředí Splunk Phantom. Vytvořené postupy mají formu scénáře, které dokumentují jednotlivé kroky potřebné k řešení daných událostí. Samotné scénáře byly také zaznamenány formou **vývojových diagramů** vytvořených pomocí aplikace Inkscape. Návrhy scénářů byly tvořeny na základě poznatků v oblasti bezpečnosti a jejich podoba je tedy vlastním návrhem postupu, který má za účel reagovat na události vzniklé v rámci sledované organizace.

2.3.1 Kontrola souboru

Úkolem tohoto vlastního návrhu postupu pro řešení bezpečnostní události ve formě scénáře je šetření události, ve které byl soubor označen jako potenciálně nebezpečný. Zásadní body jsou kontrola dostupnosti souboru pro analýzu, vyhodnocení úrovně rizika a průběžné reportování zjištění.

Nyní následuje popis jednotlivých částí a bloků vytvořeného scénáře pro kontrolu bezpečnosti souboru, vývojový diagram viz obrázek 2.8. Blok **Start** je klasickým úvodním elementem diagramu. Jeho účelem je označení začátku procesu. **Detekce hrozby** je blok, ve kterém dochází k detekci potenciální hrozby systému. Jedná se o první blok z kategorie **Sběr**. Zdrojem této informace může být například splnění podmínky korelačního pravidla SIEM systému. Po detekci hrozby je vygenerován report, jehož obsahem jsou informace důležité k šetření události. Nyní je třeba vytvořit událost. K tomu slouží blok **Vytvoření události**, jehož úkolem je vytvořit logickou strukturu, pod kterou budou dále shromažďovány důkazy, poznatky, komentáře, a další užitečné informace důležité například pro další zkoumání události odpovědnou osobou. Po úspěšném vytvoření události je čas na shromáždění informací potřebných k automatickému šetření incidentu k tomu je využit blok **Shromáždění informací**. K nejdůležitějším informacím bude v případě šetření nebezpečnosti souboru patřit cesta k souboru, zdroj dat z pohledu systému pro detekci, datum detekce, zdroj odkud soubor vzešel. Po této úvodní sérii bloků je nutné zkontrolovat zda jsou zásadní informace ve správné formátu a tím tedy použitelné pro další postup šetření. To je zajištěno rozhodovacím blokem **Kontrola správnosti formátu**. V případě zjištění problému s formátem, což běžně bývá nesprávně generovaný objekt, chyba v kódování, nepoužitelný formát objektu nebo úplně chybějící objekt, je rozhodnuto záporně a přechází se do kategorie **Hlášení**, konkrétně bloku **Přidání zprávy**, jehož úkolem je přidat zprávu do vytvořené události o této skutečnosti. To má za následek ukončení scénáře blokem **Konec**. V případě kladného rozhodnutí blokem je proces posunut do další fáze. Tou je blok **Ověření dostupnosti dat**, jehož účelem je ověření, zda se na zkontrolované cestě soubor nachází a zde je možné soubor stáhnout pro další použití, například z hlediska dostatečného oprávnění. Zde je v případě kladného rozhodnutí pokročeno do další fáze procesu, na druhou stranu v případě záporného rozhodnutí, nemá scénář stěžejní data pro svoji funkci a dochází opět k ukončení scénáře blokem **Konec** s přednostním průchodem bloku **Přidání zprávy**. Dále blok **Získání souboru ke kontrole** má na starosti využití informace o umístění souboru ke stažení do lokální databáze, zároveň soubor přiloží k události jako **artefakt**. Po stažení souboru přichází na řadu poslední rozhodovací blok kategorie **Sběr Ověření získání souboru**. Jedná se o kontrolu zda i přes již ověřenou dostupnost souboru došlo k jeho stažení a uložení do udá-



Obr. 2.8: Diagram scénáře pro kontrolu souboru

losti. Problémy, které mohou zapříčinit záporné vyhodnocení tohoto bloku a tím také ukončení scénáře spolu s hlášením, jsou náhlé problémy s připojením, nebo jiné nepředvídatelné události, ke kterým by mohlo dojít zejména v čase potřebném na vykonání tohoto bloku.

Po úspěšném ukončení fáze sběr scénář pokračuje do fáze **zkoumání**. Zde přichází na řadu dva bloky zároveň. Prvním z nich je **Detonace souboru VirusTotal** jeho úkolem je zajistit, že bude soubor nahrán na servery VirusTotal, kde dojde k jeho porovnání s databázemi antivirů a následně také obdržení výsledku počtu pozitivních označení nezávislými antivirovými společnostmi. Zároveň je proveden také blok **Detonace souboru sandbox**, cílem je nahrát soubor na předem nastavený sandbox. Po skončení analýzy blok obdrží identifikátor nebezpečnosti souboru ve formě Skóre. Scénář pokračuje kontrolou správného provedení analýzy, a to jak pro blok zajišťující získání informací ze serveru VirusTotal, tak pro blok odpovědný za komunikaci s vybraným sandboxem. K tomu slouží rozhodovací blok **Ověření analýzy souboru**, kdy při záporném vyhodnocení zapříčiněném chybějícím výsledkem analýzy, nebo chybovou hláškou o uploadu v případě nedostupnosti serveru, přejde opět k blokům kategorie **Hlášení** a následně také k ukončení scénáře. Pokud je však rozhodnuto kladně, scénář přechází k blokům jejichž účelem je rozhodnout o výsledku analýz detonačních bloků. V případě rozhodovacího bloku **Pozitivní > 5** jde o hranici nastavenou na rozhodnutí pro označení souboru jako infikovaný blokem **Označení jako infikováno**. Tuto hranici je nutno nastavit na základě zkušeností analytika. V případě záporného rozhodnutí bloku, tedy situace, kdy je počet antivirových služeb domnívajících se o nebezpečnosti souboru menší než nastavená hranice, je scénář pozastaven v prvním bloku kategorie **Analýza Negativní oba zdroje**. Toto pozastavení vyčkává na výsledek druhého rozhodovacího bloku **Score > 5**, ten má opět možnost kladně vyhodnotit výsledek získaný detonací v sandboxu a to v případě že je dosažené skóre vyšší než číslo 5 a tím se přesunout do bloku **Označení jako infikováno**. Blok **Negativní oba zdroje** tak vyčkává, zda získá dvě záporné vyhodnocení předchozích vyhodnocovacích bloků. Pokud k tomu nedojde, předpokládá se že jeden z výsledků je pozitivní, a při vyslání potvrzení z bloku **Označení jako infikováno**, je tato větev ukončena přesunem k větvi, jejímž výsledkem bude označení souboru jako infikovaný. Ještě je zde jedna možnost, a to že blok **Negativní oba zdroje** dostane dvě záporné vyhodnocení a tím se přesune k jeho finální fázi označení události jako false positive blokem **Označení jako false positive**. Po něm přichází na řadu finální krok kategorie **Hlášení** této větve, jenž má na starosti blok **Nastavení statusu a komentáře** po jehož úspěšném provedení je událost úspěšně uzavřena.

Situace kdy se scénář přesune k bloku **Nastavení statusu a komentáře** ve větvi obsahující blok **Označení jako infikováno** vyústí k přesunu k dalšímu z bloků

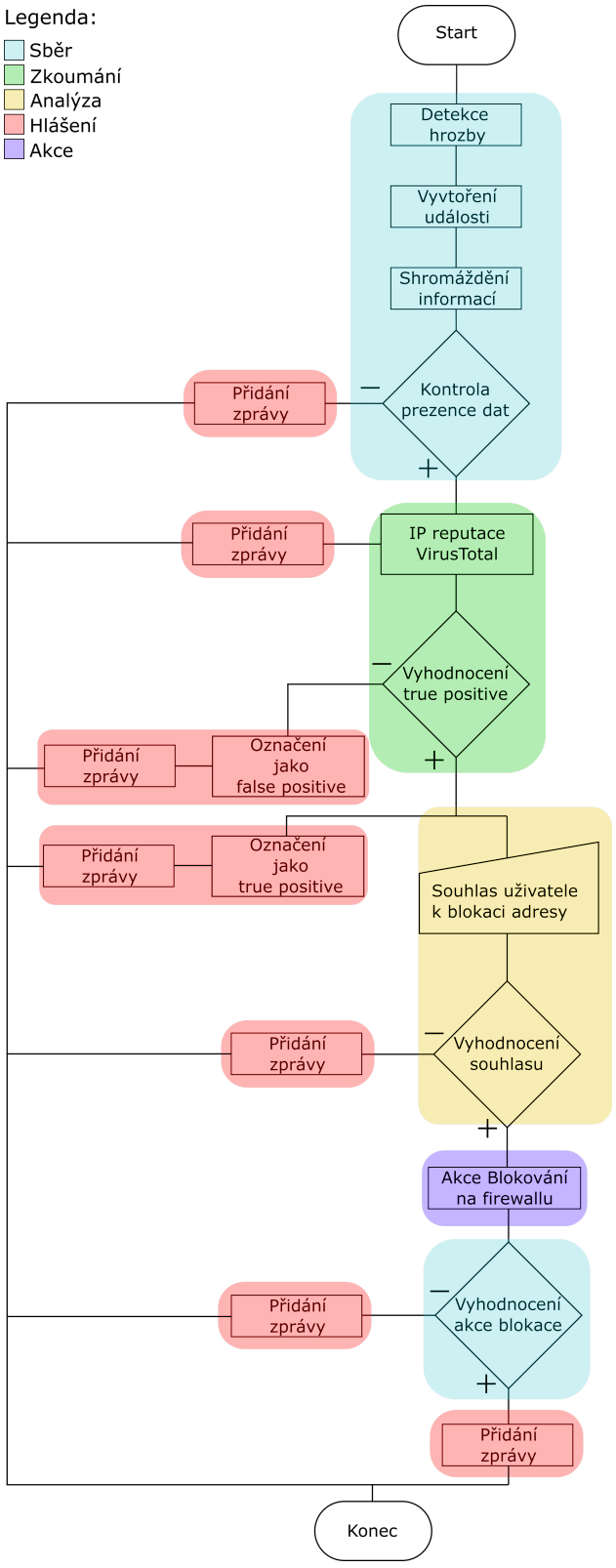
kategorie **Analýza** a to **Souhlas uživatele ke smazání souboru**. Účelem hlášení v této větvi je veškeré skutečnosti z kategorie **Analýzy** zapsat k příslušné události tak, aby následně mohl uživatel reagovat na výzvu ke smazání souboru co možná nejrychleji. Dále má za úkol tato větev zapsat závažnost celé události a případně na základě nastavených podmínek povýšit událost na případ. Zároveň je možné nastavit další ukazatele jako například závažnost celého incidentu atp. Uživatel následně zhodnotí informace získané automatickým scénářem a odešle odpověď. Tuto odpověď vyhodnocuje blok **Souhlas se smazáním**, v případě záporné odpovědi je ponechána událost ve stavu odpovídajícím nastaveným podmínkám bloku kategorie **Hlášení** a přechází do konečného bloku **Konec**. V případě kladného vyhodnocení uděleným souhlasem analytika se přechází k poslední a nejdůležitější kategorii automatického řešení incidentu **Akce**. Tou je v případě tohoto scénáře smazání souboru označeného jako infikovaný blokem **Smazání souboru**. Po jeho vykonání se přechází k bloku **Konec**, tím je celý scénář vykonán i poslední možnou cestou.

2.3.2 Kontrola IP adresy

Tento scénář, který byl navržen v rámci vlastního přínosu diplomové práce slouží k prošetření síťové události, ve které je důvodné podezření na možné narušení bezpečnosti. Nejzávažnější události jsou takové, kde se některá z lokálních IP adres snažila přistupovat na vnější IP adresu, která je dle informací získaných ze zdrojů třetích stran označena jako nedůvěryhodná. V tomto případě se může jednat o komunikaci stanice umístěné ve vnitřní síti infrastruktury s *Command and Control* (CaC) serverem, takovéto události pak musejí být šetřeny s vysokou prioritou. Druhou možností využití tohoto scénáře je automatická kontrola veškerých vnějších adres, které se snaží připojit na adresy vnitřní sítě. V tomto případě se jedná o běžně se vyskytující události a události mohou být řešeny s nižší prioritou. V každém případě z hlediska bezpečnosti je vhodné při zjištění, že se podezřelá vnější adresa snaží připojit na vnitřní adresu, takovouto komunikaci zablokovat a neumožnit potenciálnímu útočníkovi provádět akce jako například sken otevřených portů nebo využití síťových útoků nultého dne.

Nyní k popisu jednotlivých částí a bloků scénáře pro kontrolu podezřelé IP adresy, vývojový diagram viz obrázek 2.9. Začátek diagramu značí blok **Start** po němž následuje první blok z kategorie **Sběr** s názvem **Detekce hrozby**. Úkolem tohoto bloku je při implementaci detekovat podezřelou komunikaci, ve které je jedna z IP adres komunikujících stran označena třetí stranou jako podezřelá. K tomuto účelu může v praxi sloužit například veřejný nebo privátní seznam takovýchto IP adres spravovaný třetí stranou. Jedním z příkladů je seznam instituce SANS [27].

Po úspěšné detekci hrozby přichází na řadu vytvoření události v příslušném sys-



Obr. 2.9: Diagram scénáře pro kontrolu IP adresy

tému typu SIEM nebo jemu podobném. V přímé návaznosti poté následuje třetí blok kategorie **Sběr** s názvem **Shromáždění informací**. V tomto kroku je při implementaci nutné zahrnout veškerá relevantní data do události tak, aby byly následně k dispozici při šetření. Vzhledem k účelu scénáře pro kontrolu podezřelé komunikace jsou dvěma nejdůležitějšími informacemi *zdrojová* IP adresa a *cílová* IP adresa. Dále je důležité zajistit informace týkající se systému, který detekci provedl, artefakty které ho k označení vedli jako například prezence jedné z adres na dříve zmíněném seznamu, nebo v případě pokročilejších systémů artefakty spojené s cílem komunikace jako například skenování portů. Jako další doplňující informace pak lze využít například záznamy událostí z ostatních systémů, které o události mají vypovídající hodnotu, ze kterých je poté možno do události přidat další rozšiřující artefakty.

Poté, co je událost naplněna relevantními informacemi, přichází na řadu poslední blok kategorie **Sběr** s názvem **Kontrola prezence dat**, který by v praxi při automatické kontrole události byl implementován v systému SOAR, nebo při manuálním řešení by byla tato kontrola provedena operátorem. Pro účely této práce předpokládáme automatické řešení incidentu a proto je tento blok zodpovědný za kontrolu prezence, správnosti a úplnosti dat potřebných pro správné provedení šetření. Úvodní kontrola běžně slouží pro ověření pouze nejnutnějších informací, což by v tomto případě byla zdrojová a cílová adresa, společně s identifikací zařízení, které by mohlo provést **akci blokace**. Identifikace zařízení je však nutná pouze v případě, že se v síti nachází vícero zařízení schopných provést blokaci, zejména pak v případě rozsáhlých sítí. V případě záporného vyhodnocení prezence potřebných dat je třeba vytvořit o tomto výsledku zprávu a přiložit ji k události. V případě kladného vyhodnocení je možné přesunout se k sekci scénáře zodpovědné za **Zkoumání** události.

Prvním blokem z kategorie **Zkoumání** je blok s názvem **IP reputace VirusTotal**. Pro jednoznačnost schématu byl zvolen web VirusTotal, ale v případě reálné implementace může být tento blok doplněn nebo nahrazen jiným webem, který vede seznamy reputace IP adres. Tento blok slouží k obohacení události o informace třetí strany o podezřelé adrese. Očekávaným výstupem bloku je poté tvrzení ve zpracovatelném formátu ohledně nebezpečnosti adresy. Například pro VirusTotal se může jednat o počet označení adresy jako **podezřelá**, **nebezpečná**, **bezpečná** a **nedetekovaná**. Souhrn získaných informací je poté opět přiložen k události jako zpráva, slouží tak operátorům jako podklad k prověření události v případě manuálního ověření. S touto informací lze následně přejít k druhému bloku kategorie **Zkoumání** s názvem **Vyhodnocení true positive**. Zde je vhodné implementovat logiku kontrolující počet jednotlivých typů detekcí s důrazem na počty detekcí **podezřelé** a **nebezpečné**. Podle splnění nebo nesplnění hranic nastavených v rámci této podmínky je poté přecházeno do větvení, které událost označí jako **true positive**, nebo

false positive.

Událost je vyhodnocena jako **false positive** blokem **Vyhodnocení true positive** v případě, že není splněn minimální počet **podezřelých** nebo **nebezpečných** detekcí. Po tomto rozhodnutí následuje blok **Označení jako false positive**, který v tomto schématu zaštiťuje veškeré akce **Hlášení** nespadající do systému pro automatické řešení incidentů. V praxi by to byla například aktualizace události v jiném informačním systému a aktualizací je míněno například komentování, či změnění urgence události. Následně je provedena akce **Hlášení** v rámci systému pro automatické řešení incidentů.

Událost je vyhodnocena jako **true postive**, pokud je počet detekcí vyšší než jaká je nastavena hranice pro kladné vyhodnocení podmínky **Vyhodnocení true positive**. V takovém případě je nejprve provedeno označení události jako **true positive** blokem **Označení jako true postive** společně s blokem **Přidání zprávy** a to stejným způsobem jako v předchozím případě. Oproti false positive větvi diagramu je však na řadě doplnění o blok **Souhlas uživatele k blokaci adresy**, který je schopen informovat uživatele o dosavadním výsledku šetření společně se všemi informacemi týkajícími se události potřebnými k dalšímu rozhodnutí o blokaci IP adresy. Tento souhlas operátora je většinou podmíněn splněním časového okna nastaveného k odpovědi v případě, že uživatel není z nějakého důvodu schopen odpovědět v čas, je vhodné implementovat globální, nebo lokální podmínku, která je zodpovědná za výchozí akci povolení nebo zamítnutí dalších akcí scénáře. Dalším krokem v rámci kategorie **Analýza** je vyhodnocení odpovědi uživatele, za něž je zodpovědný blok **Vyhodnocení souhlasu**. Ten v případě záporné odpovědi provede akci **Hlášení** a zaznamená operátorovo odmítnutí blokovat IP adresu na firewallu ve formě zprávy. V případě operátorovi kladné odpovědi je na řadě provedení bloku s názvem **Akce Blokování na firewallu** z kategorie **Akce**.

Pro úspěšné provedení blokace na firewallu je potřeba znát dvě základní informace a to IP adresu k blokování a příslušný firewall na kterém má být blokace uskutečněna. V rámci tohoto diagramu je provedena pouze jedna akce, avšak běžně je na základě možností cílové infrastruktury implementovat například kontrolu zařízení v interní síti, které komunikaci inicializovalo, a to z důvodu možného kontaktování již dříve zmíněného CaC serveru. To znamená potenciální škodlivý kód na zařízení, který po jeho odhalení spadá do scénáře k šetření škodlivého kódu.

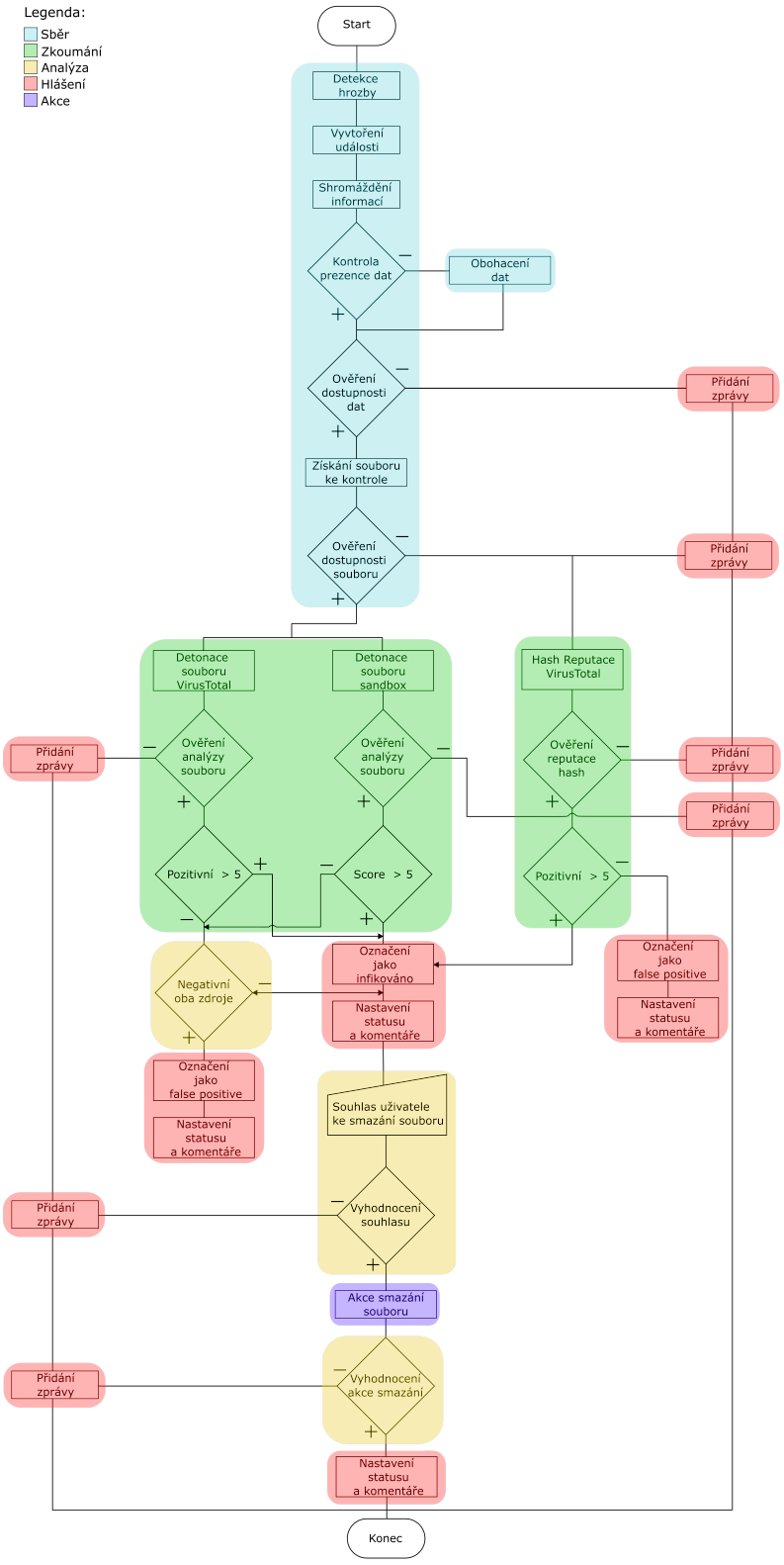
Finálním krokem je poslední blok kategorie **Sběr** s názvem **Vyhodnocení akce blokace**. Ten slouží pro ověření, že byla akce blokování provedena úspěšně. Oba výsledky vyhodnocení, ať už kladné či záporné, spouští blok **přidání zprávy**. Odlišnost pak bude v obsahu zprávy a to především při neúspěchu takovéto blokace by měla být zpráva formátována, aby vybízela k provedení akce. S tím také souvisí finální uzavření události, které může být provedeno pouze pokud je blokace na

firewallu úspěšná. Závěr scénáře je pak naznačen posledním blokem **Konec**.

2.3.3 Kontrola souboru na vzdáleném zařízení

Tento vlastní návrh postupu pro řešení bezpečnostní události byl vytvořen pro šetření událostí, kde vzniklo podezření na infikování vzdálené stanice škodlivým kódem. Události tohoto typu jsou závažné a je třeba provést šetření v co možná nejkratším čase. Východiskem pro tento scénář je detekce podezřelého chování vzdálené stanice, která může indikovat pokus o kompromitaci stanice, nebo aktivitu spojenou s prováděním neoprávněných akcí na stanici útočníkem. Samotná detekce je běžně provedena antivirovým řešením dostupným na stanici, které je poté zaznamenáváno a přeposíláno dále ke zpracování. V praxi je řešení těchto události časově náročné a to především z důvodu zhoršeného přístupu ke vzdálenému zařízení, čímž je omezena schopnost operátora reagovat na případnou hrozbu v dostatečně krátkém čase. Vzhledem k nutnosti provedení kontroly souboru, tento scénář vychází ze scénáře popsaném v části práce 2.3.1. V rámci scénáře je také naznačena akce smazání souboru, nicméně v praxi je možné volit akce jako uvalení karantény na koncovou stanici, provádění skenu zařízení, změny přístupů uživatele stanice atp.

Dále následuje popis jednotlivých částí a bloků scénáře pro kontrolu souboru na vzdáleném zařízení, vývojový diagram viz obrázek 2.10. Začátek scénáře je značen blokem **Start** po němž přichází bloky z kategorie **Sběr**. Prvním blokem zmíněné kategorie je tak **Detekce hrozby**, kdy je pomocí některého antivirového řešení zjištěn podezřelý soubor na vzdáleném zařízení. Následuje blok **Vytvoření události**, který je prováděn v rámci systému typu SIEM. Poté je blokem **Shromáždění informací** zaznamenáno co největší množství relevantních informací dostupných v daném systému zodpovědném za vytvoření události. Následně je blokem **Kontrola prezence dat** provedena kontrola prezence potřebných polí a jejich obsahů, pokud je vyhodnocení této podmínky záporné, je třeba doplnit informace k události. Při tomto typu událostí je běžné dohledávat informace v dalších informačních systémech organizace to je možné provést na začátku šetření, ale také v průběhu. Pokud je podmínka **Ověření dostupnosti dat** vyhodnocena kladně, je možné přejít k dalšímu bloku, kterým je **Ověření dostupnosti dat**. Zde je vhodné provést kontrolu dostupnosti dat například pro případ, že je třeba získat data ze zařízení nenacházejícího se uvnitř lokální sítě. V případě nedostupnosti těchto dat je nutné přejít k bloku kategorie **Hlášení**, ukončit šetření a zajistit dostupnost. Pokud jsou však data dostupná je možné přejít k bloku **Získání souboru ke kontrole**, kde je zajištěno stažení souboru ze vzdáleného zařízení, například pomocí *Windows Remote Management* (winRM) v případě infrastruktury založené na zařízeních společnosti Microsoft. Posledním blokem kategorie **Sběr** je potom **Ověření dostupnosti**



Obr. 2.10: Diagram scénáře pro kontrolu souboru na vzdáleném zařízení

souboru k analýze, kde dochází k větvení diagramu. V případě kladného rozhodnutí bloku je provedena analýza souboru pomocí dostupných nástrojů, zde je využit server VirusTotal a Cuckoo sandbox. Pro případ záporného rozhodnutí ohledně dostupnosti souboru k analýze je provedena větev kontrolující hash souboru.

Nejprve bude rozebrán případ, kdy je soubor dostupný pro další analýzu. Pro podrobný popis této části je možné nahlédnout do části práce 2.3.1, zde bude tato větev rozebrána jen zkráceně. Nejprve je současně spuštěna detonace souboru v dostupných systémech VirusTotal a Cuckoo sandbox. Poté je ověřeno provedení těchto analýz, v případě záporného vyhodnocení je provedeno **Hlášení**. Je-li potvrzen úspěch analýz, následuje vyhodnocení dle splnění hranice, zda bude soubor celkově vyhodnocen jako **false positive**, nebo **true positive**. Pokud jsou oba zdroje vyhodnoceny jako nižší než hranice, je provedeno **Hlášení** a uzavření události jako **false positive**. Pokud však alespoň jeden ze zdrojů hlásí soubor jako **True postive** je provedeno **Hlášení true postive**. Dále se scénář dostává k bloku **Souhlas uživatele ke smazání souboru**, tento postup bude popsán dále v práci.

Nyní následuje rozbor větve scénáře, starající se o provedení analýzy i v případě, že není dostupný soubor k analýze. Na úvod je paralelně provedeno **Hlášení** o nedostupnosti souboru k analýze a Kontrola reputace hashe souboru blokem **Hash Reputace VirusTotal**. Zde dochází k porovnání hashe s databází VirusTotal pro případ, že byl soubor s tímto hashem již analyzován. Úspěšnost této kontroly je poté ověřena blokem **Ověření reputace hash** patřícího do kategorie **Zkoumání**. Pro případ, že je tento kontrolní blok vyhodnocen jako nepravda, je proveden blok **Hlášení** a následuje ukončení provádění scénáře, neboť ke kontrole hash dochází pouze v případě, že není dostupný soubor k analýze. Pro případ kladného vyhodnocení podmínky je implementován další blok kategorie **Analýza** s názvem **Pozitivní $>$ 5**, kde je naznačena hranice nutná ke splnění, aby byl hash souboru označen jako **true positive**. Pokud dojde ke splnění této podmínky, je provedeno **Hlášení** typu **true positive** a následuje blok **Souhlas uživatele ke smazání souboru**. V opačném případě podmínka splněna není a je možné označit soubor jako **false positive** a následně událost uzavřít.

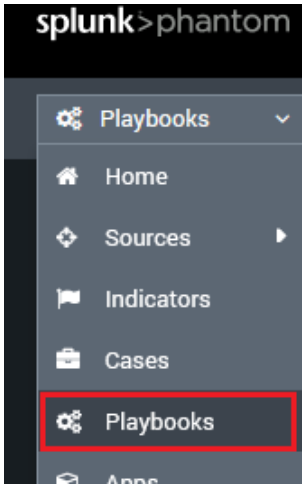
Blok spadající do kategorie **Analýza** zodpovědný za dotázání se operátora na smazání souboru na vzdálené stanici se dostává na řadu v případě, že je událost jedním ze způsobů označena jako **true positive**. V tomto bloku je implementováno zdržení na dostatečně dlouhou dobu, poskytující možnost operátorovi rozhodnout se, jak bude dále postupováno. Pro ověření operátorova rozhodnutí slouží blok s názvem **Vyhodnocení akce smazání**, který v případě záporného vyhodnocení provede **Hlášení** a uzavření šetření. Pro případ kladné odpovědi operátora je proveden blok s názvem **Akce smazání souboru** spadající do kategorie **Akce**. Pro finální **Hlášení** je třeba ověřit úspěšnost **Akce** smazání souboru. K tomu slouží blok **Vyhodnocení**

akce smazání, který v případě záporné odpovědi ohlásí neúspěch akce a upozorní tak na nutnost podniknutí nezbytných kroků před uzavřením události. Pokud je potvrzeno smazání souboru, lze provést poslední možnou akci z kategorie **Hlášení** a to blokem **Nastavení statusu a komentáře**, po němž bude událost uzavřena a vhodně okomentována. Dále následuje pouze symbolický blok **Konec** pro vymezení konce scénáře.

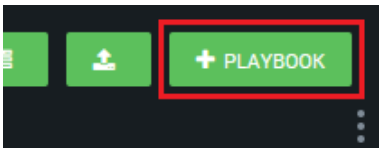
2.4 Playbook

Hlavním cílem práce je implementovat vlastní návrhy postupů pro řešení bezpečnostních událostí ve formě scénářů do funkční technologie. Playbooky představují vlastní implementaci navržených scénářů. V rámci integrace těchto postupů do funkční technologie Splunk Phantom je použito označení „playbook“. V praxi je playbook označení scénáře obsahujícího postup pro řešení bezpečnostní události, převedeného do praktického využití, ve formě počítačového kódu programovacího jazyka Python.

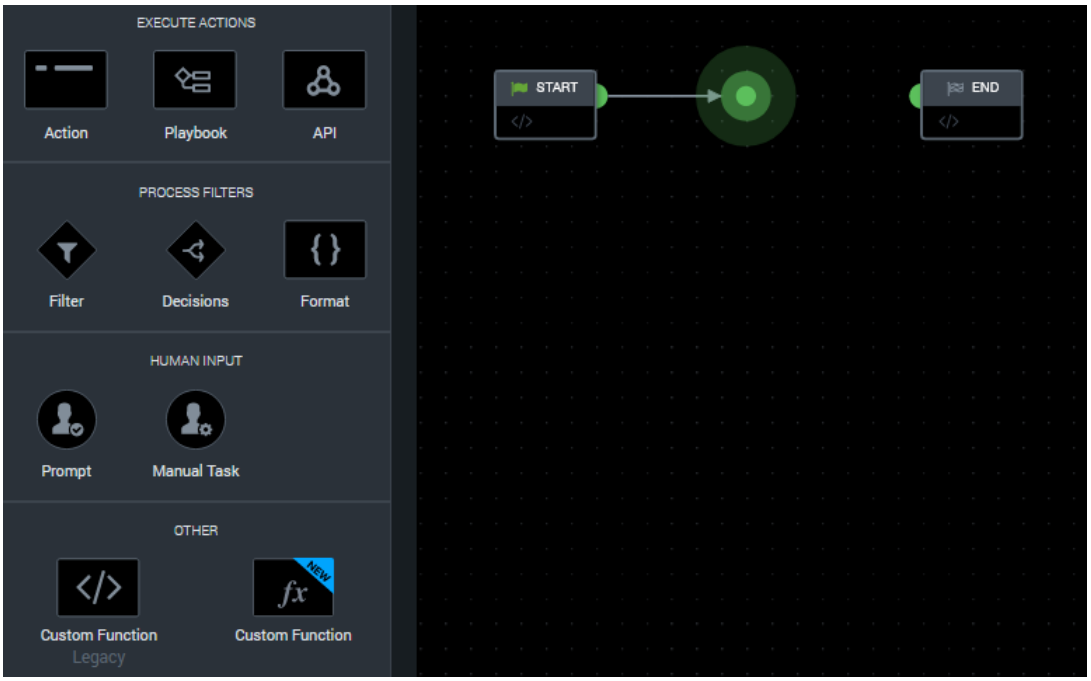
Vytvoření nového playbooku v prostředí Splunk Phantom je možné přesunem do sekce *Playbooks* viz obrázek 2.11 a následným kliknutím na ikonu *+ Playbook* viz obrázek 2.12. Zde je připraveno intuitivní prostředí pro tvorbu playbooků, vybírat můžeme z bloků Action, Playbook, API, Filter, Decisions, Format, Prompt, Manual Task, Custom Function (legacy) a Custom Function (New). Náhled tohoto prostředí je na obrázku 2.13. Zdrojové kód playbooků v jazyce pyhton a ve formátu json jsou obsahem přiloženého CD.



Obr. 2.11: Navigace k přehledu playbooků



Obr. 2.12: Tlačítko pro vytvoření nového playbooku



Obr. 2.13: Náhled prostředí pro tvorbu playbooků

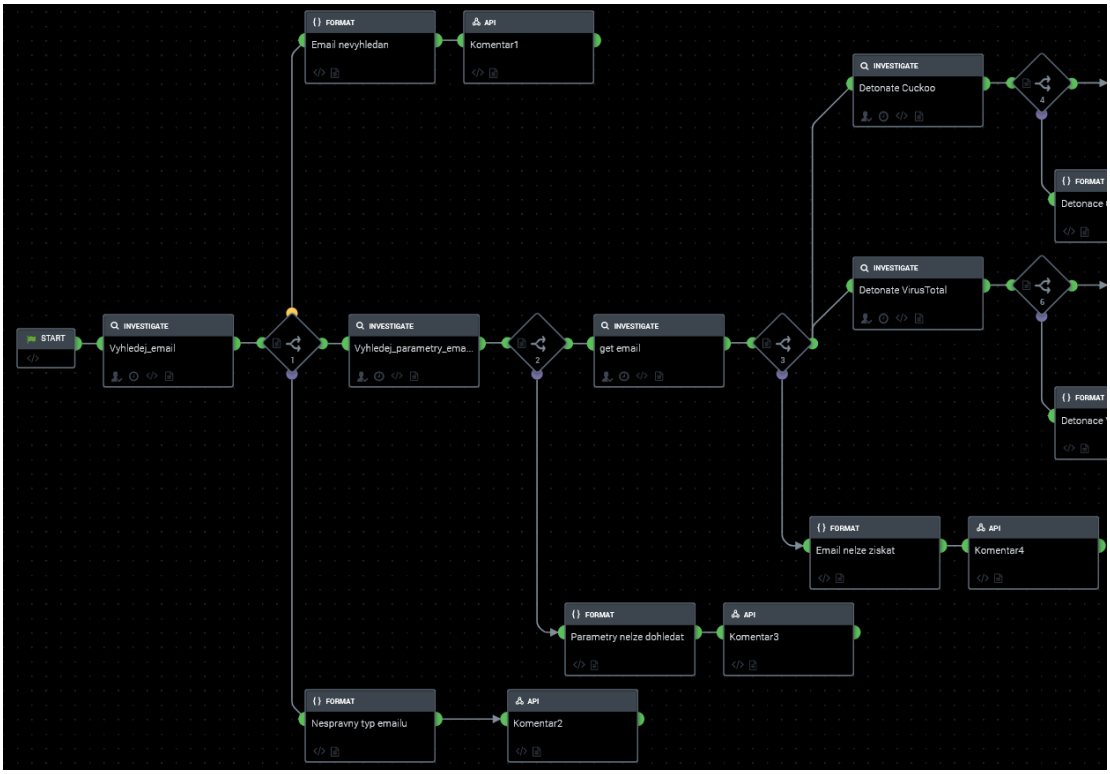
Popis playbooku bude formou rozebrání jednotlivých implementovaných bloků v rámci kategorií spolu s náhledem jejich grafické implementace pro podporu přehlednosti a návaznosti jednotlivých bloků. Rovněž bude sloužit k vyobrazení toku informací, které je nutné si při automatickém řešení incidentu playbooky předávat.

2.4.1 Kontrola přílohy emailu

Tento playbook je výsledkem vlastního návrhu řešení události, která byla detekována a označena jako **Network - Suspicious Email Attachment**, tedy jako potenciálně nebezpečná příloha emailu. Úkolem tohoto playbooku je vyjít ze scénáře popisujícího postup řešení události tohoto typu popsaného v části práce 2.3.1. Hlavními částmi playbooku tak zůstávají kategorie **Sběr**, **Zkoumání**, **Analýza**, **Hlášení** a **Akce**. Náhled celého playbooku a jeho struktury viz příloha A.1.

Nyní následuje popis playbooku pro kontrolu přílohy emailu s názvem *Kon-*

trola_prilohy_emailu_automat_DP. Tento playbook je také k dispozici na přiloženém CD viz příloha B. První část tvoří bloky a logika nutná pro přípravu a sběr dat k vyšetřování. Přehled těchto bloků viz obrázek 2.14. Playbook za-



Obr. 2.14: Náhled první části playbooku pro shromáždění a kontrolu dat

číná klasickým blokem **START**, který slouží jako grafická pomůcka označení začátku logiky. Následuje blok typu **Action** používající funkce assetu 2.2.1 popsaného výše. Úkolem tohoto konkrétního bloku s názvem **"Vyhledej_email"** je za pomocí **akce** lookup email vyhledat konkrétní emailovou schránku na základě parametru `artifact:*.cef.recipient`.

Dále je postoupeno do prvního rozhodovacího bloku, který má určit následující postup. První podmínka je vždy definována zeleně, druhá fialově a třetí žlutě. Obecný účel této podmínky je kontrola úspěšnosti minulého bloku a také ověření typu formátu emailové schránky. První podmínka je ověření zda se pole `Vyhledej_email:action_result.data.*.t_Mailbox.t_MailboxType` rovná obsahu řetězce `Mailbox`, pokud ano, je tím ověřen typ emailové schránky a také předpoklad úspěšného spuštění předchozího bloku. Je-li první zelená podmínka vyhodnocena kladně, je postoupeno k dalšímu bloku s názvem **Vyhledej_prarametry_emailu**.

Dříve než se dostaneme k popisu této funkce, je zde stále druhá podmínka prvního rozhodovacího bloku, tentokrát s fialovým označením. V ní je kontrolováno zda

se pole `Vyhledej_email:action_result.status` rovná řetězci `success` a zároveň není-li obsahem pole `Vyhledej_email:action_result.data.*.t_ Mailbox.t_MailboxType` řetězec `success`. Tím je ošetřen případ, kdy je emailová schránka dohledána, ale její typ nesedí a tím pádem není vhodný pro tento playbook. V případě kladného vyhodnocení této podmínky se playbook přesouvá k **formátovacímu** bloku s názvem **Nespravny typ emailu**. V tomto bloku je formátována zpráva informující o zdařené operaci vyhledání emailu s přiloženým obsahem pole `Vyhledej_email:action_result.status`. Zároveň je obsahem zprávy informace o špatném typu schránky s vloženým polem `Vyhledej_email:action_result.data .*.t_Mailbox.t_MailboxType`. Aby byla zpráva s tímto formátováním přidána do příslušné události ve formě komentáře, je třeba implementovat poslední blok fialové větve. Tím je **API** blok s názvem **Komentar2**. Zde je použito API s názvem **add-comment**, obsahem komentáře je potom pole `Nespravny_typ_emailu:formatted_data`. Zde bych rád podotkl, že Splunk Phantom v použité verzi 4.9.33153 jeví zvláštní chování v případě výběru pole formátovaných dat ukončené hvězdičkou například `Nespravny_typ_emailu:formatted_data.*`. V takovém případě je přidaný komentář nečitelný a Python se projevuje chybovou hláškou.

Poslední větví rozhodovacího bloku je žlutá podmínka, která má za úkol ověřit, zda se pole `Vyhledej_email:action_result.status` rovná řetězci `failed`. Pokud tomu tak je a podmínka je vyhodnocena kladně, playbook se přesouvá k **formátovacímu** bloku **Email nevyhledan**. Zde je naformátována hláška o neúspěšném vyhledání emailu spolu s dynamickým obsahem pole `Vyhledej_email:action_result.status` pro kontrolu. Pro vepsání opět následuje API blok **Komentar1**, který jako zdroj dat pro API add comment používá pole `Email_nevyhledan:formatted_data`.

Nyní se vraťme k bloku **Vyhledej_parametry_emailu** jedná se o další blok patřící do kategorie **Sběr**. Typem bloku je **Action**, zvolená aplikace **Microsoft Exchange On-Premise EWS**, vybraná akce **run query**, konfigurace assetu pro paramter **email** je `Vyhledej_email:artifact:*.cef.recipient`, jedná se tak o data získaná blokem **Vyhledej_email** konkrétně pole **recipient**. Dále pro parametr `folder` je použito `Vyhledej_email:artifact:*.cef.threat_key` opět získáno pomocí předchozího bloku pro **sběr**, stejně jako data pro pole `internet_message_id`, jež je ve formátu `Vyhledej_email:artifact:*.cef.internet_message_id`. Posledním parametrem je `range`, který je nastaven na 0-10. Účelem tohoto bloku je, najít informace o konkrétním emailu z vyhledané emailové schránky na základě definovaných parametrů, v tomto případě se jedná o **Simple Search**.

Následuje kontrolní blok číslo dvě, který má dvě definované podmínky. První z nich se zeleným barevným označením, kontroluje obsah pole `Vyhledej_parametry_emailu:action_result.status` oproti řetězci `success`, rovná-li se tato podmínka, je vyhodnocena kladně a je možné postoupit k dalšímu bloku **get email**. Před popi-

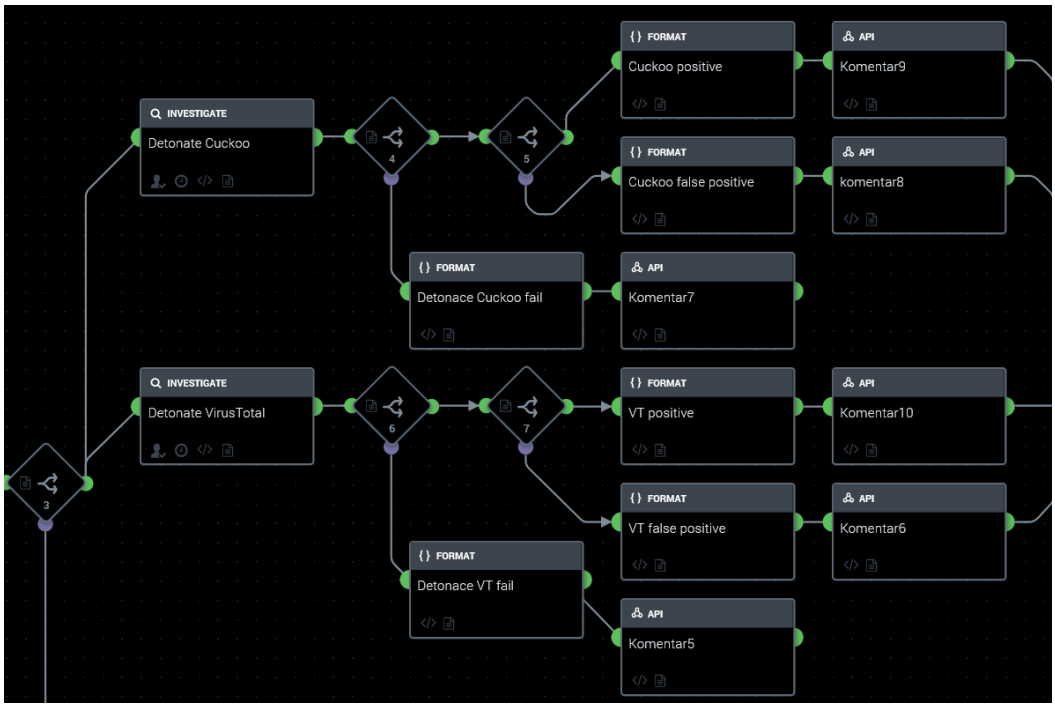
sem dalšího bloku je zde také možnost kladného vyhodnocení druhé podmínky s fialovou barvou. Zde se kontroluje stejné pole jako v podmínce první, avšak nyní se porovnává s řetězcem `failed`. Je-li vyhodnocena kladně, playbook se přesune k formátovacímu bloku **Parametry nelze dohledat** a vytvoří zprávu složenou ze tří dynamických polí. Tyto pole jsou `Vyhledej_parametry_emailu:artifact:*.cef.recipient`, `Vyhledej_parametry_emailu:artifact:*.cef.threat_key}` a `artifact:*.cef.internet_message_id`. Formátovaná zpráva tedy slouží k zapsání informace o chybějícím či nesprávném parametru do komentáře, případně pro urychlení nalezení příčiny selhání playbooku. Takto naformátovaná zpráva je následně zapsána API blokem **Komentar3** za pomoci funkce add comment.

Posledním **Action** blokem sloužícím ke **sběru** dat je blok s názvem **get email**. Ten opět využívá aplikaci **Microsoft Exchange On-Premise EWS**, spolu s akcí **get email**. Jejím účelem je parsing dat a je-li specifikováno také vytváření kontejnerů a artefaktů. K vytvoření těchto kontejnerů slouží Parametr **ingest_email**, ten je v případě tohoto bloku nastaven na hodnotu **True**, stejně jako parametr **use_current_container**, to zapříčiní vytvoření artefaktu v rámci stejného kontejneru. Pro nalezení správného emailu je použit parametr **id**, tím je docíleno extrahování dat ze vzdáleného serveru, tvar obsahu parametru je `Vyhledej_parametry_emailu:action_result.data.*.t_ItemId.Id`.

Kontrolu poslední fáze **sběru** dat provádí kontrolní blok číslo tři. Zde jsou definovány dvě podmínky. První, zelenou podmínkou, se ověřuje obsah pole `get_email_1:action_result.status` je-li rovno řetězci `success`, je možné přejít k další fázi **zkoumání**, kde jsou bloky **Detonate Cuckoo** a **Detonate VirusTotal**. Ovšem vpřípadě kladného vyhodnocení druhé fialové větve, která testuje zda je obsahem stejného pole jako ve větvi jedna, řetězec `failed`. Pokud ano a playbook tak pokračuje k bloku typu **format** s názvem **Email nelze získat**, je sestavena zpráva s obsahem informujícím o této skutečnosti, doplněné o obsah testovaného pole *status*. Poté je API blokem `Komentar4` tato zpráva zapsána do komentáře. Tímto je fáze **sběru** dat ukončena.

Nyní následuje fáze **zkoumání** dat. Náhled části playbooku viz obrázek 2.15. Pokud se playbook úspěšně dostane do této fáze, následuje spuštění dvou bloků najednou, jedná se o bloky **Detonate Virus** a **Detonate Cuckoo**.

Jako první bude popsána větev sloužící k analýze přílohy na serveru VirusTotal. První blok v této větvi je typu **action** využívající assetu popsaného v části práce 2.2.1, s použitím funkce **detonate file**, jejímž jediným parametrem je `vault_id`. Do tohot parametru je vložen obsah pole kontejneru `artifact:*.cef.vaultId`. Toto pole označuje jedinečně soubor přílohy, získaný v sekci bloku **get email**. Rád bych zdůraznil poslední úpravu tohoto bloku. Jedná se o doplnění parametru `scope='all'` do funkce **phantom.collect2**, bez jehož doplnění automatické



Obr. 2.15: Náhled druhé části části playbooku sloužící pro zkoumání dat

spouštění playbooku opakovaně končí chybou *The given parameters look like they were automatically generated by phantom.act() because an empty parameters list was passed to phantom.act()*. Důvodem této chyby je vytvoření nových artefaktů funkcí **get email**, které potom nejsou při automatickém spuštění kontrolovány a tím dojde k zmíněné chybě. Celá podoba funkce viz výpis 2.1.

Výpis 2.1: Doplnění parametru scope

```
container_data = phantom.collect2(container=container,
    datapath=['artifact:*.cef.vaultId', 'artifact:*.id'],
    scope='all')
```

Tímto blokem je docíleno odeslání vzorku na server ViusTotal k analýze. V případě že dojde k opakované analýze stejného vzorku, analýza nebude provedena znovu, ale budou staženy již existující výsledky analýzy.

Nyní je rozhodovacím blokem číslo šest kontrolováno pole `status` předcházející operace, konkrétní tvar je `Detonate_VirusTotal:action_result.status`. První z podmínek se zeleným barevným označením porovnává obsah pole s řetězcem `success`, rovnají-li se, je rozhodnuto o úspěchu akce a je pokračováno k rozhodovacímu bloku číslo sedm. V případě, že je splněna druhá z podmínek s fialovým barevným označením, je na řadě **format** blok informující o neúspěšné detonaci spolu s dynamickým obsahem testovaného pole `success`. Takto formátovaná zpráva

je **API** blokem **Komentar5** přidána k události ve formě komentáře. Rozhodovací blok číslo sedm spadá do kategorie **Analýza** a bude popsán dále v textu.

Druhá popisovaná větev začíná blokem **Detonate Cuckoo**. Obdobně jako v předešlých případech se jedná o **action** blok, který využívá assetu popsaného v části práce 2.2.1. Využívá akce **detonate file** a vstupem jsou dva parametry, prvním z nich je `vault_id` stejně jako při analýze s využitím VirusTotal serveru, slouží k identifikaci analyzovaného souboru v rámci kontejneru a obsahem je tak `artifact:*.cef.vaultId`. Druhým volitelným parametrem je `file_name` jako vstup je nastaven obsah pole `get_email_1:artifact:*.cef.fileName`, který bude následně použit jako označení souboru na serveru Cuckoo sandbox. Tento blok byl doplněn o parametr `scope='all'` ze stejných důvodů jako v případě bloku **Detonate Cuckoo**.

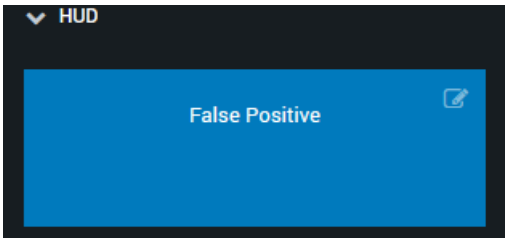
Pro kontrolu detonace je využit kontrolní blok číslo čtyři, který pomocí první zelené podmínky testuje, zda obsah pole `Detonate_Cuckoo:action_result.status` je roven řetězci `success`. V případě že ano, je postoupeno k rozhodovacímu bloku číslo pět, který však spadá do kategorie **Analýza** a bude popsán dále v textu. Druhá podmínka s fialovým označením testuje opět pole `Detonate_Cuckoo:action_result.status` zda je nerovno řetězci `success`. V případě, že je vyhodnocena kladně, přechází se k části playbooku zodpovědného za **hlášení**. Formátovací blok **Detonace Cuckoo fail** vytváří zprávu o nezdařené detonaci souboru na serveru cuckoo spolu s dynamickým polem `Detonate_Cuckoo:action_result.status`. Tato zpráva je blokem **Komentar7** následně zapsána do události ve formě komentáře.

Rozhodovací blok číslo sedm je prvním blokem kategorie **Zkoumání** a jeho účelem je analyzovat data získaná ze serveru VirusTotal. První podmínka testuje, zda pole `Detonate_VirusTotal:action_result.summary.positives`, které vrací numerickou hodnotu, je větší než číslo 5. Pokud je vyhodnocena jako kladná, pokračuje playbook zelenou větví k **format** bloku **VT positive**, který má za úkol vytvořit rychlou zprávu informující o rozhodnutí playbooku spolu s dynamickým obsahem pole `Detonate_VirusTotal:action_result.summary.positives`, jež vyjadřuje počet pozitivních označení souboru a polem `Detonate_VirusTotal:action_result.summary.total_scans`, vyjadřující počet nezávislých antivirů, které testování provedli. Tato zpráva je poté přidána **API** blokem k události. Dále následuje rozhodovací blok číslo osm. Ten bude popsán dále v textu. Druhou možností rozhodovacího bloku sedm je podmínka `else`, ta bude vyhodnocena kladně v každém případě, kdy bude počet detekcí jiný, než větší než 5. V takovém případě playbook přechází do bloku **VT false positive**, který formuje zprávu o rozhodnutí playbooku spolu s obsahem dynamických polí pro počet pozitivních detekcí a také počtu nezávislých antivirů provádějících analýzu. O zapsaní formované zpráva se stará blok **Komentar6**.

Rozhodovacím blokem číslo pět se testuje první zelená podmínka a to zda je dosaženo skóre reportované Cuckoo sandboxem vyšší než číslo 6. Pole využité k získání dynamické hodnoty je `Detonate_Cuckoo:action_result.data.*.report.info.score`. Číslo 6 bylo zvoleno z důvodů současného nastavení a filtrace sandboxu. Jeho implementace je popsána výše v části práce 2.2.2. Současný stav je z hlediska testování vhodné nastavit na hodnotu 6 z důvodu základní hladiny skóre, kterého dosahují i čisté soubory podrobené široké analýze a to skóre kolem hodnoty 5. Čísla 6 je tedy dosaženo pouze je-li zjištěno opravdu podezřelé chování. Pro další účely testování v části práce 2.5 bude tato hodnota upravována a úpravy budou popsány. Pokud je podmínka vyhodnocena kladně, postupuje playbook k bloku **Cuckoo positive**, ten vytvoří zprávu sestávající z pěti dynamických polí. Prvním polem je `Detonate_Cuckoo:action_result.status`, druhým `Detonate_Cuckoo:action_result.data.*.report.info.score` informujícím o dosaženém skóre, třetím polem je pak `Detonate_Cuckoo:action_result.summar y.id` nesoucí informaci o ID analýzy v rámci sandboxu, dále čtvrté `Detonate_Cuckoo:action_result.summary.results_url` s URL analýzy a nakonec páté pole `Detonate_Cuckoo:action_result.data.*.report.info.machine.label` vypovídající o názvu Cuckoo stroje provádějícím analýzu. Poté je formátovaná zpráva předána bloku **Komentar9**, který komentář přidá k události a přejde k rozhodovacímu bloku číslo osm. Druhá možnost vyhodnocení rozhodovacího bloku s číslem pět je splnění druhé fialové podmínky **else**, která spustí **formát** blok s názvem **Cuckoo false positive**. Tento blok poté sestaví zprávu o rozhodnutí spolu s obsahem dynamického pole `Detonate_Cuckoo:action_result.data.*.report.info.score` a pomocí bloku **Komentar8** jej zapíše k události ve formě komentáře.

Pokračováním kategorie **Analýza** je rozhodovací blok číslo osm, jehož náhled spolu s navazujícími bloky, spadající do příslušné kategorie viz obrázek 2.16. Tento blok v druhé fialové podmínce testuje pole `Detonate_Cuckoo:action_result.data.*.report.info.score` zda je menší nebo rovno číslu 6 a pole `Detonate_VirusTotal:action_result.summary.positives`, zda je menší nebo rovno číslu 5. Podmínka bude vyhodnocena pravdivě pouze tehdy, pokud bude vyhodnocen soubor jako false positive oběma testovacími nástroji. V takovém případě playbook přejde ke spuštění bloku **Pin Close Severity**, který pro událost nastaví jasně viditelnou kartu parametrem `pin type` nastavenou na hodnotu `card` a parametrem `pin color` nastaveným na `blue` pro barevnou výplně karty. Obsahem připnuté karty pak je zpráva **False Positive** nastavená parametrem `message`. Výsledek takového šetření viz obrázek 2.17. Tento **API** blok také nastavuje za pomoci akce **set severity** závažnost události na hodnotu **Low** a akcí **set status** parametrem `status` na **Closed**. V případě shody zdroje Cuckoo sandbox a VitusTotal o false positive detekci je tento blok finálním.

Obr. 2.16: Náhled třetí části části playbooku sloužící pro analýzu dat

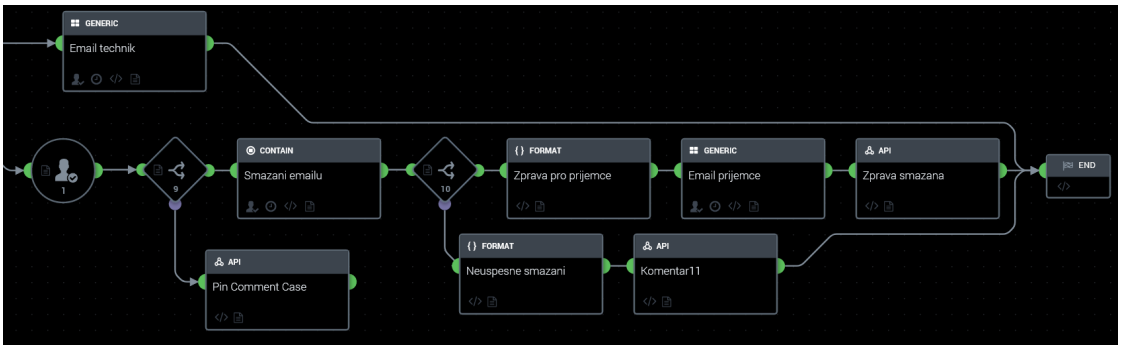


Obr. 2.17: Náhled připojené karty po vyhodnocení události jako false positive

První podmínka s barevným označením zelené barvy rozhodovacího bloku číslo osm testuje, zda je pole `Detonate_Cuckoo:action_result.data.*.report.info.score` větší než hodnota 6, nebo zda je pole `Detonate_VirusTotal:action_result.summary.positives` větší než hodnota 5. Podmínka je vyhodnocena pravdivě pokud alespoň jeden ze zdrojů analýzy souboru vyhodnotí soubor jako nebezpečný. Následně se playbook přesune ke spuštění bloku **Pin Positive**, který pomocí parametrů `pin type`, `pin color` a `message` v rámci akce **pin**, nastaví události červenou kartu s popiskem **Positive**. Dále také funkcí **set severity** nastaví závažnost události na hodnotu **High**. Playbook poté přechází k formátu zprávy, jejímž obsahem je informace o označení testovaného souboru jako pozitivní. **Formátovací** blok také přikládá obsahy dynamických polí `Detonate_VirusTotal:action_result.message` pro popis pozitivních VirusTotal detekcí, `Detonate_Cuckoo:action_result.data.*.report.info.score` k vyjádření dosaženého skóre při testování Cuckoo sand-

boxem a `Detonate_Cuckoo:action_result.summary.results_url` nesoucí odkaz s dostupnou detailní analýzou souboru sloužící jako dodatečný podklad pro další šetření.

Následující část textu popisuje poslední část playbooku, obsahujícího bloky z kategorií **analýza hlášení** a **akce**, náhled viz obrázek 2.18. První blok této části je



Obr. 2.18: Náhled poslední části playbooku

typu **Action**, využívá asssetu popsaného v části práce 2.2.1, spadá do kategorie **Akce** a jeho jméno je **Email technik**. Akcí využitou v rámci assetu je **send email**, parametry použité pro funkci bloku jsou `from` s obsahem `phantom@domain.cz` identifikující zdroj emailu, `to` se současným obsahem potřebným pro testovací účely `hons@domain.cz` tedy email technika zodpovědného za řešení playbook událostí. Dále parametr `subject` s vepsaným řetězcem `Podezřelý email detekován`, slouží pro definování předmětu emailu. Nakonec parametr `body` jehož obsahem je dříve formátovaná zpráva `Format_zpravy:formatted_data`. Tento blok slouží k informování o nebezpčené události emailem.

Zároveň s blokem **Email technik** je vykonáván blok typu **Prompt** spadající do kategorie **Analýza**. Prvním nastaveným parametrem je `Approver`, který je nastaven na účet Splunk Phantom v tomto případě tak na Kamil Hons. Parametr označuje kdo bude schvalovat následné smazání emailu. Druhým parametrem je `Required response time` nastavený na 30 minut. Obsahem parametru `message` je předformátovaná zpráva předchozího bloku. Prametr `Response Type` má za úkol nabídnout operátorovi při schvalování typ odpovědi pro vyhodnocení, v tomto případě je zvolena hodnota `Yes/No`.

Rozhodovací blok číslo devět slouží k určení, zda nebyl překročen limit daný pro odpověď operátora, stejně tak jako zda nebyla jeho odpověď **Ne**, k tomu slouží podmínka `else`. Ta je závislá na první podmínce srovnávající obsah pole `prompt_1:action_result.summary.responses.0` s řetězcem `Yes`, pokud je vyhodnocena jako pravdivá, přechází se k bloku s názvem **Smazani emailu**. V opačném případě je

vyhodnocena podmínka else jako pravdivá a playbook končí blokem **Pin Comment Case**, který přidává k události červenou kartu se zprávou **Require action** pomocí parametrů `message`, `pin type` a `pin color`. Dále blok přidává komentář k události s obsahem `Nebyla potvrzena akce smazání emailu` informující o skutečnosti nepovolení smazání emailu, zároveň je akcí **promote to case** povýšena událost na případ, spolu s nastaveným parametrem `template` na hodnotu **Suspicious Email**. Pro uzavření případu je nutné manuální věnování se povýšené události operátorem.

Blok Smazání emailu je blok typu **Action**, kategorie **Akce** využívající assetu **Microsoft Exchange On-Premise EWS** a akce **delete email**. Jeho jediným parametrem je `id`, které ozančuje jaký email je třeba smazat. Pole označující tento email je `get_email_1:action_result.parameter.id`. Výsledkem úspěšného spuštění akce je smazání emailu na emailovém serveru. Pro vyhodnocení úspěšnosti akce je implementován rozhodovací blok číslo 10. Ten testuje, zda se obsah pole `Smazani_emailu:action_result.status` rovná řetězci `success`. Pokud je první zelená podmínka vyhodnocena jako pravda, plybook pokračuje k dalšímu bloku Zprava pro prijemce. V opačném případě je spuštěna fialová podmínka **else**, ta následně spustí blok **Neuspesne smazani**, který naformátuje zprávu informující o neúspěchu akce delete email předchozího bloku, společně s obsahem proměnného pole `Smazani_emailu:action_result.status`. Tato zpráva je poté blokem **Komentar11** akcí **add comment** s parametrem `comment` přidána k události. Dále je přiložena červená karta se zprávou **Required action** a událost je povýšena na případ.

Formátovací blok **Zprava pro prijemce** vytváří zprávu, jež informuje příjemce emailu, o jeho označení jako **nebezpečný** spolu s obsahem polí `get_email_1:action_result.data.*.t_From.t_Mailbox.t_EmailAddress` zdroj emailu a `get_email_1:action_result.data.*.t_Subject` předmět emailu. Formátovaná zpráva je následně blokem **Email prijemce** spadajícím do kategorie **Akce** odeslána akcí **send email** na základě parametru `body`. Dále pak parametry `from` phantom@xxx.cz, `to` s obsahem `get_email_1:action_result.data.*.t_ToRecipients.t_Mailbox.*.t_EmailAddress` pro identifikaci příjemce originální zprávy, `subject` nastavený na `Nebezpečný email detekován a odstraněn` specifikují podrobné informace pro akci `send email`.

Předposledním blokem je **Zprava smazana**, který nastavuje událost pomocí akce `set status` na **Closed**, akcí `pin` přidává modrou kartu se zprávou **Email deleted** a akcí `add comment` komentuje událost zprávou **Zpráva byla smazána**. Poté následuje pouze blok **END**, vymezující konec playbooku.

Nastavení celého playbooku je pak definováno parametry `Operates on` nastaveno na **email**, `Category` jako **Threat Response**, `Run as` **automation** a `tags` na **malware**. Přepínač Logging je v poloze **ON**, zbylé přepínače jsou vypnuté.

2.4.2 Kontrola podezřelé IP adresy

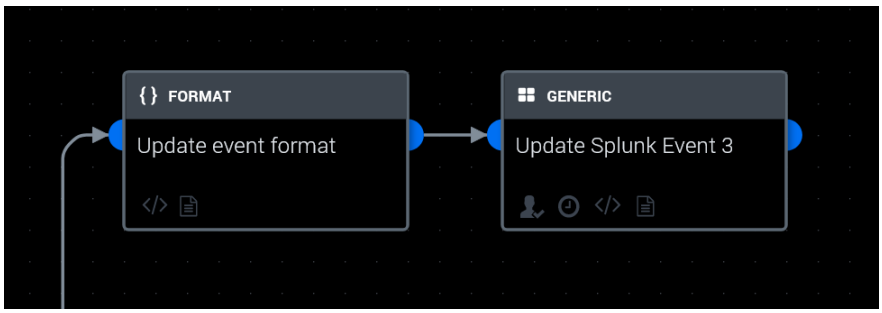
Tento playbook je praktickou implementací scénáře popsaného v části práce 2.3.2. Úkolem scénáře je provádět automatické šetření událostí, kdy je jedna z komunikujících IP adres podezřelá. Tyto události jsou zachyceny korelačním pravidlem *Threat - Threat list Actitivy*. Stěžejními částmi playbooku jsou kontrola reputace IP adresy, vyhodnocení rizika, průběžné reportování a akci sloužící k blokaci na firewallu. Náhled celého playbooku viz příloha A.2.

Nyní bude popsána funkce playbooku s názvem **Investigate Remote IP DP** po jednotlivých blocích. Tento playbook je obsahem přiloženého CD viz příloha B. Po bloku **START** následuje rozhodovací blok s číslem **1**, jehož úkolem je dle předlohy navrženého scénáře zkontrolovat prezenci dat. Toho je docíleno kontrolou splnění podmínky, že obsahem parametru `source` je řetězec *Threat - Threat List Activity - Rule*. Splněním této podmínky je zajištěno, že vstupující událost pochází z předem zamýšleného korelačního pravidla. Dále je tímto zamezeno automatickému spouštění playbooku nad událostmi, které by v budoucnu mohli být opatřeny štítkem **network**. Rovněž je pomocí operandu **AND** připojena druhá podmínka kontrolující zda je obsahem CEF pole `sourceAddress` nenulový řetězec. Tímto blokem je tak navíc zajištěno, že jsou k dispozici nejnutnější parametry umožňující hlavní kontrolu a reporting playbooku. Oproti teoretickému scénáři zde není kontrola cílové adresy, neboť je předpokládáno, že určení směru komunikace proběhlo v rámci korelačního pravidla mimo systém Splunk Phantom.

Dalším krokem scénáře je kontrola reputace IP adresy. K tomu playbooku slouží akční blok **ip reputation**, který vyžaduje jako vstupní parametr IP adresu k prošetření. Parametrem ze strany Phantomu je již zkontrolované pole `sourceAddress`. Blok následně v rámci šetření odešle adresu na stranu serveru VirusTotal k prošetření a po dobu několika desítek sekund se dotazuje na dostupnost výsledků pro jeho dotaz. Po skončení dotazování je blok ukončen buď jako **success** nebo **failure**. Podmínka číslo **4** slouží ke kontrole, zda byla akce detonace na serveru VirusTotal úspěšná, pokud ano, následuje paralelní spuštění dalších dvou větví playbooku. V případě, že je vyhodnocena jako else, přechází playbook k bloku **VT fail comment**, který slouží jako blok **hlášení** chyby ve formátu lidsky čitelné zprávy.

Jak již bylo zmíněno, úspěšná detonace předchází paralelnímu spuštění větví playbooku, první z nich, je větev obsahující bloky **Update event format** a **Update Splunk Event 3**. Při kategorizaci dle navrženého scénáře by větev spadala do kategorie **Hlášení**. Náhled této větve viz obrázek 2.19. První blok slouží k předformátování zprávy a pro svůj dynamický obsah využívá tří návratových hodnot uložených a dostupných v cestě `ip_reputation_2:action_results`. Konkrétně se pak formátovaná zpráva skládá z parametrů `summary.malicious`, `summary.suspsicious`

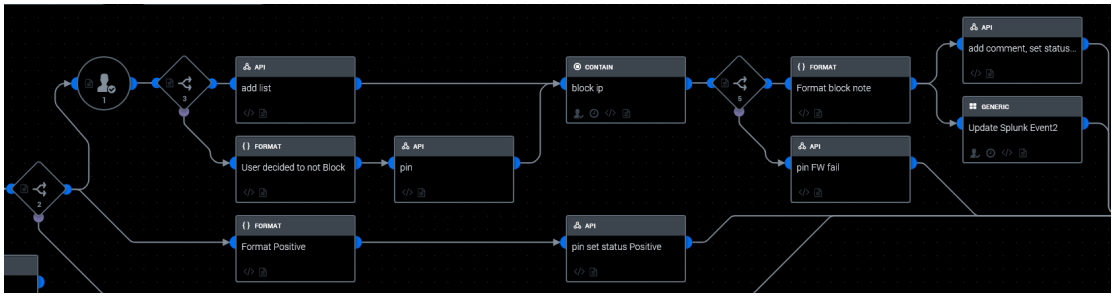
a `harmless`. Každý jeden z těchto parametrů obsahuje číselnou hodnotu dle počtu detekcí na VirusTotal daného typu. Takto předformátovaná zpráva je poté předána druhému bloku větve, který se stará o zaznamenání těchto informací v prostředí Splunk Enterprise Security. Tento způsob reportování do systémů odlišných od samotného Splunk Phantom je běžný a závisí na požadavcích specifikovaných při předběžném návrhu playbooku. Parametry tohoto bloku jsou povinný `event_ids`, pro který byl využit parametr artefaktu `event_id` jednoznačně identifikující událost v rámci Splunk Enterprise Security a nepovinný parametr `comment` jehož obsahem je výstup předchozího formátovacího bloku.



Obr. 2.19: Náhled první větve playbooku pro kontrolu IP adresy

Paralelně je spouštěna větev začínající rozhodovací podmínkou číslo **2**. Tato podmínka je stěžejním blokem pro kategorii **zkoumání** a zde dochází k rozhodnutí playbooku, zda jde o událost typu **false positive**, nebo **true positive**. Vyhodnocení podmínky pak závisí na prametrech `summary.malicious` a `summary.suspicious`, které jsou popsány výše. Pokud je alespoň jeden z těchto parametrů vyšší než specifikovaná celočíselná hodnota, dochází k vyhodnocení jakožto **true positive**. V případě nesplnění podmínky je vyhodnocena událost jako **false positive**.

Větev playbooku spuštěna po identifikaci události jako **true positive** je k nahlédnutí viz obrázek 2.20. Prvním blokem větve je potom dotaz pro operátora s čís-



Obr. 2.20: Náhled true positive větve playbooku pro kontrolu IP adresy

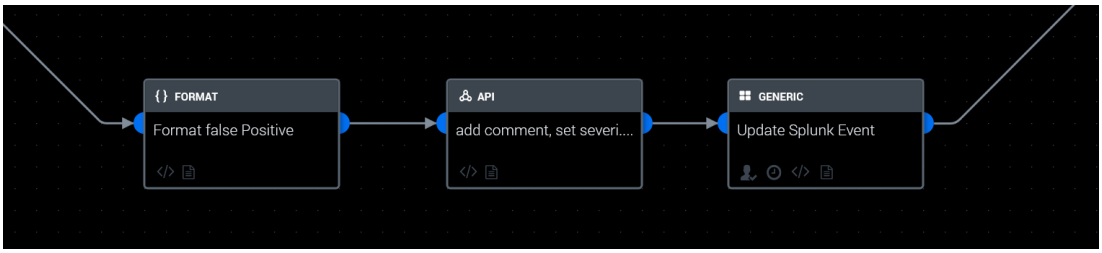
lem **1**. Tento dotaz bude vyvolán po dobu 3 minut a obsahem zprávy pro operátora bude strukturovaná zpráva s dynamickým obsahem třech parametrů `sourceAddress`, `summary.malicious` a `summary.suspicious`. Operátor pak má možnost odpovědět **Yes** nebo **No** na otázku, zda je vhodné zdrojovou adresu blokovat na firewallu. Výchozí odpovědí pro případ vypršení časového limitu je **Yes**. Z hlediska bezpečnosti je takto navržená politika výchozího blokování vhodná, ale pro nasazení playbooku v praxi je nutné takovouto politiku pečlivě zvážit, a informovat o ní cílové správce systémové infrastruktury.

Po zodpovězení dotazu operátorem následuje blok rozhodovací podmínky číslo **3**, který odpověď vyhodnocuje. Modrá podmínka je vyhodnocena jako pravda pokud operátor zvolí odpověď **Yes**, nebo v případě že je dotaz nezodpovězen, tedy **failed**. Takto je prosazena politika výchozího blokování IP adresy na firewallu. Následně je proveden blok **add list**, který slouží jako výstup playbooku a obsahuje všechny adresy určené k blokaci. Adresa k blokování je pak předána již známým parametrem `sourceAddress`. Další postup playbooku blokem **block ip** bude popsán později v textu, nejprve následuje popis sledu událostí pro případ fialové rozhodovací podmínky bloku **3**. Zde dochází k porovnání odpovědi operátora oproti řetězci **No**. Takovýto běh pro playbook znamená přechod do formátovacího bloku **user decided to not Blok**, kde je připravena správa s dynamickým obsahem již probíraných parametrů `sourceAddress`, `summary.malicious` a `summary.suspicious`. V rámci tohoto playbooku je pak zpráva přidána pouze do události v rámci Splunk Phantomu události a to pomocí bloku s názvem **pin**. Tento blok přidává graficky modifikovanou zprávu v podobě modré karty spolu s obsahem předformátované zprávy. Díky grafické odlišnosti je zajištěno **zdůraznění skutečnosti o odmítnutí blokace** operátorem.

Blok nesoucí název **block ip** spadá do kategorie **Akce** a slouží k provedení blokace za pomoci příslušného assetu na cílovém firewallu. V rámci tohoto playbooku se jedná o firewall výrobce Fortigate a díky dvěma povinným vstupnímu parametru `sourceAddress` v poli `ip` a řetězce **Block bad IPs** v poli `policy` je schopen prosazovat akci blokace při běhu playbooku. Nyní je podmínkou číslo **5** kontrolováno, zda je stav předchozího bloku blokace **úspěšný**. Pokud tomu tak není, je vytvořena graficky modifikovaná zpráva v podobě červené karty s informací o neúspěchu požadované blokace. V případě, že je však úspěch předchozí akce potvrzen, může playbook pokračovat spuštěním bloku **Format block note**, kde je naformátována zpráva o úspěchu blokace na firewallu. Jediným dynamickým parametrem je zde `sourceAddress`. Obsah této zprávy je poté **hlášen** bloky **add comment, set status** formou zprávy, grafické karty, závažnosti, upravením statusu na hodnotu **Closed** a **Update Splunk Event2** formou komentáře do příslušného externího systému.

Nyní se vrátíme k popisu dvou zbývajících bloků, pro běh playbooku, kdy je o události rozhodnuto jako **true positive** a to konkrétně **Format Positive** a **pin set status Positive**. Běh playbooku tyto dvě akce provádí paralelně a účelem je provést **hlášení** pro událost v rámci Splunk Phantom. První z dvojice bloků provádí formátování zprávy stejným způsobem jako dříve a to z parametrů `sourceAddress`, `summary.malicious` a `summary.suspicious`. Druhým blokem je poté provedeno komentování události a zároveň upravení statusu události na hodnotu **open**.

Playbook obsahuje pouze tři bloky pro případ spuštění větve **false positive**, následující po rozhodovacím bloku číslo **2**. Náhled těchto bloků viz obrázek 2.21. Prvním blokem je opět formátování zprávy stejně jako v předchozích případech



Obr. 2.21: Náhled false positive větve playbooku pro kontrolu IP adresy

a zakládá se tedy na známých parametrech `sourceAddress`, `summary.malicious`, `summary.harmless` a `summary.suspicious`. Druhý blok se stará o zapsání zprávy do události v rámci Splunk Phantom a dále o nastavení závažnosti na hodnotu **low**, statusu na **Closed** a taktéž zprávu zvýrazňuje jako kartu modré barvy pro zdůraznění závěru běhu. Poslední blok **Update Splunk event** slouží opět k **hlášení** do externího systému.

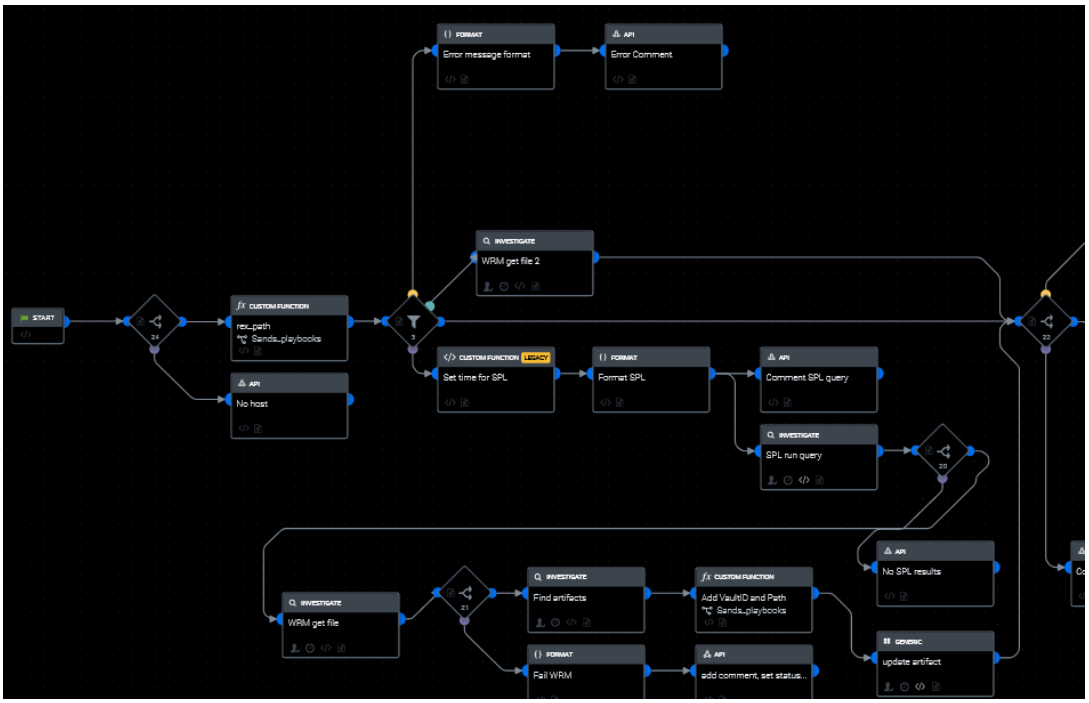
Průběh tohoto playbooku tak zajišťuje bloky ze všech kategorií scénáře a je schopen událost označit jako jednu z možností **false positive**, nebo **true postive**. Pro případ některých možných očekávaných chyb, je zároveň přítomno několik podmínek na testování těchto skutečností spolu s patřičným **hlášením** do události v rámci aplikace splunk Phantom. Playbook je v tomto stavu připraven pro řešení incidentů, kdy dochází ke komunikací dvou IP adres, z nichž jedna se nachází v lokální síti a druhá, podezřelá ve vnější síti.

2.4.3 Kontrola souboru na vzdáleném zařízení

Tento playbook je praktickou implementací scénáře v prostředí Splunk Phantom. Scénář sloužící jako teoretické východisko pro tuto implementaci je popsán v části práce 2.3.3. Zdrojem událostí, nad kterými má playbook operovat, je korelační pravidlo *Threat - Threat List Activity*, ze kterého jsou následně vybírány události, které

splňují kritéria detekce potenciálně nebezpečného souboru na vzdáleném zařízení. Tyto události jsou přeposílány do prostředí Splunk Phantom, kde jsou poté automaticky analyzovány právě popisovaným playbookem. Tento playbook se drží předem definovaných kategorií postupu pro řešení bezpečnostní události a to konkrétně **Sběr**, **Zkoumání**, **Analýza**, **Hlášení**, **Akce**. Pro účely tohoto playbooku byl brán zvláštní zřetel na umožnění **Hlášení** do systému Splunk Enterprise, kde jsou události detekovány a je tak docílena demonstrace možnosti komplexního **Hlášení** do systému jiného než Splunk Phantom. Náhled celého playbooku viz příloha A.3.

Nyní bude popsána funkce playbooku s názvem **Endpoint Malware DEV DP** po jednotlivých částech, vždy s přiloženým ilustračním obrázkem pro lepší přehlednost. Tento playbook je obsahem přiloženého CD viz příloha B. Nejprve část playbooku, která se věnuje kontrole a přípravě potřebných dat viz obrázek 2.22. Úvod playbooku značí blok **Start**, po kterém následuje rozhodovací blok s číslem



Obr. 2.22: Náhled kontrolní a přípravné části playbooku pro kontrolu souboru na vzdáleném zařízení

24, jehož úkolem je provést kontrolu pole `destinationHostName`. V případě, že je obsah pole řetězec **unknown**, nelze provést šetření, a je tak využito **API** bloku **No host**, který uzavírá událost v prostředí Phantom a provádí akce nutné pro splnění **Hlášení** s příslušnou informativní zprávou o absenci pole. Pokud není testované pole prázdné, je použit **Custom function** blok s názvem **rex_path**, který provádí hledání znaku \ uvnitř řetězce vstupního pole `fileName`. Důvodem je ověření, zda

se nejedná o cestu k souboru chybně zapsanou do pole fileName v některém z parsingů aplikací třetích stran. Výstupem této funkce je poté pole `matchCount`, které obsahuje číselnou hodnotu počtu výskytů znaku v řetězci.

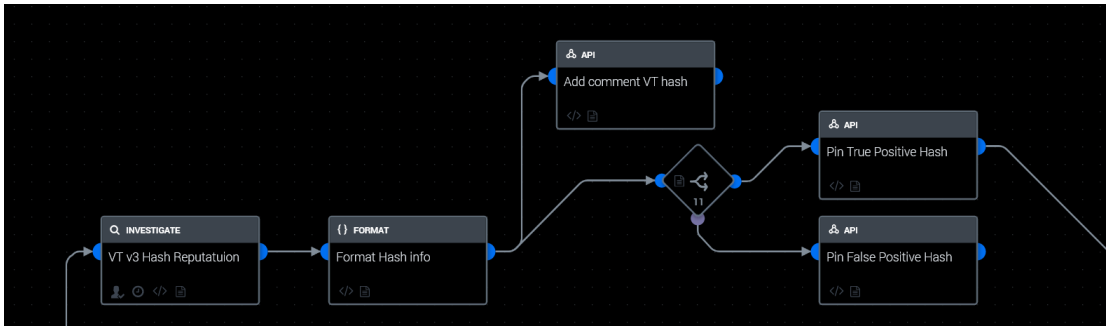
Pole `matchCount` je poté využito v podmínce číslo **3**, která slouží k ověření dostupných dat a následnému rozhodnutí, kterou větví se má playbook vydat. Podmínka s pořadovým číslem tři, se žlutým barevným označením testuje, zda není přítomen artefakt souboru a to na základě nulovosti polí `vaultId` a `fileName` v rámci prošetřovaného artefaktu. Pokud je podmínka vyhodnocena kladně, jsou na řadě bloky **Error message format** a **Error Comment**, které provádí **Hlášení** do systému Phantom. Podmínka s pořadovým číslem čtyři, odlišená zelenou barvou, testuje, zda je hodnota pole `matchCount` vyšší než 0. Pokud je tato podmínka vyhodnocena kladně, je k dispozici cesta k souboru, avšak v nesprávném poli. Toho je využito a pomocí následovného bloku **WRM get file 2** je proveden pokus o stažení souboru ze vzdáleného zařízení na základě obsahu pole `fileName` a pole `destinationHostName`. Podmínka s pořadovým číslem **1** odlišena modrou barvou a slouží k testování dostupnosti souboru a ke kontrole pomocí zkoušky, zda není obsah pole `vaultId` prázdný. Pokud je podmínka vyhodnocena kladně, je postoupeno k podmínce číslo **22**, která logicky odlišuje pozdější sekci playbooku jako začátek bloků kategorie **Zkoumání** a bude popsána později.

Fialová podmínka s pořadovým číslem **2** je v případě pravdivého vyhodnocení začátkem sledu bloků sloužících jako praktická implementace bloku **Obohacení dat**, který je součástí obecného scénáře. V této podmínce je testováno, zda není přítomen soubor ke kontrole, zároveň zda je k dispozici název souboru a zároveň zda je obsah pole `matchCount` roven nule. Pouze pokud jsou všechny tyto podmínky splněny, je postoupeno k bloku **Custom function** s názvem **Set time for SPL**, kde je pole `startTime` posunuto o 30000 jednotek linuxového času dozadu a pole `endTime` o 30000 do předu. Tento krok je stěžejní neboť bude následovat prohledávání indexu na systému Splunk enterprise, a to musí být provedeno v rozšířeném časovém okně, zajištěném právě touto funkcí. Poté může být blokem **Format SPL** sestaven řetězec pro vyhledávání na vzdáleném indexu. Parametry pro vyhledávání jsou předem definovaný, pravidelně ukládaný SPL dotaz, dále pole `filename`, `destinationHostName` a dvojice posunutých časů předchozím blokem. Takto připravený dotaz je vždy blokem **Comment SPL query** zapsán do události. Blok **SPL run query** je **akčním** blokem, který odesílá naformátovaný dotaz na příslušný splunk server a poté se dotazuje na výsledek v pravidelných intervalech. Pro ověření správnosti slouží podmínka číslo **20** jejíž modrá podmínka kontroluje, zda není status předchozí akce **failed**, a zároveň zda nově vytvořené pole `Path` není prázdné. Pokud je podmínka splněna, lze přejít k dalšímu bloku a provést pokus o stažení podezřelého souboru ze vzdáleného zařízení. Druhá podmínka označena fialově je vyhodnocena jako pravda,

pokud je počet výsledků SPL dotazu 0. Poté je blokem **No SPL results** provedeno **Hlášení**.

Blok **WEM get file** se pokusí stáhnout soubor dle pole `destinationHostName` pro určení vzdálené stanice, ze které je třeba soubor stáhnout a pole `Path`, které určí umístění souboru. K vyhodnocení úspěšnosti akce slouží blok s číslem **21**, který modře označenou podmínkou zjišťuje, zda byla akce stažení souboru úspěšná. Pokud je podmínka vyhodnocena jako nepravda, je provedeno **Hlášení** o nezdaření operace a událost je nastavena jako **Open**. Prvním blokem po úspěšném získání souboru je **Find artifacts**, ten za pomoci Phantom REST assetu vyhledá artefakty dle pole `file_intel` a `container:id`. Nyní následuje **Custom function** blok **Add VaultID and Path**, úkolem bloku je provést zapsání zjištěných hodnot `Path` a nového `VaultID` vytvořeného stažením souboru, do struktury *JavaScript Object Notation* (JSON) artefaktu. Tato nová struktura je poté blokem **update artifact** zapsána způsobem, že vstupem je nový JSON a výstupem je současný kontejner. Tímto je ukončena fáze playbooku pro **Sběr** a za následující podmínkou číslo **22**, již následují bloky z dalších kategorií.

Nejprve popis větve založené na žluté podmínce bloku číslo **22**, jejíž náhled viz obrázek 2.23. Playbook se ke spuštění této větve dostane pouze tehdy, není-li

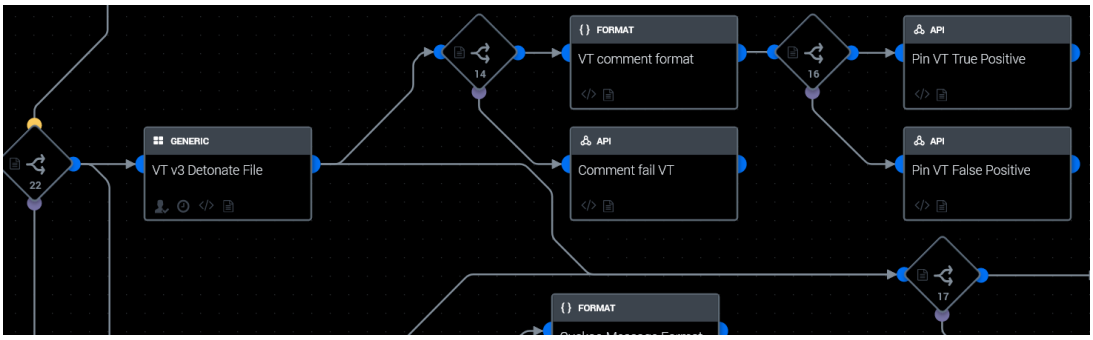


Obr. 2.23: Náhled kontroly hashe playbooku pro kontrolu souboru na vzdáleném zařízení

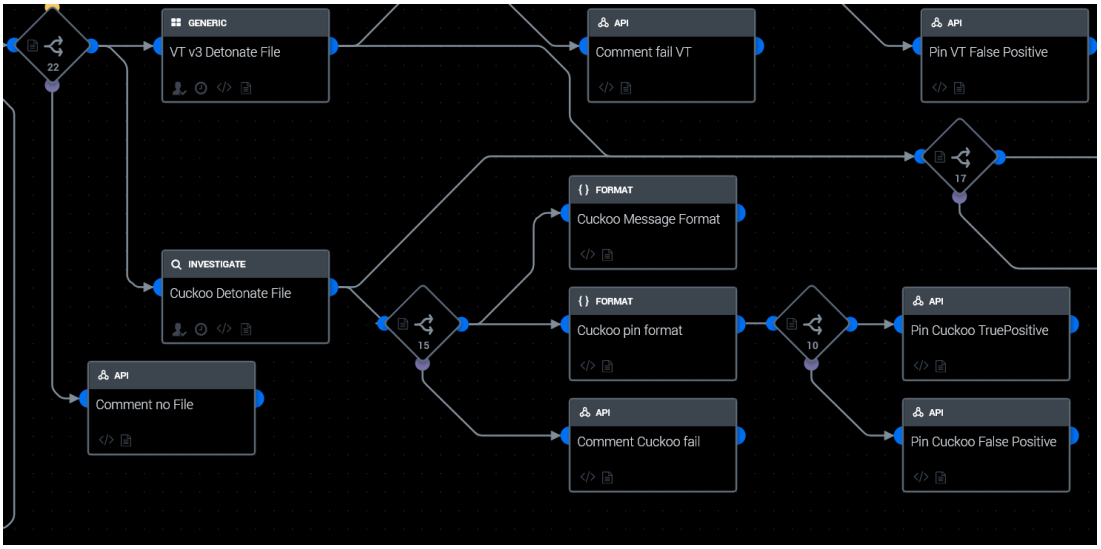
k dispozici žádný soubor ke zkoumání a zároveň je-li k dispozici pole `fileHash`, nad kterým je provedena následná detonace blokem **VT v3 Hash reputation**. Poté co je bloky **Format Hash info** a **Add comment VT hash** provedeno **Hlášení**, rozhodovací blok s čílsem **11** provede vyhodnocení **Flase**, nebo **true positive**. Pro případ modré podmínky je třeba, aby jedno z polí `summary.malicious`, nebo `summary.suspicious` obsahovalo hodnotu vyšší než hraniční limit 3. Následně by bylo provedeno **Hlášení** a přesun k bloků kategorie **Akce**, které budou popsány níže, neboť jsou společné i pro větev analyzující soubor. Pro pokrytí druhého případu

výsledku rozhodovacího bloku s číslem **11**, označeného fialovou barvou, je zde další blok **Hlášení**, který však uzavírá událost jakožto **false positive**.

Rozhodovací podmínka číslo **22** je vyhodnocena jako pravda, pokud existuje soubor k analýze, tato podmínka je barevně označena modře a následují větve pro kontrolu souboru pomocí serveru VirusTotal viz obrázek 2.24 a Cuckoo sandbox viz obrázek 2.25. Obě tyto větve jsou již detailně popsány v části práce 2.4.1, proto



Obr. 2.24: Náhled detonace VirusTotal playbooku pro kontrolu souboru na vzdáleném zařízení

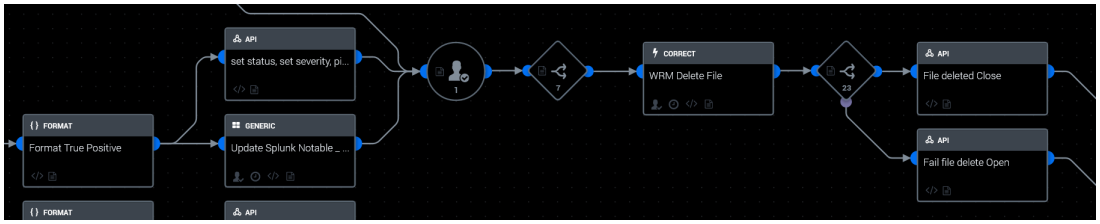


Obr. 2.25: Náhled kontroly Cuckoo sandbox playbooku pro kontrolu souboru na vzdáleném zařízení

v rámci tohoto playbooku bude sepsáno pouze krátké **funkční** shrnutí obou větví, nezbytné pro kompletní obraz funkce playbooku. Bloky s názvem **VT v3 Detonate File** a **Cuckoo Detonate File** slouží jako výchozí bloky pro **Zkoumání**. Dále je podmínkami **14** a **15** otestován úspěch obou detonací s příslušným **Hlášením** pro

možné výsledky succes a failed. rozhodovací bloky **16** a **10** poté vyhodnocují zda se jedná o **false positive**, nebo **true positive** soubor za účelem dalšího **Hlášení** pro událost v rámci aplikace Splunk Phantom. Rozhodovací podmínka číslo **17** poté provádí **Analýzu** výsledků obou zdrojů, za účelem celkového rozhodnutí o **False** nebo **true positive** souboru.

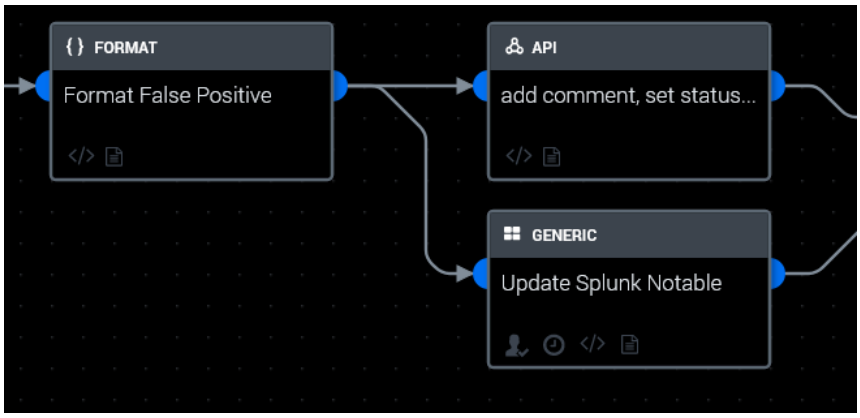
Vyhodnocení modré podmínky bloku **17** závisí na přesažení jednoho z limitů počtu detekcí serveru VirusTotal `summary.malicious`, `summary.suspicious`, nebo výší skóre Cuckoo sandbox parametrem `report.info.score`. Poté je blokem **Format True Positive** připraven komentář pro blok **UpdateSplunk Notable_infected** a **API** blok sloužící pro hlášení do Phantomu. Dále je nastavena závažnost události v obou systémech na **high** a následuje blok **Dotazu uživatele ke smazání souboru**. Pokud uživatel odpoví **Yes**, je blokem **WRM Delete File** proveden pokus o smazání infikovaného souboru. Rozhodovací podmínka **23** pak zjišťuje úspěch této akce pomocí výstupního pole předchozí funkce `status`, aby mohla událost uzavřít jedním z bloků **File deleted Close**, nebo **Fail file delete Open**. Náhled části playbooku s bloky pro **true positive** soubor viz obrázek 2.26



Obr. 2.26: Náhled postupu pro true positive soubor playbooku pro kontrolu souboru na vzdáleném zařízení

Poslední možné spuštění plabooku je závislé na nesplnění modré podmínky bloku **17**, kdy je vždy využita fialová podmínka. Toto indikuje rozhodnutí o **false positive** události. Sled bloků pro takovouto událost viz obrázek 2.27. Zde je blokem **Format False Positive** připraven řetězec k **Hlášení** pro bloky **API add comment, set status** a **Update Splunk Notable**. Tyto bloky také uzavírají událost spolu s komentářem v obou systémech.

Tímto jsou popsány všechny možné varianty průběhu playbooku pro kontrolu souboru na vzdáleném zařízení. V rámci implementace bylo využito všech kategorií dle navrženého scénáře. Jednotlivými průběhy je docíleno začlenění dvou možných pokusů o získání souboru dle odlišných parametrů, dále detonace na dvou serverech provádějící antivirovou kontrolu a pokus o obohacení události o další informace nutné k provedení šetření. Playbook je doplněn o logiku nutnou k provedení **Zkoumání** a **Analýzu**. Součástí je také průběžné **hlášení** do systému Splunk Enterprise



Obr. 2.27: Náhled postupu pro false positive soubor playbooku pro kontrolu souboru na vzdáleném zařízení

a Splunk Phantom. Cílem tohoto playbooku je správné označení nebezpečnosti souboru, detekovaného v rámci zdrojového korelačního pravidla a následně pokus o provedení **Akce** eliminující, nebo alespoň napomáhající eliminovat hrozbu.

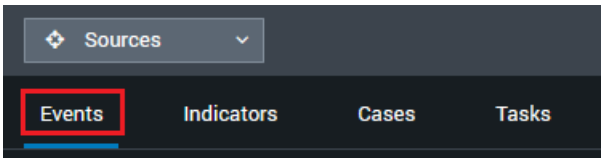
2.5 Testování

Pro použití playbooků nad daty je využito prostředí Splunk Phantom *Events*. Navigace do toho prostředí skrze tlačítko *Sources* viz obrázek 2.28 a dále záložku *Events* viz obrázek 2.29. Zde je k dispozici rychlý přehled veškerých událostí, které byly

Obr. 2.28: Navigace k přehledu událostí

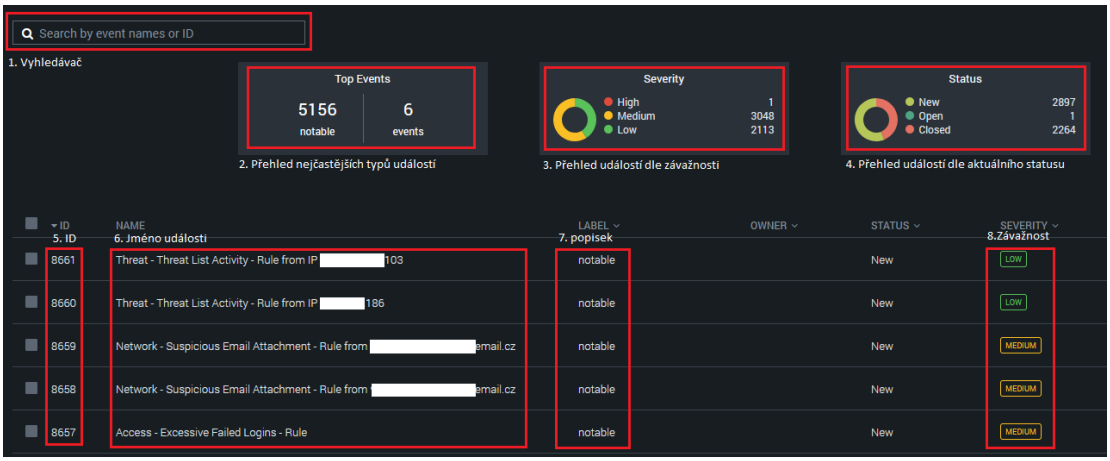
zaznamenány různými detekčními aplikacemi a následně, spolu s příslušnými informacemi k jednotlivým incidentům, předány do Splunk Phantom. Náhled tohoto přehledu viz obrázek 2.30 spolu s vyznačenými důležitými poli tohoto přehledu. Nyní následuje popis označených polí:

- 1.vyhledávač - toto pole slouží k rychlému vyhledání událostí na základě vyhledávaného řetězce.



Obr. 2.29: Záložka events pro zobrazení událostí

- 2. Přehled nejčastějších typů událostí - zobrazuje součet nejfrekventovanějších typů událostí.
- 3. Přehled událostí dle závažnosti - informace o závažnosti je v průběhu nakládání s událostí měněn na základě vyvolaných akcí.
- 4. Přehled událostí dle aktuálního statusu - tato informace se rovněž mění na základě prováděných operací.
- 5. ID - unikátní identifikátor události.
- 6. Jméno událost - poskytuje rychlý náhled na obsah a důvod vytvoření události.
- 7. Popisek - je dynamicky upravován v průběhu nakládání s událostí. Prvotní tvar získává při vzniku.
- 8. Závažnost - toto pole je dynamicky měněno.

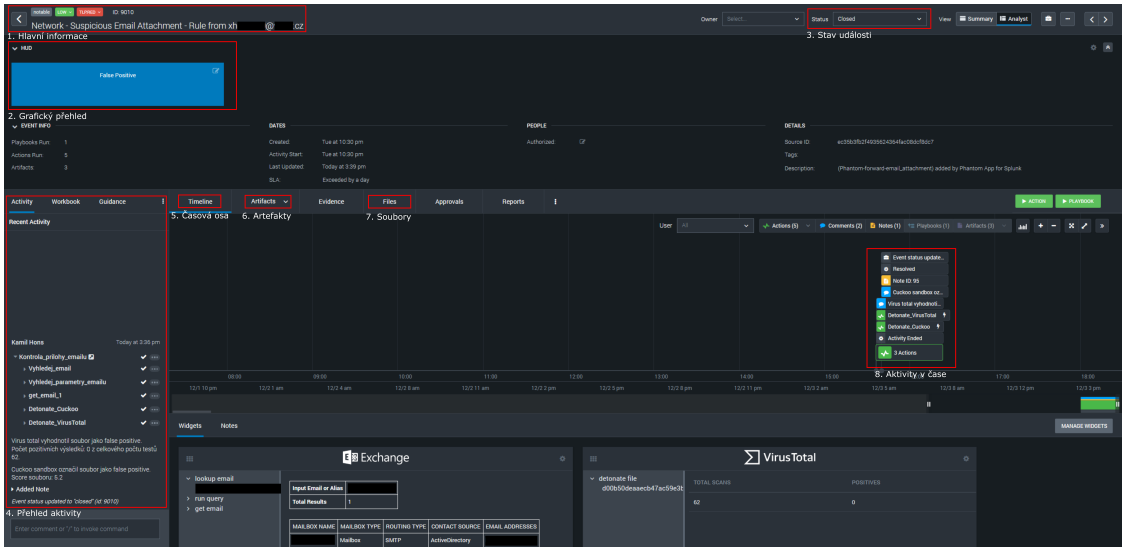


Obr. 2.30: Náhled přehledu událostí

Přehled události

Po kliknutí na konkrétní událost se dostaneme do přehledu události. K popisu hlavních částí tohoto okna byla využita jedna z testovacích událostí pro playbook popsaného v části práce 2.4.1. Vybraná událost má v této části pouze deskriptivní

charakter funkcí přehledu, náhled viz obrázek 2.31, jehož součástí je také číslované



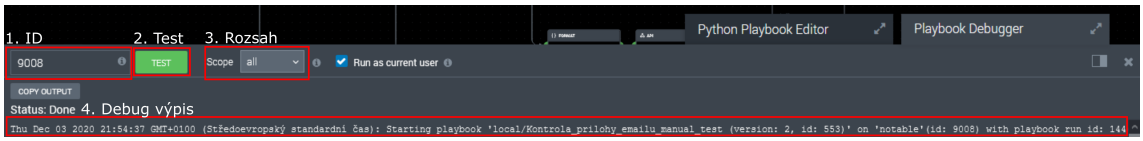
Obr. 2.31: Náhled přehledu vybrané události

zvýraznění součástí popsaných v následujícím seznamu:

- 1. hlavní informace - tato část zobrazuje štítky událost spolu se jménem události.
- 2. Grafický přehled - HUD zobrazuje nejdůležitější informace které mají být ihned viditelné, jako například právě zobrazenou kartu.
- 3. Stav události - zde je možné použít Open, Closed a New.
- 4. Přehled aktivity - po jednotlivých bodech je zde zobrazena každá akce v rámci daných playbooků, nebo samostatně.
- 5. Časová osa - v této záložce je zobrazen přehled aktivity v čase.
- 6. Artefakty - záložka slouží k zobrazení dostupných artefaktů a jejich podrobností.
- 7. Soubory - Je-li součástí události nějaký soubor, je k nalezení v této záložce s připojeným VaultId.
- 8. Aktivity v čase - aktivita je zobrazena v blocích, zejména pak jsou zde k vidění komentáře.

Debugger

Pro Manuální testování bude použito prostředí **Playbook Debugger** viz obrázek 2.32. Tento nástroj je dostupný z kteréhokoli playbooku ve spodním pravém rohu. Popis obrázku zajišťuje následující číslovaný seznam:



Obr. 2.32: Náhled debuggeru z prostředí playbooku

- 1. ID - Do tohoto pole je zadáno ID události, která bude použita k testování playbooku, do jejího přehledu budou poté zapsány veškeré výsledky.
- 2. Test - toto tlačítko spouští debug simulaci. Po spuštění je jím možné simulaci předčasně zastavit.
- 3. Rozsah - jednotlivé nastavení ovlivňují jaké artefakty budou analyzovány. Přepínač **all** slouží k zahrnutí veškerých artefaktů.
- 4. Debug výpis - zde jsou zobrazovány výpisy spuštěného debugování. Obsah výpisů může být přikládán při manuálním testování.

2.5.1 Manuální testování přílohy emailu

V následující části práce bude popsáno manuální testování a dosažené výsledky playbooku popsaného v části 2.4.1. Toto testování nebude provedeno pro zbývající playbooky implementované v rámci této práce, neboť účel zdokumentování postupu takovéhoto testování je přenositelný a lze jej analogicky provést pro jakýkoli playbook. Hlavním účelem manuálního testování je ověřit, že budou naplněny veškeré varianty playbooku. Toho bude docíleno manipulací s daty události tak, aby výsledkem spuštění playbooku byl právě požadovaný výsledek. Jednotlivé výsledky se budou hodnotit dle splnění kritéria spuštění definovaného bloku, struktura playbooku a jeho bloků viz obrázek A.1. Testovacím emailem je sada emailů odeslaných z domény VUT. Příjemcem emailu je emailová adresa s firemní doménou. Tento mail je dále modulem Threat Inteligence aplikace Enterprise Security programu Splunk vyhodnocen jako email s potenciálně infikovanou přílohou a je vytvořena událost v prostředí Splunk Phantom. Dále primárním účelem testování není kontrola vnitřních funkcí assetů, bude-li však pozorován významný problém ve funkci assetu, který je potřebný pro fungování playbooku, bude tento problém zdokumentován rovněž.

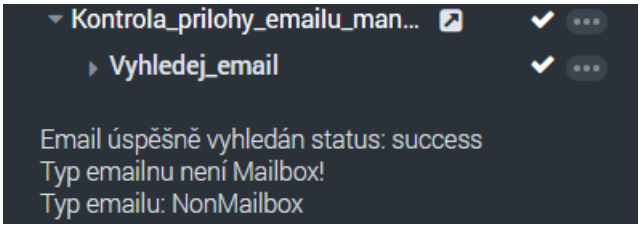
Test Komentar2

Úspěšnost tohoto testu nastane, pokud je výsledkem přidání komentáře, zapisovaného blokem **Komentar2**. Tedy situace, kdy emailová schránka není typu Mailbox. Způsob docílení je vytvořením bloku typu **Custom Function** uvnitř duplikovaného originálního scénáře. Obsah této funkce viz výpis 2.2.

Výpis 2.2: Úprava pole Mailbox

```
ZmenaMailboxu__NewType = "NonMailbox"
    phantom.debug("Mail␣box␣type␣to␣check:")
    phantom.debug(ZmenaMailboxu__NewType)
```

Tím je vytvořeno pole `NewType`, které je poté použito v testovací podmínce namísto pole `Vyhledej_email:action_result.data.*.t_Mailbox.t_MailboxType`. Výsledek testu viz obrázek 2.33. Zde je prokázáno, že komentář byl zapsán a že obsahem je uměle vytvořené pole.



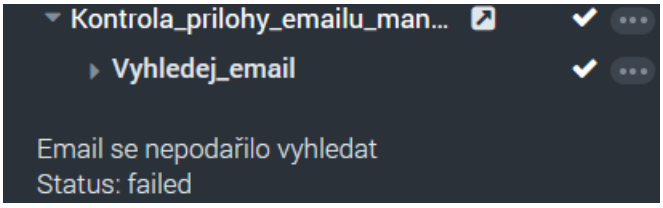
Obr. 2.33: Úspěšný výsledek testu bloku Komentar2

Test Komentar1

Vytvoření nového pole s obsahem `failed` pomocí **Custom Function** bloku viz výpis 2.3. Tento test tedy kontroluje pouze funkci testovací podmínky, pro otestování správného komentáře je však dostatečný, výsledek viz obrázek 2.34

Výpis 2.3: Úprava pole Status

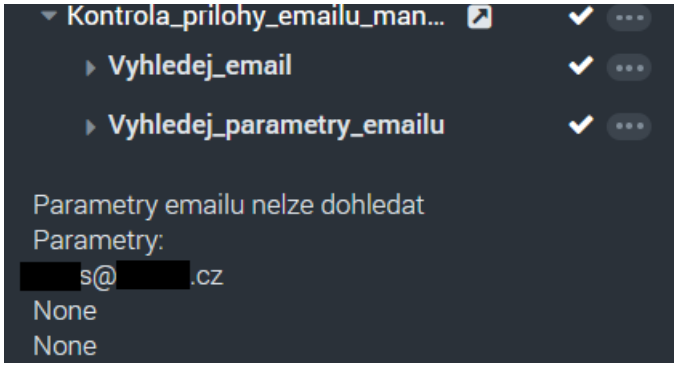
```
ZmenaPromene__NewStatus = "failed"
    phantom.debug("Status␣to␣check:")
    phantom.debug(ZmenaPromene__NewStatus)
```



Obr. 2.34: Úspěšný výsledek testu bloku Komentar1

Test Komentar3

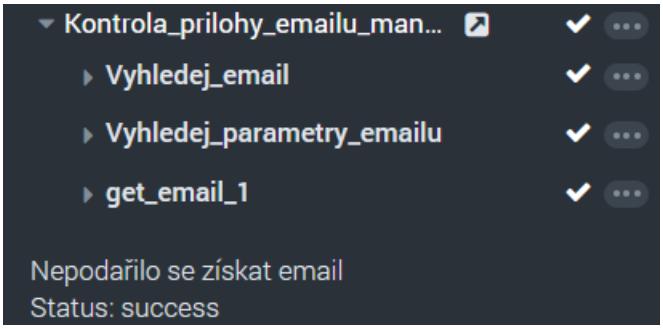
Opět se jedná o testování rozhodovacího bloku. K tomu nám postačí funkce definovaná výše viz výpis 2.3. Výsledkem je přidaný komentář s obsahem polí použitých jako vstup pro blok **Vyhledej_parametry_emailu** viz obrázek 2.35. Operátor je takto schopen ihned rozlišit, které pole bylo důvodem neúspěchu. V tomto případě byl stav `failed` navozen uměle.



Obr. 2.35: Úspěšný výsledek testu bloku Komentar3

Test Komentar4

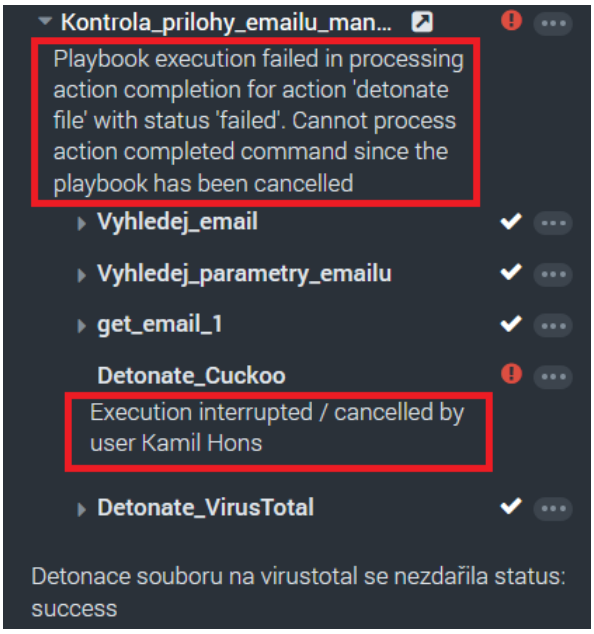
Komentář číslo čtyři závisí na výsledku rozhodovacího bloku tři, kterému byl podsunut uměle vytvořený status neúspěchu stejně jako v předchozích testech viz výpis 2.3. Výsledkem by měl být komentář o neúspěchu akce s přiloženým polem `status`. Obsahem pole má být `success` vzhledem k ponechání originálního zdroje pole `get_email_1:action_result.status` blokem **get email**. Výsledek testu viz obrázek 2.36.



Obr. 2.36: Úspěšný výsledek testu bloku Komentar4

Test Komentar5

Cíleným výsledkem testu, je podsunout rozhodovacímu bloku číslo šest vytvořené pole s obsahem `failed`. Výsledkem tedy bude komentář o neúspěchu akce a obsah proměnné `Detonate_VirusTotal:action_result.status` tedy `success`. Výsledek testu viz obrázek 2.37, na kterém jsou červenými rámečky vyznačena hlášení o předčasném zrušení testu. Test byl předčasně zrušen z důvodu dosažení cíle testu komentářem o neúspěchu detonace assetem VirusTotal spolu s výpisem hodnoty **success**.



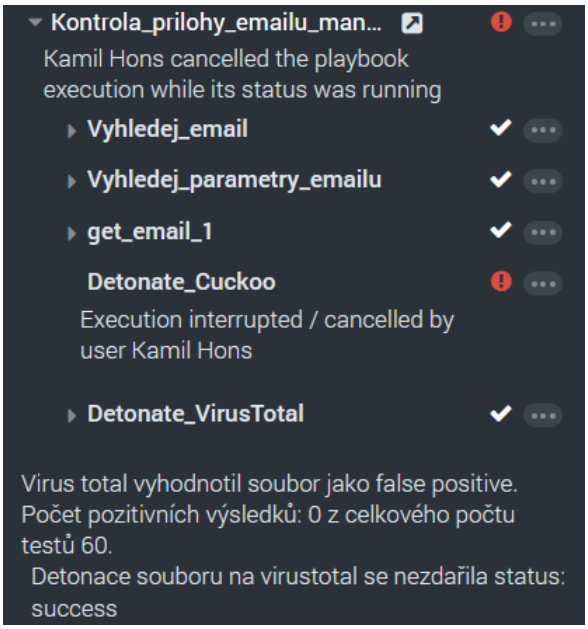
Obr. 2.37: Úspěšný výsledek testu bloku Komentar5

Test Komentar6

Dosažení výpisu tohoto bloku lze docílit spuštěním nad testovacím emailem, vzhledem k tomu, že přílohou není nebezpečný soubor. Spuštění je tedy provedeno bez úprav scénáře. Výsledek testu viz obrázek 2.38. Součástí výpisu aktivity události, jsou opět hlášení o předčasném ukončení debugování.

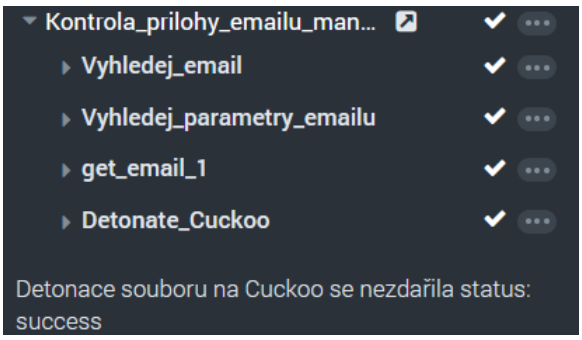
Test Komentar7

Pro dosažení výpisu tohoto bloku je zapotřebí opět rozhodovacímu bloku číslo čtyři podsunout řetězec k vyhodnocení s obsahem **failed**. Toho je docíleno opětovným zapojením bloku typu **Custom function**, kde je tento řetězec vytvořen. Dále pro



Obr. 2.38: Úspěšný výsledek testu bloku Komentar6

zkrácení výpisu byla odpojena větev virustotal, vzhledem k paralelnímu spouštění není pro účely tohoto testování zapotřebí. Předpokladem úspěšného testu je přidání komentáře o neúspěchu detonace v prostředí Cuckoo sandbox spolu s výpisem proměné `Detonate_Cuckoo:action_result.status`, jejímž obsahem má být řetězec **success**. Výsledek testování viz obrázek 2.39.

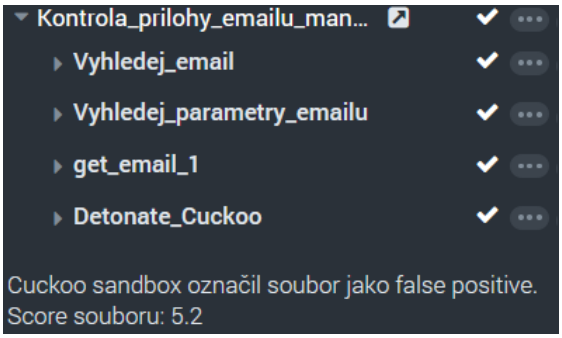


Obr. 2.39: Úspěšný výsledek testu bloku Komentar7

Test Komentar8

V tomto testu zůstává vynechána větev detonace souboru na VirusTotal vzhledem k nepotřebnosti jejího spuštění. Dále není třeba využít žádného upravení playbooku oproti originální verzi popsané v části práce 2.4.1. Úspěch testu je založen na vypsání

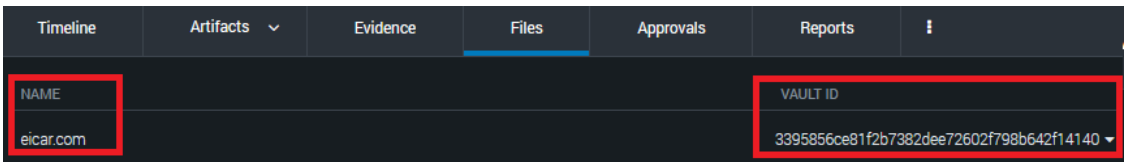
komentáře informujícím o false positive detekci společně s dosaženým skóre analýzy. Výsledek testu viz obrázek 2.40.



Obr. 2.40: Úspěšný výsledek testu bloku Komentar8

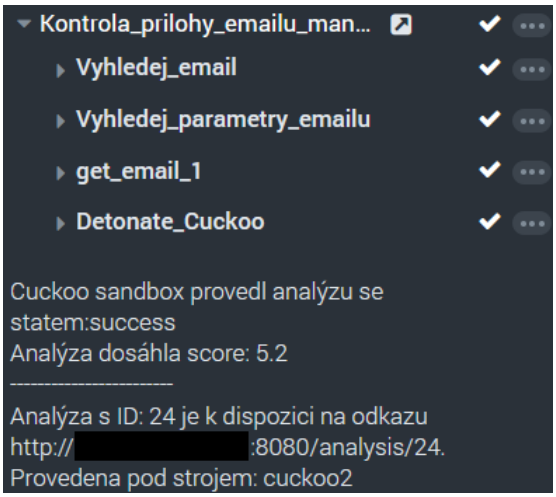
Test Komentar9

První test tohoto bloku spočívá v podsunutí infikovaného souboru do události. Toho je docíleno nahráním souboru přímo do události viz obrázek 2.41. Dále je třeba odpojit blok **get email**, který by artefakt nahradil původní přílohou emailu. Vstupní parametry pro blok **Detonate Cuckoo** byly manuálně nahrazeny tak, aby vzali podsunutý soubor eicar.com. Použitá pole jsou označeny červenými rámečky viz obrázek 2.41. Výsledkem analýzy je score 1.2, které neoznačí soubor jako nebezpečný. Proto je tento test neúspěšný, avšak ukazuje možnost podsunutí souboru playbooku.



Obr. 2.41: Podsunutí souboru eicar.com do testovací události

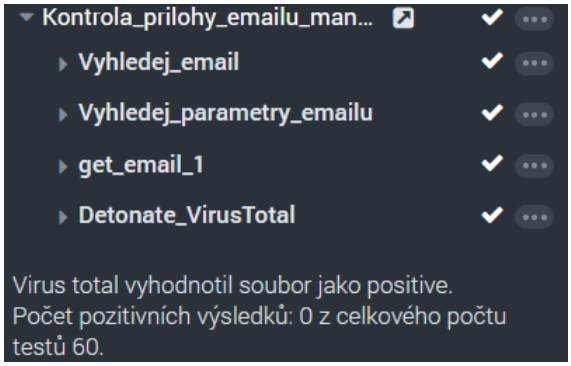
Druhý test je tedy založen na pouhé změně parametrů pro vyhodnocení výsledku jako pozitivní. Toho je dosaženo použitím originální přílohy, která dosahuje score vyššího než 4. V rozhodovacím bloku číslo pět je tedy změněna hranice způsobem označujícím soubor jako pozitivní, je-li score vyšší než 4. Výsledek testu viz obrázek 2.42.



Obr. 2.42: Úspěšný výsledek testu bloku Komentar9

Test Komentar10

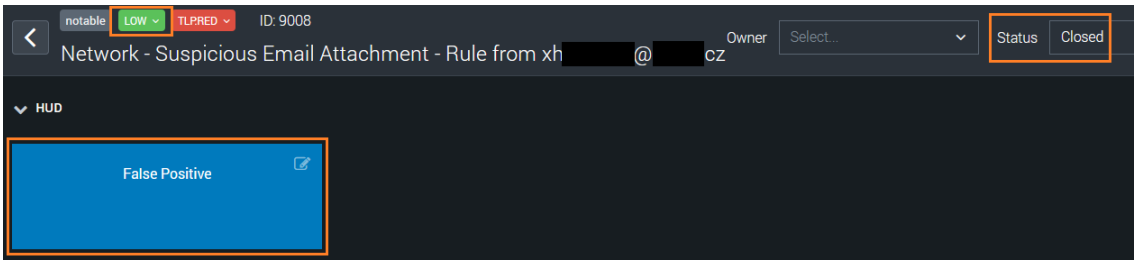
V tomto testu bude upravena hranice detekce VirusTotal. Pokud je počet pozitivních označení souboru menší než 5, je formátována zpráva blokem **VT positive**. Úspěšný test nastane v případě, že bude zapsán komentář vypovídající o vyhodnocení souboru jako positive, i přes číslo detekcí rovné 0. Výsledek testu viz obrázek 2.43



Obr. 2.43: Úspěšný výsledek testu bloku Komentar10

Test API Pin Close Severity

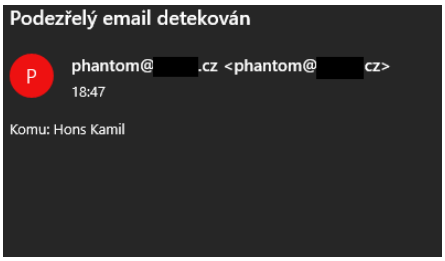
Testovací email je pro účely tohoto testu dostatečný v podobě, ve které se nachází. Předpokládaný výsledek testu připne kartu modré barvy a příslušného komentáře, dále nastaví závažnost události na **low** a status na **Closed**. Výsledek testu viz obrázek 2.44, součástí jsou oranžové rámečky označující důležité pole k ověření úspěšnosti testu.



Obr. 2.44: Úspěšný výsledek testu bloku Pin Close Severity

Test Email technik

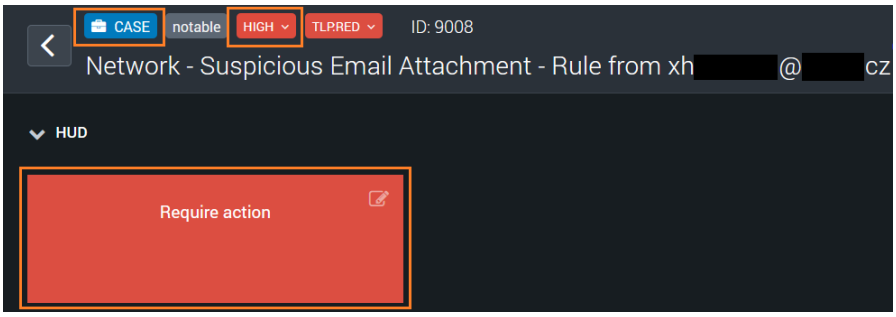
Pro dosažení odeslání zprávy technikovi je třeba snížit opět hranici vyhodnocení souboru jako pozitivního, a to jak v podmínce číslo pět tak číslo osm. Výsledný email viz obrázek 2.45 z nějž je patrné, že součástí emailu není tělo zprávy. Jedná se o chybu použité verze assetu SMTP. V současnosti je pro potřeby testování v testovacím prostředí předmět emailu považován za dostatečný, do budoucna je předpoklad odstranění této chyby v nové verzi assetu, případně využití alternativního assetu k odeslání emailu. Na základě výše uvedeného je test považován za úspěšný pro potřeby experimentálního prostředí.



Obr. 2.45: Náhled obdrženého emailu technika

Test Pin Comment Case

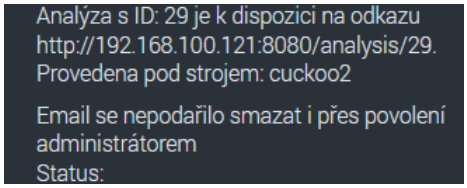
Tento test vychází z předchozího testu bloku **Email technik**, kde na výzvu ke smazání souboru bylo odpovězeno **No**. Tím je umožněno spuštění bloku **Pin Comment Case**, který připojí červenou kartu se zprávou **Require action**, nastaví závažnost události na **high** a povýší událost na případ. Výsledek testu viz obrázek. 2.46.



Obr. 2.46: Úspěšný výsledek testu bloku Pin Comment Case

Test Komentar14

V tomto testu je rozhodovacímu bloku nastaven vstup podmínky pomocí bloku **Custom function** na hodnotu **failed**. Dále byl vynechán blok **Smazani emailu** a s ním také dynamické pole formátovacího bloku Úspěšný výsledek testu bloku **Neuspesne smazani**. Předpoklad úspěšného testu je přiložení komentáře o neúspěchu akce smazání emailu. Výsledek testu viz obrázek 2.47.



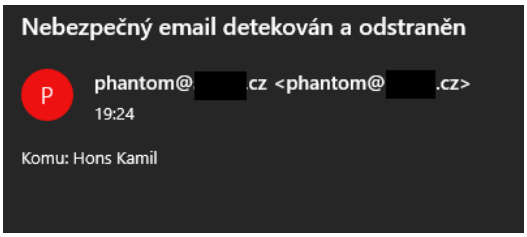
Obr. 2.47: Úspěšný výsledek testu bloku Komentar14

Test Email prijemce a Zprava smazana

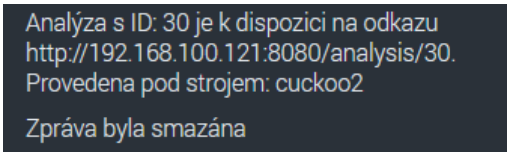
Posledním manuálním testem je ověření, zda je odeslán email o smazání nebezpečného emailu příjemci a zda je přidán komentář k události o této skutečnosti. Výsledek viz obrázek 2.48 a 2.49.

2.5.2 Automatické testování přílohy emailu

Tato část práce se věnuje automatickému testovaní playbooku popsaného v části práce 2.4.1. Tento playbook byl vykopírován do nového plybooku s názvem **Kontrola_prilohy_emailu_automat** z důvodu úprav, před spuštěním automatického testování. Zmíněné úpravy byly provedeny za účelem zamezení odstraňování emailů a odesílání zpráv příjemců, aby bylo předejito odstranění potenciálně důležitých



Obr. 2.48: Náhled obdrženého emailu příjemce



Obr. 2.49: Úspěšný výsledek testu bloku Zprava smazana

emailů. Úpravami se rozumí odpojení bloku **Smazani emailu** a s tím také zbytku následujících bloků, které na něm závisí. Playbook je nastaven na spouštění nad událostmi s položkou `label` nastavenou na hodnotu `email`. Pro nastavení spuštění byl **status** playbooku změněn na **Active**.

Výsledky automatického testování

Playbook byl testován po dobu čtyř dnů a výsledky tohoto testování jsou zobrazeny v tabulce 2.7. Testování probíhalo ode 1.12.2020 do 11.12.2020 vzhledem k tomu, že data z prvních šesti dní byla nekonzistentní a neměla dobrou výpovědní hodnotu. Budou prezentovány pouze výsledky ze dnů 8.12. až 11.12., které pokryly potřeby testování dostatečně a poukázaly na možné situace, ke kterým při automatickém spouštění playbooku dochází. Nejprve bych rád popsal výsledky analýzy ze dne 8.12., tento den bylo provedeno 10 spuštění playbooku, všech 10 událostí bylo vyhodnoceno jako false positive, všechny byly uzavřeny a správně ozančeny. Druhý testovacím den 9.12. bylo provedeno pouze 9 úspěšných spuštění playbooku, které skončily uzavřením události. Prvním otevřeným případem byla událost, která neobsahovala přílohu a je ukazatelem nutnosti implementace ošetření pro tento případ. Druhá událost selhala při analýze pomocí VirusTotal a Cuckoo sandbox, playbook se tak zachoval správně, avšak z důvodu problému na síti nebyl úspěšně ukončen. Dne 10.12. bylo úspěšně provedeno 8 spuštění playbooku, které skončily označením události jako false positive. První neúspěch analýzy tohoto dne byl způsoben chybou na síti při získávání emailu ze serveru a spuštění bylo pouze nedokončené. Druhý neúspěch byl z důvodu chybějící analýzy ze serveru virus total pravděpo-

dobně z důvodu nedostupnosti a měl by vést k implementaci pojistky nepotřebnosti obou zdrojů analýzy. Dne 11.12 pak bylo zaznamenáno 9 úspěšných provedení playbooku z toho 7 s označením false positive a 2 true positive. Při zkoumání označeného incidentu bylo v prvním případu patrné, že se pravděpodobně jedná o phishingový útok. V druhém případě byl odhalen virus trojský kůň v příloze emailu.

Tab. 2.7: Tabulka výsledků automatické analýzy playbooku pro kontrolu emailu

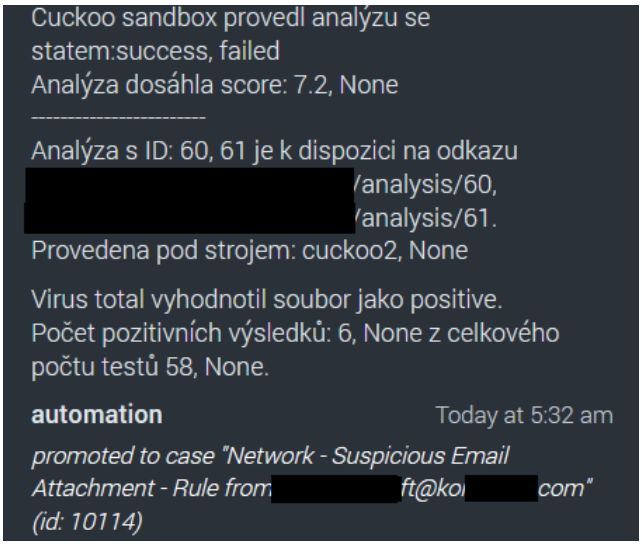
Datum	Počet událostí	Close	Open	Úspěšné spuštění	Neúspěšné spuštění	False positive	True positive
11.12.2020	9	7	2	9	0	7	2
10.12.2020	10	8	2	8	0	8	0
9.12.2020	11	9	2	9	0	9	0
8.12.2020	10	10	0	10	0	10	0

Výsledky prvního pozitivního případu

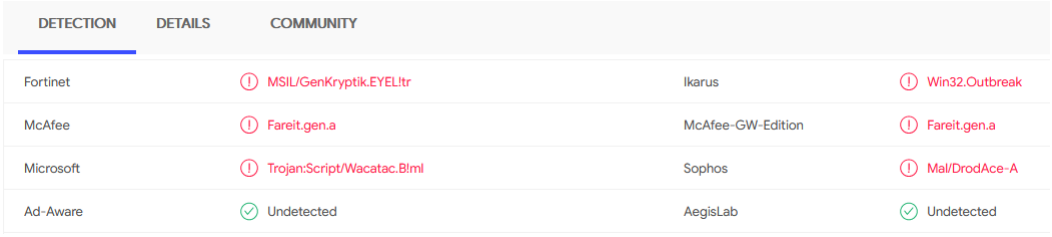
Výpis po spuštění playbooku viz obrázek 2.50. Co se druhé analýzy souboru týče, jednalo se o obrázek s příponou .png a byla neúspěšná, jak na serveru Cuckoo sandbox, tak VirusTotal. Důležité výsledky z analýzy prvního souboru však poukazovaly na nebezpečí tohoto souboru. Cuckoo sandbox extrahoval artefakt v podobě skriptu, dále analýza chování zjistila vytvoření nestandardního podprocesu, do kterého potom byl vložen skript, proběhla také kontrola zda není soubor spouštěn v prostředí debuggeru. Výsledek analýzy score 7.2 tak potvrzuje domněnku o true positiv události. Virus total analýza souboru s hashem **799385cf0dde9df399b1f3df68215c9b7fd375d57465e41843b466e91cd4da90** poukazuje na 6 true positive detekcí, její náhled viz obrázek 2.51

Výsledky druhého pozitivního případu

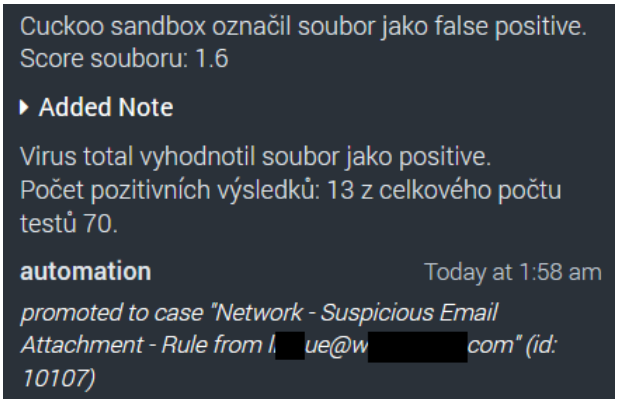
Výpis této analýzy viz obrázek 2.52, z něj je patrné, že cuckoo sandbox škodlivý soubor neoznačil vysokým score, je to z důvodu nespuštění .exe souboru. Ten provedl 4 krát kontrolu, zda není spuštěn ve virtuálním prostředí a následně jej detekoval. Tento výsledek bude sloužit k upravení Cuckoo sandbox nastavení. Přesto se cuckoo sandboxu povedlo zjistit dodatečné informace o procesu, na základě kterých je analytik poté schopen získat podezření o souboru. Analýza serveru ViruTotal nad souborem s hashem **3f63cc0f2e8b677cc1908affd7ed40931944b5af2ffb52cb289dcfa8ce0c2ed1** se sedmnácti pozitivními detekcemi pak označuje soubor jako virus Trojský kůň viz obrázek 2.53.



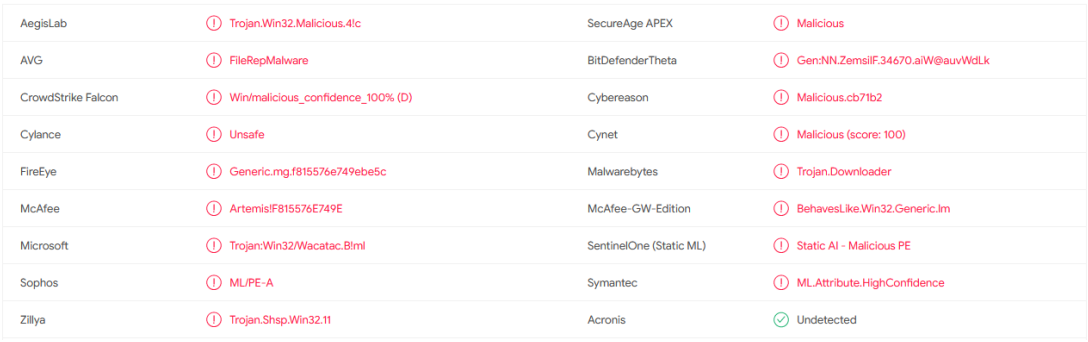
Obr. 2.50: Výpis playbooku prvního incidentu



Obr. 2.51: Náhled VirusTotal prvního incidentu



Obr. 2.52: Výpis playbooku druhého incidentu



Obr. 2.53: Náhled VirusTotal druhého incidentu

2.5.3 Automatické testování playbooku pro kontrolu IP adresy

Pro vyhodnocení výsledku playbooku byly vybrány události ze dnů 9.5.2021 až 15.5.2021. Tento týden testování je zpracován viz tabulka 2.8. Vzhledem k absenci manuálního testování bylo pro tento playbook vytvořeno video, ve kterém je mimo jiné ukázáno manuální spuštění playbooku nad událostí, která byla již kontrolována automaticky. Odkaz na toto video s názvem „Kontrola_IP.mkv“ je k dispozici na přiloženém CD viz příloha B.

Tab. 2.8: Tabulka výsledků automatické analýzy playbooku pro konrolu IP adresy

Datum	Počet událostí	Close	Open	Úspěšné spuštění	Neúspěšné spuštění	False positive	True positive
15.5.2021	10	10	0	10	0	3	7
14.5.2021	13	13	0	13	0	5	8
13.5.2021	8	8	0	8	0	1	7
12.5.2021	7	7	0	4	0	3	4
11.5.2021	15	15	0	15	0	6	9
10.5.2021	11	11	0	11	0	7	4
9.5.2021	14	14	0	14	0	5	9

Hodnocení úspěšnosti tohoto playbooku je přímo závislé na označení události jako **New**, **Open**, nebo **Close**. Pro případ, že je událost z nějakého důvodu nastavena na **Open**, nebo **New**, je třeba prošetřit příčinu takového závěru. Dalším kritériem pro úspěch playbooku je manuální ověření, zda se vyskytla některá z interních IP adres jako předmět šetření. Posledním kritériem úspěchu je vizuální kontrola karet, zda bylo správně rozhodnuto o **false positive**, nebo **true positive** události.

Za dobu týdenního testování byly všechny události nastaveny na **Closed**, dále nebyla nalezena žádná vnitřní IP adresa, která by byla nesprávně šetřena. Nakonec

byly zkontrolovány také karty událostí, které v každém z šetřených případů rozhodly o **false positive**, nebo **true positive** události správně. Výsledkem tohoto automatického týdenního testování je tak chování playbooku dle očekávání, kdy nebyla nalezena žádná odchylka. Playbook je tímto připraven k řešení událostí, kdy vnější podezřelá IP adresa navazuje kontakt s vnitřní adresou a je detekována korelačním pravidlem v systému SIEM. Zároveň tak může být playbook použit jako výchozí scénář určený k modifikaci dle cílové infrastruktury a specifických požadavků.

2.5.4 Automatické testování playbooku pro kontrolu souboru na vzdáleném zařízení

Vzhledem k velkému počtu událostí k prošetření bylo pro vyhodnocení výsledku playbooku vybráno celkově 60 událostí v rozsahu dnů testování od 11.5.2021 do 14.5.2021. V jednotlivých dnech bylo vždy vybráno 15 po sobě jdoucích událostí. Přehled vybraných dní spolu s dalšími informacemi viz tabulka 2.9. Vzhledem k absenci manuální kontroly také v případě tohoto playbooku bylo vytvořeno video s názvem „Kontrola_souboru_vzdalene.mp4“, jehož odkaz na sdíleném disku je obsahem přiloženého CD viz příloha B. Účelem tohoto videa je mimo jiné ukázat manuálně spuštěný playbook nad již automaticky prošetřenou událostí.

Tab. 2.9: Tabulka výsledků automatické analýzy playbooku pro konrolu souboru na vzdáleném zařízení

Datum	Počet událostí	Close	Open	Úspěšné spuštění	Neúspěšné spuštění	False positive	True positive
14.5.2021	15	6	9	13	2	6	0
13.5.2021	15	6	9	13	2	6	0
12.5.2021	15	5	10	11	4	5	0
11.5.2021	15	6	9	11	4	6	0

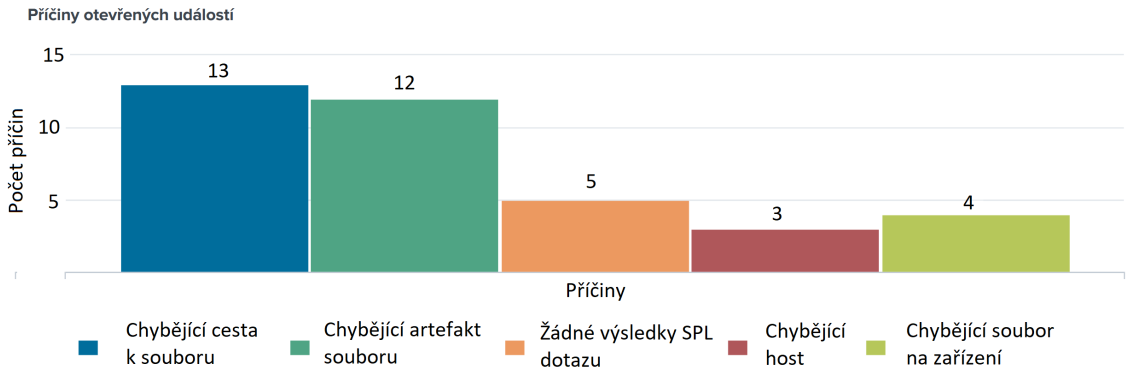
Hodnocení úspěšnosti tohoto playbooku bylo založeno na manuální kontrole jednotlivých událostí. Tato manuální kontrola byla za účelem hledání odlišností od běžného chodu playbooku. Vzhledem k technickým výzvám spojeným s tímto typem playbooku je bráno jako správný výsledek šetření ponechání otevřené události, například z důvodu chybějící cesty k souboru, neschopnosti provést stažení souboru pomocí Windows remote služby, nebo nedohledání parametrů na serveru Splunk Enterprise. Pro události, které byly označeny jako **Close**, bylo předpokladem úspěchu rozhodnutí, zda je soubor vyhodnocen jako **true**, nebo **false positive**.

V rámci testování bylo ověřeno, že každá uzavřená událost, byla správně vyhodnocena v testovaných případech vždy jako **false positive**. Příčiny nastavení události

na **Open**, jsou zpracovány viz tabulka 2.10. V rámci tohoto testování bylo odhaleno celkem 12 chybných spuštění playbooku, které nebyly v souladu s předpokladem. Na základě prověření playbooku se ve všech dvanácti případech jednalo o neimplementování bloku pro zapsání souborového artefaktu do původního artefaktu události. Na obrázku 2.54 je zobrazen graf s poměrem jednotlivých příčin ponechání prošetřované události ve stavu otevřeno. Tento graf byl vytvořen za pomoci nástroje Splunk Enterprise. Z hlediska funkčnosti playbooku lze považovat všechny příčiny, kromě „chybějícího artefaktu souboru“, za ošetřené.

Tab. 2.10: Tabulka příčin otevřených událostí pro kontrolu souboru na vzdáleném zařízení

Datum	Open	Chybějící cesta k souboru	Chybějící artefakt souboru	Žádné výsledky SPL dotazu	Chybějící host	Chybějící soubor na zařízení
14.5.2021	9	4	2	2	0	1
13.5.2021	9	3	2	2	0	2
12.5.2021	10	4	4	1	1	0
11.5.2021	9	2	4	0	2	1



Obr. 2.54: Obrázek grafu příčin ponechání události ve stavu otevřeno

2.6 Opravy playbooků

V této sekci práce budou popsány opravy zjištěných nedostatků testovaných playbooků. Z automatického testování playbooku 2.4.1 vzešly dvě nutné opravy. První

oprava se týká **kontroly prezence přílohy** k analýze a druhá implementace logiky zajišťující vyhodnocení události pro případ, že je **jedna z detonací souboru neúspěšná**. Při testování playbooku pro kontrolu souboru na vzdáleném zařízení popsaném v části práce 2.5.4, byla odhalena chyba spojená s chybějícím blokem zápisu parametru `vaultID` a `filePath`, čímž vznikl **neaktualizovaný artefakt**.

Kontrola prezence přílohy

Pro účely zajištění správného **Hlášení** v případě chybějící přílohy, byla přidána podmínka číslo **11**, která přímo následuje po bloku **get email**. Tato podmínka kontroluje, zda je obsah parametru `vaultId` nenulový. V případě, že ano, je pokračováno k detonaci. Pokud je však kontrolovaný parametr prázdný, playbook přejde k **Hlášení** této skutečnosti pomocí **API** bloku, který událost nastaví na status **Open** a přidá k události zprávu ve formě komentáře a barevně označené karty.

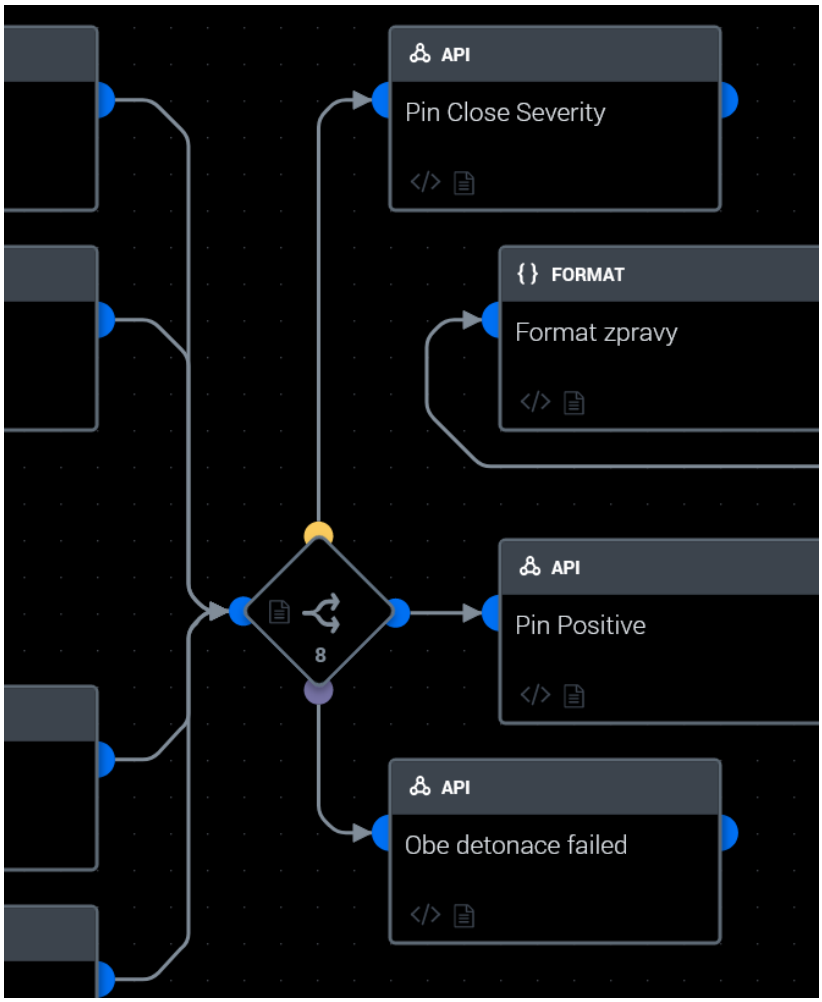
Neúspěšná detonace

Pro odstranění tohoto problému bylo nutné upravit logiku podmínky číslo **8** popsané v části práce 2.4.1. Nově upravená podmínka viz obrázek 2.55. Nově byla přidána podmínka označena fialovou barvou, která je vyhodnocena jako pravda v případě, že je obsahem polí `status` failed, a to jak pro akci detonace na serveru Cuckoo tak na serveru VirusTotal. Poté je proveden **API** blok s názvem **Obe detonace failed**, který provede nastavení statusu události na **Open** a provede **Hlášení** pomocí komentáře a červené karty. **API** blok **Pin Close Severity** nyní vychází ze žluté podmínky vždy provedené v případě, že není provedena ani jedna ze dvou předcházejících podmínek. Modře označená podmínka zůstala nezměněna.

Neaktualizovaný artefakt

Za účelem odstranění chyby v playbooku pro kontrolu souboru na vzdáleném zařízení, bylo provedeno vykopírování již existujících bloků v rámci daného playbooku. Tyto bloky byly implementovány pro odlišnou větev běhu a lze je tak využít i pro větev způsobující chybu. Samotná oprava je k dispozici k nahlédnutí viz obrázek 2.56. Popis funkce jednotlivých bloků a jejich návazností je sepsán v sekci práce 2.4.3.

Těmito finálními opravami byly odstraněny veškeré známé nedostatky playbooků. Všechny playbooky jsou tímto připraveny k praktickému použití, při kterém budou sloužit jakožto výchozí playbooky pro konkrétní implementace dle specifických požadavků pro nasazení. Veškeré playbooky ve finální, opravené podobě, včetně použitých vlastních funkcí jsou součástí přiloženého CD viz příloha B.



Obr. 2.55: Náhled opravy neúspěšné detonace

Obr. 2.56: Náhled opravy neaktualizovaného artefaktu

Závěr

Hlavním cílem diplomové práce bylo navrhnout a implementovat nejméně tři automatické postupy pro řešení bezpečnostních incidentů ve formě vývojových diagramů, a následně je implementovat v prostředí Splunk Phantom v jazyce Python. Dále bylo cílem provést analýzu současného stavu problematiky a popis celého postupu řešení události. Výsledná implementace měla být implementována v rámci experimentálního prostředí a výsledky měly být přehledně analyzovány. Výsledkem diplomové práce je tedy v úvodu provedená analýza současného stavu a vymezení klíčových pojmů důležitých pro tvorbu této práce. Dále bylo vytvořeno experimentální pracoviště popsané pomocí schématu poskytující data pro bezpečnostní události v něm vznikající. Následně byly vytvořeny tři vlastní návrhy postupů pro řešení bezpečnostních událostí, které mají formu vývojových diagramů a zakládají se na zpracování automaticky získávaných událostí. Scénáře takto reprezentují detailní postupy pro řešení bezpečnostních incidentů. Na základě těchto scénářů byly dále vytvořeny vlastní návrhy postupu řešení bezpečnostních incidentů v programovacím jazyce Python v rámci prostředí Splunk Phantom, které se nad automaticky získanými událostmi z vytvořeného experimentálního prostředí spouští. Součástí práce je také ověření funkčnosti postupu v podobě testování. Část manuálního testování zajistila ověření funkce předem navržených chybových stavů scénáře. Další část automatického testování potom ověřila funkci při nasazení scénáře v experimentálním prostředí. Na základě testování byly zjištěny tři nedostatky ošetření chybového stavu, které byly opraveny a zdokumentovány. Na závěr byla ověřena schopnost scénářů správně rozhodovat o false positive nebo true positive událostech, které byly následně přehledně analyzovány formou tabulek a manuálně zkontrolovány.

Tímto byly veškeré stanovené cíle diplomové práce splněny. Výsledkem práce jsou tak tři funkční scénáře pro automatické řešení incidentů ve firemním prostředí, které mohou sloužit jako funkční základ pro rozšířené scénáře upravené dle specifických potřeb cílového informačního systému.

Literatura

[1] DRASTICH, Martin. *Systém managementu bezpečnosti informací*. Praha: Grada, 2011. Průvodce (Grada). ISBN 978-80-247-4251-9.

[2] VAN DER KLEIJ, Rick, Geert KLEINHUIS a Heather YOUNG. *Computer Security Incident Response Team Effectiveness: A Needs Assessment*. Frontiers in Psychology [online]. 2017, 8, 1-8 [cit. 2020-11-25]. ISSN 1664-1078. Dostupné z: <https://doi.org/10.3389/fpsyg.2017.02179>.

[3] ČESKO. *Zákon č. 181/2014 Sb. ze dne 23. července 2014, o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti)*. In: Zákony pro lidi.cz [online]. © AION CS 2010-2020 [cit. 9. 12. 2020]. Dostupné z: <https://aplikace.mvcr.cz/sbirka-zakonu/ViewFile.aspx?type=z&id=27231> .

[4] KRÁL, David. *Informační bezpečnost podniku*. 2010.

[5] *Guide to integrating forensic techniques into incident response*. 2006. Aug 2006.

[6] ČSN ISO/IEC 27035 -2 (36 9799) *Informační technologie – Bezpečnostní techniky – Řízení incidentů bezpečnosti informací – Část 2: Směrnice pro plánování a přípravu odezvy na incidenty*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2018. Česká technická norma. ISBN 978 0 580 80186 0.

[7] ČSN ISO/IEC 27003 (36 9790) *Informační technologie - Bezpečnostní techniky - Směrnice pro implementaci systému řízení bezpečnosti informací*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2011. Česká technická norma.

[8] LUQUE, Amalia, et al. *The impact of class imbalance in classification performance metrics based on the binary confusion matrix*. Pattern Recognition, 2019, 91: 216-231.

[9] GRANCE, Tim; KENT, Karen; KIM, Brian. *Computer security incident handling guide*. NIST Special Publication, 2004, 800.61: 11.

[10] LONVICK, Chris. RFC3164: *The BSD Syslog Protocol*. 2001.

[11] VAARANDI, Risto. *A data clustering algorithm for mining patterns from event logs*. In: Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003)(IEEE Cat. No. 03EX764). IEEE, 2003.

[12] IBM Knowledge Center [online]. [cit. 2020-12-11]. Dostupné z: <https://www.ibm.com/support/knowledgecenter/en/>.

[13] Juniper networks [online]. Juniper Networks, c2020 [cit. 2020-12-11]. Dostupné z: <https://www.juniper.net/documentation/>.

[14] BACE, Rebecca; MELL, Peter. *NIST special publication on intrusion detection systems*. BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, 2001.

[15] Forcepoint [online]. Forcepoint, c2020 [cit. 2020-12-11]. Dostupné z: <https://www.forcepoint.com/>.

[16] Exabeam [online]. Exabeam, c2020 [cit. 2020-12-11]. Dostupné z: <https://www.exabeam.com/>.

[17] KOSŤ, Dávid Kosť. *SOAR: Security Orchestration, Automation and Response*. 23 července, 2019 [cit. 2020-12-11]. Dostupné z: <https://www.itsec-nn.com/soar-security-orchestration-automation-and-response/>.

[18] Swimlane [online]. Louisville, Colorado USA: Swimlane, c2020 [cit. 2020-12-11]. Dostupné z: <https://swimlane.com/>.

[19] Fortinet [online]. Fortinet, c2020 [cit. 2020-12-11]. Dostupné z: <https://www.fortinet.com/>.

[20] Paloalto [online]. Palo Alto Networks, c2020 [cit. 2020-12-11]. Dostupné z: <https://www.paloaltonetworks.com/>.

[21] Splunk [online]. United States: Splunk, c2020 [cit. 2020-12-11]. Dostupné z: <https://www.splunk.com/>.

[22] Splunk Phantom [online]. United States: Splunk, c2020 [cit. 2020-12-06]. Dostupné z: <https://www.splunk.com/en_us/software/splunk-security-orchestration-and-automation.html>.

[23] Splunk docs [online]. United States: Splunk, c2020 [cit. 2020-12-5]. Dostupné z: <https://docs.splunk.com/Documentation>.

[24] VirusTotal [online]. [cit. 2020-12-06]. Dostupné z: <https://www.virustotal.com/gui/>.

[25] Windows Remote Management. Microsoft: docs [online]. USA: Microsoft, c2021, 05/31/2018 [cit. 2021-4-9]. Dostupné z: <https://docs.microsoft.com/en-us/windows/win32/winrm/portal>.

[26] Elastic Docs [online]. United States: Elasticsearch B.V., c2020 [cit. 2020-12-06]. Dostupné z: <https://www.elastic.co/guide/index.html>.

[27] SANS [online]. SANS, c2021 [cit. 2021-4-27]. Dostupné z: <https://www.sans.org/>.

Seznam symbolů, veličin a zkratek

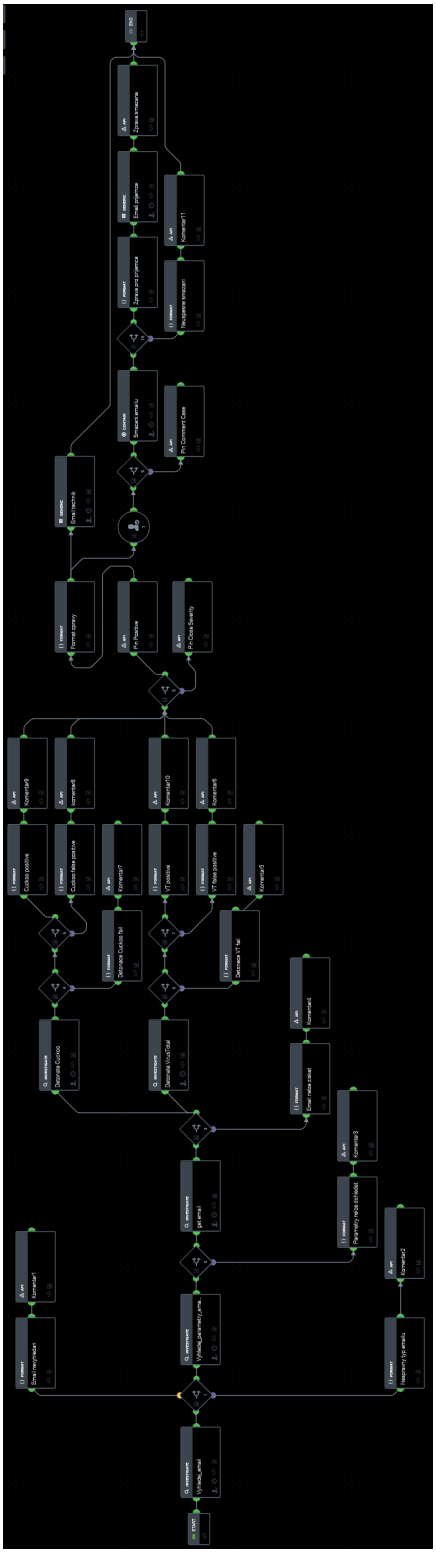
API	Application Programming Interface
CaC	Command and Control
CEF	Common Event Format
DDOS	Distributed Denial of Service
DOS	Denial of Service
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IOC	Indicator of Compromise
IPS	Intrusion Prevention System
IP	Internet Protocol
IRT	Incident Response Team
ISO	Interantional Organazation for Standardization
JSON	JavaScript Object Notation
LAN	Local Area Network
NIST	National Institute of Standards and Technology
REST	Representational State Transfer
RFC	Request for Comments
SIEM	Security Information and Event Management
SOAR	Security Orchestration, Automation, and Response
SPL	Search Processing Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator

WAN	Wide Area Network
winRM	Windows Remote Management

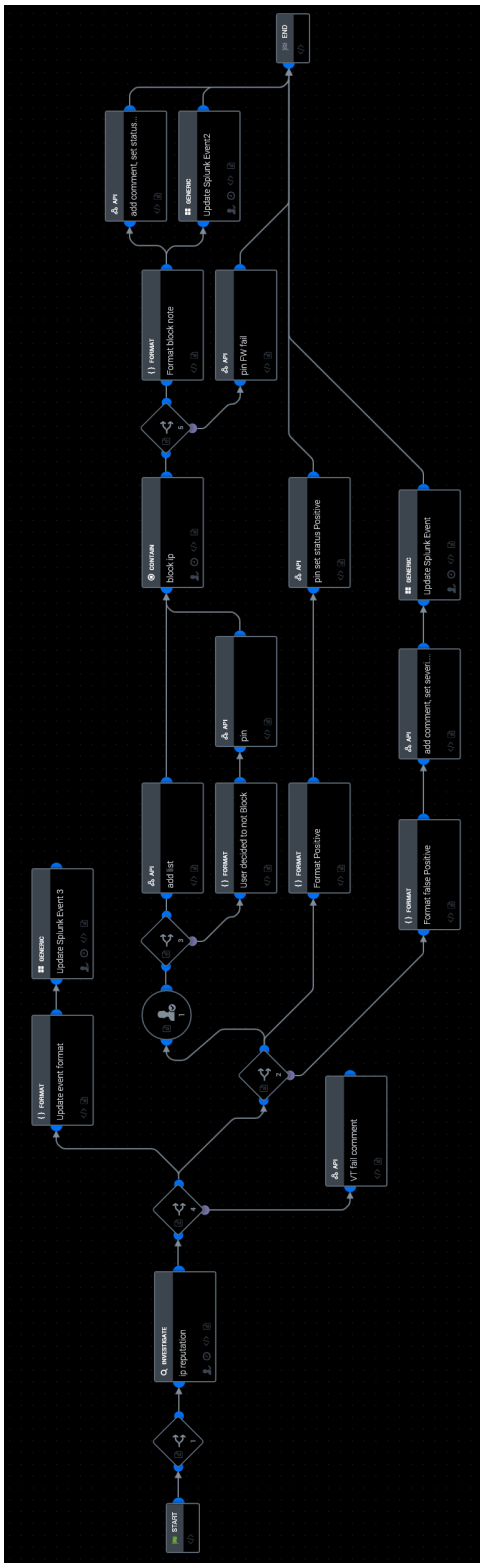
Seznam příloh

A Obrázky

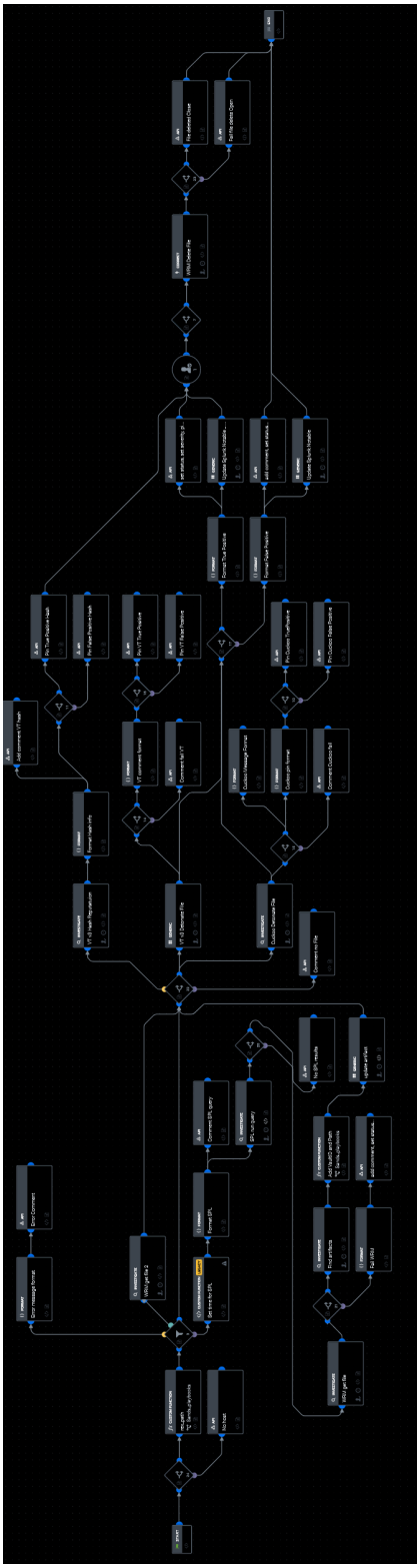
A.1 Náhled celého playbooku Kontrola přílohy emailu



A.2 Náhled celého schéma playbooku pro kontrolu IP adresy



A.3 Náhled celého playbooku pro kontrolu souboru na vzdáleném zařízení



B Obsah přiloženého CD/DVD

- / ..... kořen přiloženého CD
  - audiovizualni_obsah ..... Doplňkové demonstrační videa
    - odkaz_na_audiovizualni_obsah.txt
  - custom_functions ..... Vlastní funkce užité v playboocích
    - Format_json_for_artefact_VaultID_FilePath.json
    - Format_json_for_artefact_VaultID_FilePath.py
    - rex_path.json
    - rex_path.py
  - playbooky ..... Kompletní kód playbooků
    - Endpoint Malware DEV DP.json
    - Endpoint Malware DEV DP.py
    - Investigate Remote IP DP.json
    - Investigate Remote IP DP.py
    - Kontrola_prilohy_emailu_automat_DP.json
    - Kontrola_prilohy_emailu_automat_DP.py
  - readme.txt ..... Popis přílohy