TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Martin Chmelař 184660IVCM

Utilizing MITRE ATT&CK to create Adversary Reports of live-fire Cybersecurity Exercises for feedback purposes

Master's Thesis

Technical Supervisor
Olaf Manuel Maennel
PhD

Tallinn 2020

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author:	Martin Chmelař	....
		(signature)
Date:	May 19, 2020	

Abstract

Cybersecurity is a rapidly developing field which puts high pressure on professionals dealing with this domain. There are different approaches to keeping a pace with cyber criminals - one of them is cybersecurity exercise. The experience gained by participants provides effective learning outcomes in general. The good practice is sharing information with other colleagues in the cyber domain. Although it might be not always possible to share incident information, there are sufficient examples of sharing information about cyber incidents - one of them is MITRE ATT&CK.

This thesis is devoted to the development of a method for creation of feedback-session reports from cybersecurity exercises based on MITRE ATT&CK. Precisely, it demonstrates what Red Teamer's progress is possible to find out from data gathered during cybersecurity exercises organised by NATO CCDCOE according to MITRE ATT&CK.

The method was created by performing action research consisting of four interventions. The objective of the research is what adversary (Red Team's) techniques are possible to distinguish from a given dataset and what information supporting learning outcomes are possible to provide. The research operates with dataset gathered from Crossed Swords 2020.

Implementation of the method on the dataset results into creation of report in markdown format consisting of initial information and reading instructions, graphs of used techniques, their mitigations, detections and data sources, description of each detected technique and details for Red and Blue Teamers. Research shows that it is subjectively easy to follow reports of shorter timespans within one domain. On the other hand, the report from the whole event might be hard to understand.

The main contribution of this thesis is a novel method of creating reports for cybersecurity exercises feedback sessions based on MITRE ATT&CK. The report provides useful information chiefly for Red Teamer and Blue Teamers. Creation of feedback report also requires manual work despite the fact that a python script processes dataset. Ultimately, the feedback report shall be customised according to the needs of exercise participants.

List of abbreviations and terms

CCDCOE	The NATO Cooperative Cyber Defence Centre of Excellence
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge)
APT	Advanced persistent threat
ICS	Industrial control systems
JSON	JavaScript Object Notation
JSONl	JSON Lines
STIX	Structured Threat Information Expression
CTI	Cyber threat intelligence
TRAM	Threat Report ATT&CK
TTPs	Tactics, Techniques and Procedures
IOCs	Indicators of compromise
IP	Internet Protocol
CAR	Cyber Analytics Repository
SOC	Security Operations Center
CSE	Cybersecurity exercises
AAR	After action report
SEC	Simple Event Correlator
IDS	Intrusion detection system
IPS	Intrusion prevention system
SIEM	Security information and event management
CSV	Comma-separated values
API	Application programming interface
XS20	Crossed Swords 2020
RAM	Random-access memory
CPU	Central processing unit
SVG	Scalable Vector Graphics

Table of Contents

List of Figures

List of Tables

1. Introduction

As cybersecurity is rapidly developing, it also brings a dozen of new findings. Cybersecurity professionals have to keep updated since new threats are appearing every day. Therefore there is significant pressure for effective learning. Red – Blue Team cyber exercises are considered as an effective way of developing and evaluating professional's skills. Most of the efforts are concentrating on preparations and actual execution, less likely on feedback session. Since individuals can learn a lot from proper feedback report and feedback session, it is also vital to insist on this crucial part of cyber exercises.

1.1 Motivation

Red teaming (adversary emulation) is one of the best ways for understanding the preparation of Blue Teamers (defenders) [1]. The under-attack experience can reveal crucial weaknesses of the defence team [1]. While penetration testing reveals technical vulnerabilities, Red teaming is objectively focused and therefore provide adversary paradigm in order to benefit from the attacked target [1]. In other words, penetration testing is about testing as many vulnerabilities as possible while Red teaming is about getting in and access sensitive information in any way possible, as quietly as possible [2]. Therefore Red-Blue teaming exercises differs from penetration testing assessment by giving the outstanding experience of incident response - technical capabilities along with decision-making process[1].

Cybersecurity exercise is typically heavy-sensored environment which allows further event analysis [3]. There are a lot of great approaches for gathering data of actual progress as full packet capture, intrusion detection systems, intrusion prevention systems, event loggers and log files. On the other hand outcomes of previously mentioned approaches might be overwhelming and therefore time-consuming while understanding teams progress.

The abstraction of Red and Blue Teams approaches is necessary for better understanding of broader spectre of involved people. Several frameworks are dealing with adversary approaches such as Cyber Kill Chain by Lockheed Martin, Mitre ATT&CK, STRIDE and many others. The ATT&CK framework is becoming more and more popular, and it is progressively applied to many use cases. This thesis is dedicated to application of MITRE ATT&CK on cybersecurity exercises in order to enhance feedback experience.

1.2 Problem statement and contribution

What Red Teamer's progress is possible to find out from data gathered during cybersecurity exercise according to MITRE ATT&CK?

The goal of contribution is the creation of a method for creating reports based on Red Team behaviour for feedback session purposes.

1.3 The scope

The primary goal of this thesis is designing a method for creation of adversary technical report based on MITRE ATT&CK. The source dataset was gathered during Crossed Swords 2020 cyber exercise organised by the NATO Cooperative Cyber Defence Centre of Excellence (CCDCOE).

1.4 Novelty

This thesis provides a novel method of report creation from live cybersecurity exercises. The method combines data from ATT&CK, the Atomic Red Team and enriches reports of contextual graphs.

Post-mortem analysis of data generated by Peek, part of Frankenstack, during Crossed Swords 2020 likewise provides the novelty of this thesis.

1.5 Limitations

The limitation of this thesis is the dependency of data generated by Peek, which is part of Frankenstack. Secondly, due to the global pandemic of COVID-19, validation of generated reports could not be fully enforced. Original validation plan counted on active participation at Locked Shields 2020, another cyber exercise organised by CCDCOE. Unfortunately, this event was cancelled, and therefore it was not possible to validate the reporting method against actions taken by Red Team as well as test report user-experience. Data from Crossed Swords 2020 were used as a backup plan.

1.6 Acknowledgements

I would like to thank my supervisor, Olaf Manuel Maennel, for awesome support of my progress and finding all crucial resources.

Also, I would like to thank Mauno Pihelgas for providing access to the dataset from Crossed Swords 2020 and giving valuable inputs.

I would like to thank Juhan-Peep Ernits for setting up and maintaining the machine as my test environment.

Lastly, I would like to thank my family and friend for the surpassing support.

2. Cyber threat modelling

Cyber threat modelling gives a certain level of abstraction of adversary behaviour [3, 4]. This chapter lists an overview of diverse approaches to treat modelling based on abstraction level and its purpose. The first part lists high-lever abstraction approaches, while the second part provides an overview of low-level concepts.

2.1 High-level models

High-level models provide different paradigms. There are models for risk management, specific topic-oriented system design, threat information sharing and models dedicated to TTPs (tactics, techniques, and procedures These models have some common elements. Typically it is cyber-attack lifecycle models, cyber kill chain, attack trees or attack graphs [4].

2.1.1 Frameworks using a general risk management approach

Following models describe management of risks based on cyber resources dependency, taking into consideration the existence of bad actors in cyberspace [4].

- NIST Framework for Improving Critical Infrastructure Cybersecurity,
- CBEST Intelligence-Led Cyber Threat Modelling,
- COBIT 5 and Risk IT [4].

Cybersecurity framework created by NIST uses everyday language in order to make it easy to understand [5]. It consists of three components - Core, Tiers and Profiles. The Core represents desired cybersecurity activities, and outcomes using common language enables to group and review mitigations for identified threats [5]. Tiers describe how an organisation views cybersecurity risk management [5]. Profiles describe the identification and prioritisation of opportunities for improving cybersecurity within an organisation [5].

CBEST drafts goals, capabilities used to pursue these goals, methods and patterns of operation of cyber threat actors [4]. It uses gathered cyber threat intelligence which gives organisations a clear understanding of their level of exposure, and ability to detect and respond to real risks [6].

COBIT 5, an end-to-end business-oriented framework, builds on standards of the International Organization for Standardization (ISO) such as ISO 38500 (model for the corporate governance for IT), ISO 15504 and ISO 27001 [4, 7]. Attack scenario describes threat type, threat actor, type of event, affected assets and time [4].

2.1.2 Topic-focused frameworks and methodologies

Frameworks and methodologies listed in this part focus on specific topics rather than a generalisation of adversary behaviour. Examples:

- Cyber Prep Adversary Characterization Framework,
- Insider Threat Modeling,
- Threat Characterization Framework Developed for DRDC,
- the Cyber Kill Chain® by Lockheed Martin [4].

Cyber Prep Adversary Characterization Framewor, created by MITRE, operates on the organisational level of risk management [4]. This framework is designed for governance, operations, and Architecture and Engineering while describing fourteen aspects of organisational preparedness [4].

Cyber Kill Chain model uses military terminology, while variant attack lifecycles are typical. The main objective is the exfiltration of sensitive information [4].

Insider threat modelling focuses on employees and staff behaviour in terms of potential threat. This model takes into consideration the motivation and process of becoming a threat. It also provides the analysis and prediction of its effects [4].

2.1.3 Design analysis and testing process

Following models were designed in order to support system design and development process as well as motivate and support system design decisions [4].

- STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) by Microsoft;
- DREAD (Damage, Reliability, Exploitability, Affected Users, and Discoverability) by Microsoft;

- OCTAVE by Carnegie Mellon Software Engineering Institute;
- Threat Agent Risk Assessment (TARA) and Threat Agent Library (TAL) by Intel;
- IDDIL/ATC (Identify the assets; Define the attack surface; Decompose the system; Identify attack vectors; List threat actors; Analysis & assessment; Triage; Controls.) by Lockheed Martin [4].

DREAD and STRIDE were both developed by Microsoft. STRIDE provides a categorisation of general threat vectors and helps analysts to identify the complete threat model [4]. Threat modelling with STRIDE should start with a question "What are you building?" considering components and trust boundaries which reveals weak spots in the service design process [4]. DREAD builds on STRIDE what is concerned with the identification of threat vectors. Each vector has its score calculated as an average of five elements. Microsoft stopped using DREAD in 2010. Nevertheless, it still appears within the cybersecurity community [4].

OCTAVE framework consists of eight steps, in general deals with identifying areas of concern)and developing threat scenarios, represented as threat trees [4].

TARA and TAL approach firstly identifies most exposed threat agents, their objectives they want to accomplish and methods cross-referenced with vulnerabilities they should use [4].

IDDIL/ATC deploys a process for applying the cyber kill chain model alongside with its variant of STRIDE enriched of lateral movement and attack trees [4].

2.1.4 Frameworks supporting threat information sharing:

Cybersecurity specialists benefit from threat information sharing. To achieve this, following approaches of threat modelling are present:

- STIX$^{\text{TM}}$ (the Structured Threat Information eXpression) by OASIS;
- PRE-ATT&CK$^{\text{TM}}$ (Adversarial Tactics, Techniques & Common Knowledge for left-of-exploit) by MITRE;
- Cyber Threat Framework by ODNI [4].

Structured Threat Information eXpression (STIX) is a structured language created for the sharing of information about cyber threat and defend actions. PRE-ATT&CK is a lexicon of pre-exploit adversary behaviour, its mitigation and detection [4]. Cyber defenders may prioritise taken action based on PRE-ATT&CK [4].

Cyber threat framework initially supported information sharing in standard structure of information in published threat reports [4]. Additionally, it supports analysis, advanced decision making and trend and gap analysis due to its approach towards characterising and categorising adversary activities [4]. Categories in PRE-ATT&CK correspond either to objectives or to actions in the cyber threat framework [4].

2.1.5 Threat models describing adversary tactics, techniques, and procedures (TTPs):

Threat models considering TTPs are rarely based on frameworks mentioned in previous sections. There are two approaches. The first approach operates with adversary capabilities and attack techniques in a general technological environment, while the second approach consists of threat models for particular enterprises. Enterprise oriented models often have sensitive information and therefore, are not shared broadly [4].

Enterprise-Neutral, Technology-Focused:

- Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK$^{\text{TM}}$) by MITRE;
- Common Attack Pattern Enumeration and Classification (CAPEC$^{\text{TM}}$) by MITRE;
- Open Web Application Security Project (OWASP) [4].

MITRE ATT&CK provides a detailed characterisation of adversary behaviour. The second chapter (3) of this thesis gives further description of ATT&CK. CAPEC, also developed by MITRE, implements the knowledge of attack patterns and classification taxonomy[4]. OWASP gathers knowledge about web application security and develops several related projects. OWASP Top Ten describes most severe threats of web applications while describing threat agents or attack vectors, security weakness and impacts and further description [8].

Examples of Enterprise-Oriented, Technology-Focused frameworks:

- Threat Assessment and Remediation Analysis (TARA) by MITRE
- NIPRNet/SIPRNet Cyber Security Architecture Review (NSCSAR) by the U.S. Department of Defense;
- Notional Threat Model for a Large Financial Institution [4].

TARA is a catalogue consisting of vector groups organised in taxonomy and includes tools for matching a specified system environment [4]. Analysis using TARA starts by setting the scope, architecture, technology components, and types of adversaries [4].

NSCSAR describes four phases - pre-event, get in, stay in, act based on ATT&CK and other models. Since the threat model includes information from classified sources as well, it can not be shared [4]. The objective of Notional Threat Model for a Large Financial Institution is supporting deployed mitigations of residual risks [4]. It is delivered as mindmap and contains Business institution-specific assets.

2.2 Low-level concepts

Databases oriented on a technical description of exploits, vulnerabilities and weaknesses in the context of threat modelling:

- CVE (Common Vulnerabilities and Exposures): `cvedetails.com`, `nvd.nist.gov`, `cve.mitre.org` [9]
- CWE (Common Weakness Enumeration): `cwe.mitre.org`
- Exploit & vulnerability databases `exploit-db.com`, `cxsecurity.com/exploit`, `rapid7.com/db/` [10]

CVE reference list identifies and categorises publicly disclosed security vulnerabilities and exposures in software. Every CVE record obtains identifier from MITRE Corporation [9]. Typically, CVE lists do not contain further information on the risks, the fixes or further technical data [9].

CWE stands for Common Weakness Enumeration. It is a list of common hardware and software weaknesses developed by the community [11]. The goal of using CWE is educating developers, architects, designers and programmers in order to prevent security vulnerabilities before delivering the product [11]. CWE records describe weaknesses in everyday language, checks for weaknesses in existing IT products, evaluate coverage of tools developed for exploiting weaknesses, leverage standard for weakness identification mitigation and prevention and lastly prevent vulnerabilities before development [11].

Exploit databases provides valuable information about exploits usually linked with CVE or CWE.

3. MITRE ATT&CK

The first part of this chapter is devoted to philosophy, technical description of ATT&CK. Followingly, up-to-date (spring 2020) related projects and use cases are described.

3.1 MITRE corporation

The MITRE Corporation, a vendor of ATT&CK framework, is a US-based not-for-profit organisation consisting of 7 centres of research where one of the centres is dedicated to cybersecurity. MITRE corporation states that its goal is "Solving problems for a safer world. " [12, 13].

3.2 ATT&CK

Back to 2010, there was a need for systematical documentation of cyber adversary conducted from exercises as a part of the Fort Meade Experiment [3]. Researchers of MITRE were enabled to deploy and evaluate tools for better detection of advanced persistent threats (4.3.1).

ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) is the knowledge base of adversary behaviour in cyberspace. Whole adversary taxonomy is present – from pre-attack phase to post-exploitation. ATT&CK fills the gap between high-abstraction level threat models and low-abstraction level concepts as states Figure 1. Detailed technical information enriches high-level processes of attack life cycles (kill chains); exploit, vulnerabilities and weakness databases are getting into context of attack vector.

The ATT&CK matrix visualises and describes the relationship between tactics and techniques (as shown in Figure 2). Tactics stand for the reason of attacker behaviour while techniques describe the way of achievement and potential gain of tactical objective [3].

ATT&CK contains three technology domains – Enterprise, Mobile and Industrial Control systems [14].

- Enterprise domain matrice covers Windows, Linux, macOS and cloud solutions.

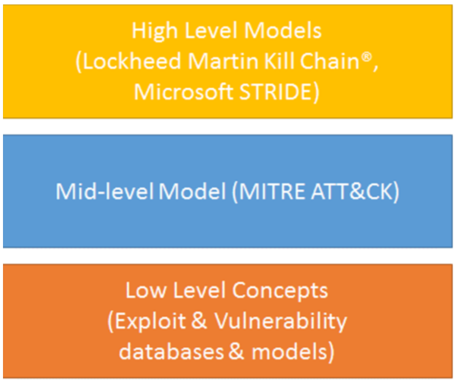
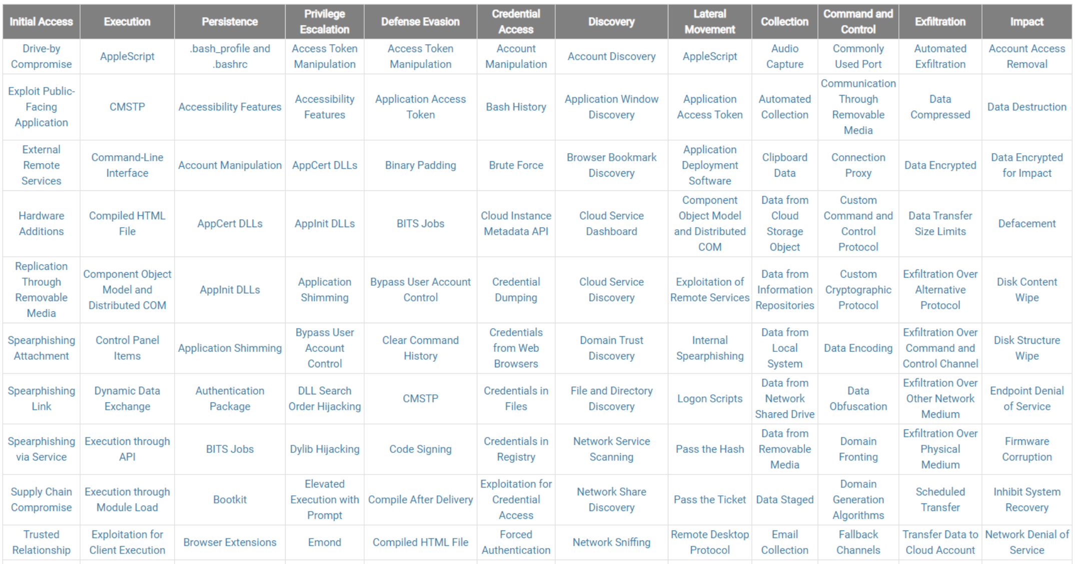


Figure 1. *Abstraction Comparison of Models and Threat Knowledge Databases , from [3]*



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Application Access Token	Bash History	Application Window Discovery	Application Access Token	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Application Deployment Software	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	BITS Jobs	Cloud Instance Metadata API	Cloud Service Dashboard	Component Object Model and Distributed COM	Data from Cloud Storage Object	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Bypass User Account Control	Credential Dumping	Cloud Service Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	Clear Command History	Credentials from Web Browsers	Domain Trust Discovery	Internal Spearphishing	Data from Local System	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	CMSTP	Credentials in Files	File and Directory Discovery	Logon Scripts	Data from Network Shared Drive	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Code Signing	Credentials in Registry	Network Service Scanning	Pass the Hash	Data from Removable Media	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compile After Delivery	Exploitation for Credential Access	Network Share Discovery	Pass the Ticket	Data Staged	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Compiled HTML File	Forced Authentication	Network Sniffing	Remote Desktop Protocol	Email Collection	Fallback Channels	Transfer Data to Cloud Account	Network Denial of Service

Figure 2. *Omitted example of ATT&CK matrix, from [14]*

- Mobile domain covers Android and iOS.
- Industrial control systems domain has a new tactic field "Impact" and "Inhibit Response Function". Since ICS is a highly heterogeneous environment, ATT&CK does not necessarily cover all assets [15].

PRE-ATT&CK matrice covers preparation phase, including supportive, not necessarily technical adversary behaviour such as information gathering or fake persona development [3].

Apart from adversarial tactics and techniques, ATT&CK contains knowledge base dedicated to mitigations of techniques used by an adversary. Mitigations section has two subcategories: Enterprise and Mobile [16].

Section "Groups" represents known threat groups - sets of related intrusion activity [17]. Each group entry has detailed fundamentals, associated group descriptions, used techniques, used software and references.

Entries of "Software" section are Tools and Malware. Software used by system administrators, defenders, penetration tester, Red Teamers or adversary is marked as tools. For instance, Metasploit, ipconfig and netstat are categorised as tools. Software for malicious purposes is identified as Malware. Example of malware entry is BUBBLEWRAP [18]. Each software entry includes a brief description, used techniques, list of groups using current software and reference

Content of the framework was created based on publicly available adversary reporting resources such as:

- Threat intelligence reports,
- Conference presentations,
- Webinars,
- Social media,
- Blogs,
- Open source code repositories,
- Malware samples [3].

The framework is updated approximately every 3 to 6 months by the MITRE Corporation [19]. The update is based on new observations gathered by MITRE as well as community contributions. Anyone can contribute to using their research results.

Evaluations of software

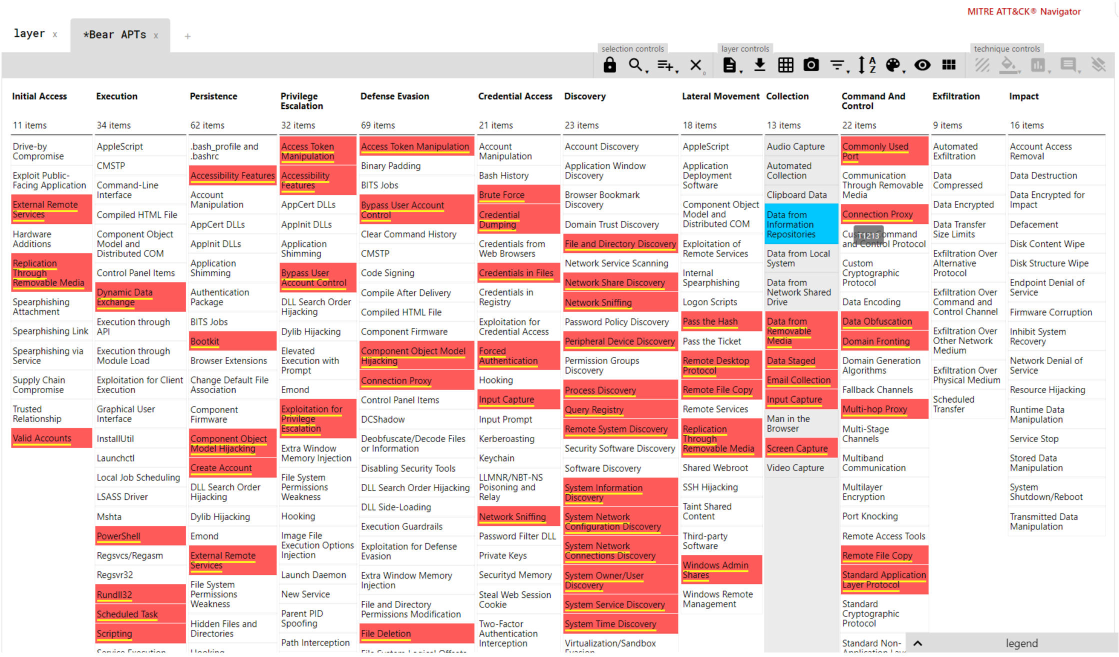
MITRE corporation is also independently evaluating end-point cybersecurity solution according to publicly available methodology `https://attackevals.mitre.org/adversary-emulation.html` based on previously known adversaries behaviours. The goal is not to select the winner using any scoring system but rather empower end-user by presenting objective insights [20]. Any vendor of end-user cybersecurity products can get evaluated.

3.2.1 Related projects

Following projects were developed for Red Teams, Blue Teams and purple teams purposes based on ATT&CK framework.

Navigator

Navigator is an open-source tool for behaviour visualisation of attackers, groups or software. Users can operate with public version operating at `https://mitre-attack.github.io/attack-navigator/` or run their instance on the local machine. Navigator currently enables working with Enterprise and mobile domain as well as covering both the preparation and action stage. User can create multiple layers and combine existing layers. As an example, techniques coverage of Fancy Bear threat group represents Figure (3). Apart from that, a user can benefit from other functions, for instance, adding comments, sorting and exporting the layer to other file formats. Navigator's data are stored in standardised JSON format while the full definition is listed at `https://github.com/mitre-attack/attack-navigator/blob/master/layers/LAYERFORMATv2_2.md`



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	34 items	62 items	32 items	69 items	21 items	23 items	18 items	13 items	22 items	9 items	16 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Input Capture	Fallback Channels		Network Denial of Service
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Man in the Browser	Multi-hop Proxy		Resource Hijacking
	InstallUtil	Component Firmware	Extra Window Memory Injection	Connection Proxy	Input Prompt	Process Discovery	Replication Through Removable Media	Screen Capture	Multi-Stage Channels		Runtime Data Manipulation
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Control Panel Items	Kerberoasting	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Service Stop
	Local Job Scheduling	Create Account	Hooking	DCShadow	Keychain	Remote System Discovery	SSH Hijacking		Multilayer Encryption		Stored Data Manipulation
	LSASS Driver	DLL Search Order Hijacking	Image File Execution Options Injection	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning and Relay	Security Software Discovery	Taint Shared Content		Port Knocking		System Shutdown/Reboot
	Mshta	Dylib Hijacking	Launch Daemon	Disabling Security Tools	Network Sniffing	Software Discovery	Third-party Software		Remote Access Tools		Transmitted Data Manipulation
	PowerShell	Emond	New Service	DLL Search Order Hijacking	Password Filter DLL	System Information Discovery	Windows Admin Shares		Remote File Copy		
	Regsvcs/Regasm	External Remote Services	Parent PID Spoofing	DLL Side-Loading	Private Keys	System Network Configuration Discovery	Windows Remote Management		Standard Application Layer Protocol		
	Regsvr32	File System Permissions Weakness	Path Interception	Execution Guardrails	Securityd Memory	System Network Connections Discovery			Standard Cryptographic Protocol		
	Rundll32	Hidden Files and Directories		Exploitation for Defense Evasion	Steal Web Session Cookie	System Owner/User Discovery			Standard Non-		
	Scheduled Task	Hooking		Extra Window Memory Injection	Two-Factor Authentication Interception	System Service Discovery					
	Scripting			File and Directory Permissions Modification		System Time Discovery					
	Service Execution			File Deletion		Virtualization/Sandbox Evasion					
				File System Logical Offsets							

Figure 3. *Coverage of Fancy Bear group visible in Navigator*

CASCADE

CASCADE is a research project created for automation of Blue Team analysis of suspicious behaviour rather than mitigations [21]. The cascade server runs analysis against data stored either in Elasticsearch or Splunk [22]. Alerts and Events are triggered, and the server generates a graph of adversary behaviour. The graph is tagged with further information from ATT&CK [22].

CALDERA

CALDERA is a framework for running autonomous and manual red-team tests [23]. As stated during spring 2020, CALDERA is under active development stage.

Atomic Red Team

Atomic Red Team is a framework created and maintained by Red Canary in order to execute atomic tests according to ATT&CK techniques from the Red teaming point of view. The philosophy of the framework stands on the fact that cybersecurity teams need to test security controls and its outcomes, the tests should be possible to run within 5 minutes, and the cybersecurity community needs to keep learning how adversaries are operating [24].

STIX

ATT&CK supports threat information sharing using STIX. Structured Threat Information Expression is a machine-readable language enabling sharing information about cyber threat intelligence (CTI) [25]. Sharing of CTI improves cooperation and faster understanding of cyber threats and enables further approaches such as collaborative threat analysis [25]. TAXII (Trusted Automated Exchange of Intelligence Information) server is commonly used for secure sharing of CTI between organisations [26].

TRAM

Threat Report ATT&CK Mapping is a tool that helps analysts to extract techniques used during the attack from existing reports. TRAM uses natural language processing, and the user can accept or reject the suggested technique [27]. It is possible to export the report in PDF format for the moment.

3.2.2 Use cases

Adversary Emulation

Adversary emulation is the process of security assessment of the domain using cyber threat intelligence [3]. Typically APT (4.3.1) groups are emulated according to the plan which includes commonly known behaviour and TTPs used by that group [28]. This information was documented from publicly available reports [3]. The customer, typically an organisation, can verify whether their detection and potentially mitigation are working correctly.

Red Teaming

ATT&CK offers new possibilities to Red Teamers in order to plan their campaigns. Red teaming might turn into routine emulation of APTs (4.3.1) and always using similar TTPs, which are "popular" at the moment [3]. Red Teamers may consider new planning approaches in order to counter the planning routine. One of them is merely rolling the dice over Tactics and choose 2-3 techniques for each Tactic [29]. This procedure encourages Red Teamers to step out of their comfort zone and becoming better; thus, Blue Teamers are getting the chance of developing defences against singular attack vectors [29].

Behavioural analytics development

Many contemporary defences focus on signatures or indicators of compromise (IOCs). Since attackers might easily evade IOCs by simple changes such as different hashes, IP addresses or different infrastructure, this approach might be insufficient [3, 30]. Therefore Behavioural analysis is used for detection of adversary behaviour. It is possible to use ATT&CK and Cyber Analytics Repository (CAR) as useful tools to develop analytics for adversarial behaviour [3, 30].

Defensive gap assessment

When an organisation needs to assess gaps in current mitigations and visibility of defences, ATT&CK can serve as a behaviour-focused adversary model for assessment. Identification of gaps shall help to prioritise investments into further security tools for decision-makers. Adversary emulation (3.2.2) and Red teaming (3.2.2) should be used in order to identify these gaps [3].

SOC maturity assessment

Security Operations Center is an essential part of enterprise networks that monitors and mitigates the threats. For determination of SOC effectiveness, ATT&CK can serve as one of the measurements [3]. "SOC Maturity assessment focuses on the processes a SOC uses to detect, understand, and respond to changing threats to their network over time." [3].

Cyber threat intelligence enrichment

Cyber threat intelligence allows organisations to protect their cyber-infrastructure from an attack better based on shared knowledge, while 75% of IT-security decision-makers states that CTI has a high or critical priority for their organisation [31]. ATT&CK can be beneficial for documentation and behaviour understanding of adversary groups [3].

Purple teaming

A Purple Team consists of Red and Blue teaming approach. Therefore offensive cyber operations are run together with cyber defence analyst within one network and evaluate the effectiveness of taken controls as a conclusion. Red-Blue cooperation can have numerous benefits [32]. Purple teaming is beneficial in many cases, especially when Red Teamers go of scope or use the information they should not in order to "win" against a Blue Team and vice versa Blue Teamers can set up too restrictive countermeasures. Therefore efforts of separated Red Teamers and Blue Teamers can lead to providing false metrics of actual security capability [33]. Purple teamers can use Vectr (`https://vectr.io/`) tool, which is based on ATT&CK framework to plan and document their progress [34].

4. Cybersecurity exercises

This chapter will briefly describe the main differences between exercises types and their purpose, organisation problematics, commonly used tools and solutions. Further describes the problematics of adversary behaviour analysis.

Cybersecurity exercises (CSE), also known as wargaming, are an excellent tool for training of personnel working for governmental and commercial organisations.

There are three main types of Cybersecurity Exercises divided by its characteristics and usage [35].

1. **Tablet top** exercise operates with hypothetical injects scripted by exercise planners and delivered via paper [35]. Preparation and execution are modest what is concerned to resources. This type of exercise is suitable for an organisation new to exercises, validation of processes or preparation for hybrid and live-fire exercises.

2. **Hybrid** exercise has scripted injects enriched by real probes, scans or email-spoofing [35]. Objectives of this exercise type are determined as an ability to detect, respond, and recover from simulated events. Participants should be familiar with inter-organisation exercises, and they should have strong knowledge of their objectives.

3. The **live-fire** exercise incorporates real scenarios and injects while providing realistic experience and training opportunities [35]. Participants should be familiar with exercises and be confident about their objectives. Live-fire exercise has high demands for resources compare to table-top and hybrid types of exercise.

Scope of this thesis is research on live-fire exercises; thus, problematics of live-fire exercises will be described followingly in detail.

Live-fire exercises are designed to train or assess wide spectre of cybersecurity experts and get an experience of working under stress. Attendees of CSE can learn a lot of technical and decision-making process as well as they can improve their working performance [36, 37].

During cybersecurity exercises, the complexity of the operating environment and consequence relationships must always be kept on organisers mind [37]. In order to provide proper experience, CSE must be as realistic as possible. CSE consist of adversaries (Red Teamers), defenders (Blue Teamers) and other supportive teams - scenario creators, infrastructure maintenance and others [37]. Thanks to the live nature of the exercise, attendees will experience the rapid occurrence of several events which can verify their attention and focus [36]. This gives a unique opportunity to assess how attendees can work under fire and evaluate current strengths and weaknesses [36].Therefore proper training outcome is necessary to consider.

4.1 Cybersecurity exercise phases

Appropriate exercise planners and participants must do exercise planning. Therefore the preparations should start several months before the exercise and typically has several phases shown in Figure (4) [35].

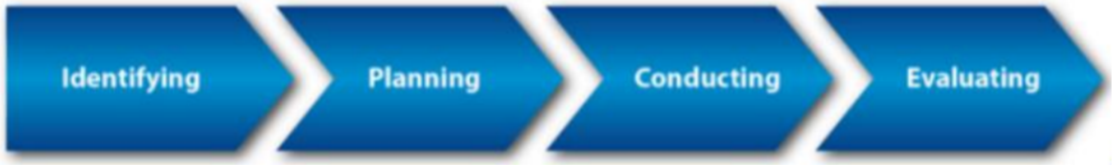


Figure 4. *The lifecycle of Cybersecurity exercise, from[38]*

Identification phase

Identification phase should include recognition and creation of participants profile, determination of type and size of the exercise, evaluation of the current scenario and consider feedback reports from previously organised exercises [38, 37].

Planning phase

Planning of CSE is a complex process where multiple aspects must be taken into consideration. Organisers must plan the budget, schedule, availability of experts,media coverage, infrastructure, logistics, technical resources and desired meaningful outcomes [36, 38]. At this stage, organisers should accurately define the role of subteams and its participants as well as determine which goals should be included in the exercise scenario [36, 38].

Implementation phase

Implementation, also defined as Conducting or as Execution, includes implementation of the exercise, implementation of scenarios and injects according to the determined sequence [36, 38]. Tools for situation awareness help participants and organisers to understand what is happening and whether the scenario should be updated [36]. Reporting participants actions and decision making is vital for further feedback and outcome phase. Many problems and faults may occur in this phase [36]. Therefore organisers must be prepared to respond quickly[38]. Implementation phase usually lasts one to five days [38].

Feedback phase

Feedback phase is considered as the most crucial phase what is concerned about individuals learning [36]. All the main operation lines should be gone through since it allows participants to ask questions and get further explanation of the events that they experienced during the exercise. Conclusion of participant's decisions, responses to incidents helps to understand what happened during an execution phase fully [36, 35]. The feedback session also shows whether the learning objectives were achieved [36]. Based on the experiment in [36], all the different actors of exercises need to participate in the feedback phase and use information gathered in the execution phase.

Exercise planners should write the AAR within the following weeks and markdown lessons learned for the future development of the exercise [35].

Additionally, organisers should also prepare documents for media coverage [38].

4.2 Reporting and analyses

4.2.1 Sensors

Sysmon

Sysmon is Microsoft tool for capturing endpoint data [30]. Once Sysmon is installed at Windows host, it runs as a service which monitors and logs system activity to the Windows event log. Detailed information about process creations, network connections, driver loads, dynamic linking library (DLL) module loads, and changes to file creation time is provided [30, 39]. Sysmon does not provide any analysis of events, and it does not protect or hide from adversaries [39].

Autoruns

Another Microsoft tool that provides data of automatically started scheduled programs Configuring a program within the host [30]. Scheduled programs are typical adversary tactic to achieve persistence. Autoruns Hide Signed Microsoft Entries in order to follow only third-party auto-starting images [40].

Snoopy

Snoopy is a library enabling logging all executed commands and arguments on Linux-based systems [41]. Sample outcome of Snoopy represents Listing 4.2.1.

Listing 4.1. Snoopy output

```
2015-02-11T19:05:10+00:00 labrat-1 snoopy[896]: [uid:0 sid:11679 tty:/dev/pts/2 cwd:/root filename:/usr/bin/cat]: cat
        /etc/fstab.BAK
```

NetFlow

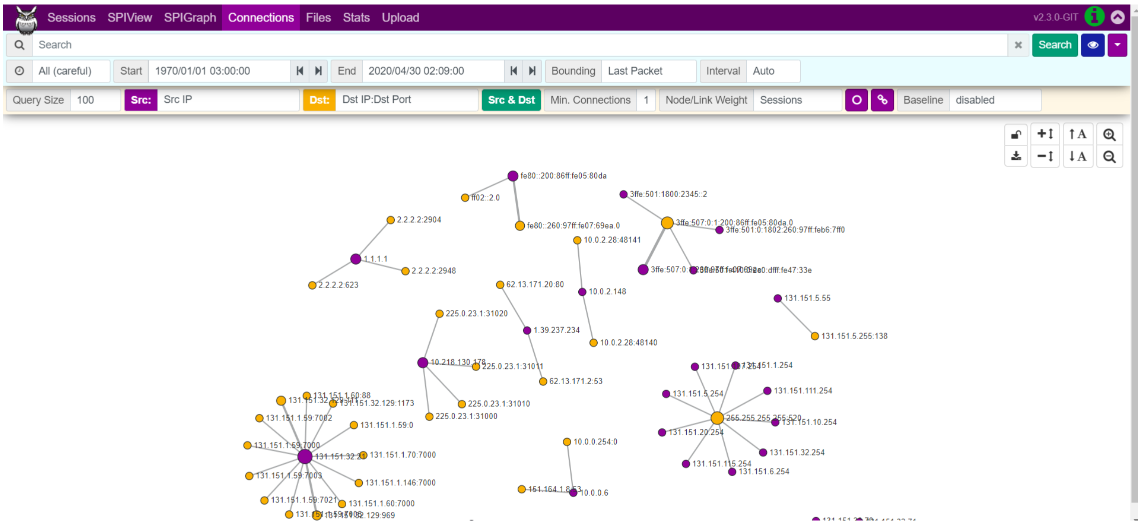
NetFlow record, known as flow records or flows, contains a high-level summary of network connections. Cisco introduced NetFlow, but many vendors are providing NetFlow as a standard nowadays [42]. NetFlow record consists of the start time, end time or duration, source and destination IP, source and destination port, layer 4 protocol (TCP, UDP, or ICMP), Bytes sent, bytes received, and TCP flag [42]. Example of a NetFlow record represents Listing (4.2.1).

Listing 4.2. NetFlow output

```
 Date flow start          Duration Proto   Src IP Addr:Port        Dst IP Addr:Port      Packets    Bytes Flows
2010-09-01 00:00:00.459     0.000 UDP     127.0.0.1:24920    ->  192.168.0.1:22126         1        46     1
```

Full packet capture

The advantage of full packet capture is obtaining original traffic data in PCAP format, which is beneficial for precise further investigation [42]. On the other hand, full packet capture has a high demand for resources [42]. Moloch is as an example of an open-source tool for full packet capturing, indexing, and database system of large scale applications [43]. Preview of connections of sample full packet capture shows Figure (5).



All (careful) Start 1970/01/01 03:00:00 End 2020/04/30 02:09:00 Bounding Last Packet Interval Auto

Query Size 100 Src: Src IP Dst: Dst IP:Dst Port Src & Dst Min. Connections 1 Node/Link Weight Sessions Baseline disabled

Figure 5. *Moloch preview of connections*

4.2.2 Events processing

Simple Event Correlator

SEC is a lightweight, platform-independent solution for event correlation of streamed event processing [44]. Event correlation processes the stream of events in order to detect and act on certain event groups that occur within predefined time windows [44].

IDS/IPS

Intrusion detection system stands for "hardware or software products that gather and analyse information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organisations) and misuse (attacks from within the organisations)."[45]

When an IDS detects suspicious activity, it generates an alert [42]. This alert might be based on a match against one of its signatures or anomaly detection and should contain further details necessary for the SOC analyst [42].

IDS alerts sometimes also include a reference to the raw PCAP which likely helps to find the link between an attacker and target [42]. When the system is also capable of active threat blocking, it is called Intrusion prevention system (IPS) [42]. Sensors of IPS and IDS are typically installed to the network bottlenecks [42]. The sample of IDS architecture represents Figure (6).

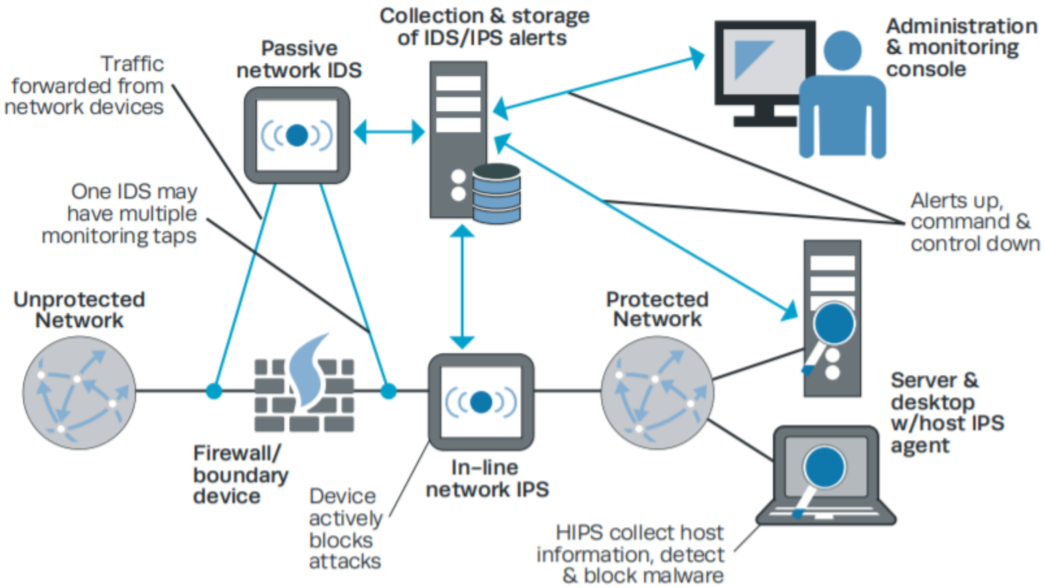

Traffic forwarded from network devices
Passive network IDS
Collection & storage of IDS/IPS alerts
Administration & monitoring console
One IDS may have multiple monitoring taps
Alerts up, command & control down
Unprotected Network
Protected Network
Firewall/ boundary device
Device actively blocks attacks
In-line network IPS
Server & desktop w/host IPS agent
HIPS collect host information, detect & block malware

Figure 6. *Typical IDS Architecture, from [42]*

Suricata is an open-source example of network intrusion detection (NIDS) and prevention (NIPS) tool [46]. It allows protocol analysis, content searching and matching and inspects the network traffic using rules and signature language [46]. Another widely used open-source IDS is Zeek, formerly known as "Bro".

5-Timestamp Methodology

The methodology of five timestamps is another approach for the improvement of learning outcomes. The main scope is evaluation of Blue Team effectiveness assessment [47]. It combines Blue and Red Teams reporting with the collection of timestamps from PCAPs from gamenet and management network [47]. Five timestamps method enables to get valuable detailed output thanks to 5 measurement points. The 5-timestamp methodology shows Figure (7). Precisely, this method provides the following time intervals:

- incident response time,
- time to mitigate,
- the time between mitigation and detection,
- time between compromise and detection,
- time to compromise.

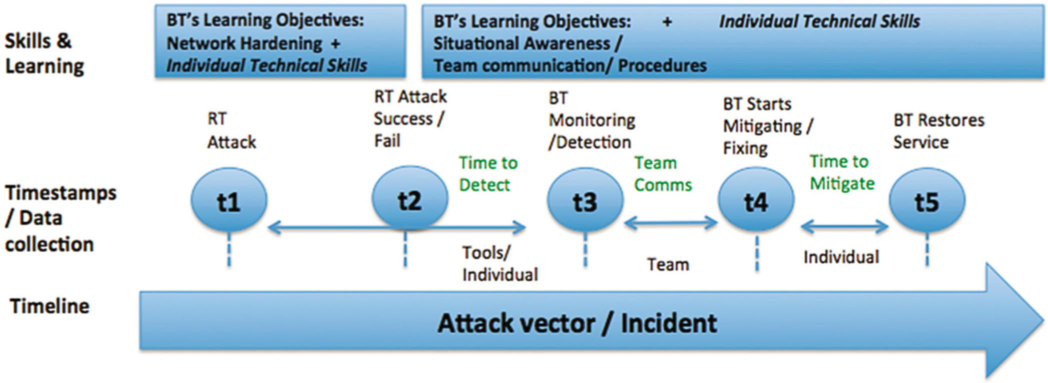

Skills & Learning
BT's Learning Objectives: Network Hardening + Individual Technical Skills
BT's Learning Objectives: + Individual Technical Skills Situational Awareness / Team communication/ Procedures
RT Attack
RT Attack Success / Fail
BT Monitoring /Detection
BT Starts Mitigating / Fixing
BT Restores Service
Timestamps / Data collection
t1
t2
t3
t4
t5
Time to Detect
Team Comms
Time to Mitigate
Tools/ Individual
Team
Individual
Timeline
Attack vector / Incident

Figure 7. *5-timestamp, from [47]*

Feedback based on five timestamps method was mainly assessed as significant in terms of learning outcomes during the test event [47].

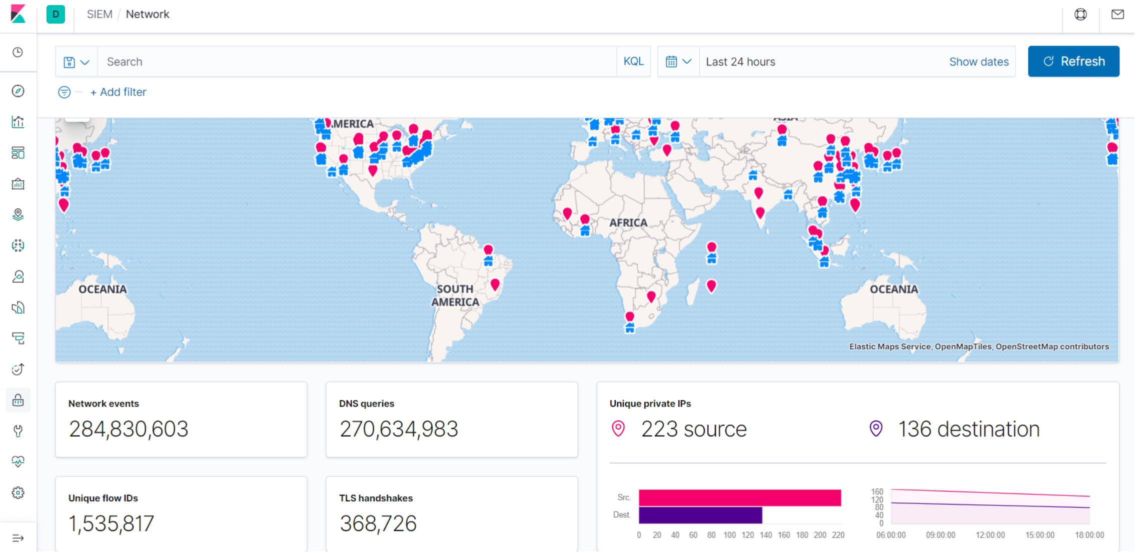
4.2.3 Situation awareness visualisations

Graphical visualisation of situation awareness (SA) is necessary for security management as well as for analyst[46]. Meaningful visualisation allows operators to deal with a large number of inputs while giving a better picture of the complex cyber world; furthermore, situation awareness is essential for making correct decisions and taking appropriate actions

[46, 48].

Kibana

Kibana is open-source browser-based analytics, view and search dashboard based on Elasticsearch [49]. It is equipped with various visualisation sub-tools, such as tables, charts, maps and histograms. Kibana is suitable for operations with a large amount of data stored in JSON format [49]. The preview of Kibana presents Figure 8.



Network events 284,830,603

DNS queries 270,634,983

Unique private IPs 223 source 136 destination

Unique flow IDs 1,535,817

TLS handshakes 368,726

Figure 8. *Demonstraion of Kibana*

Grafana

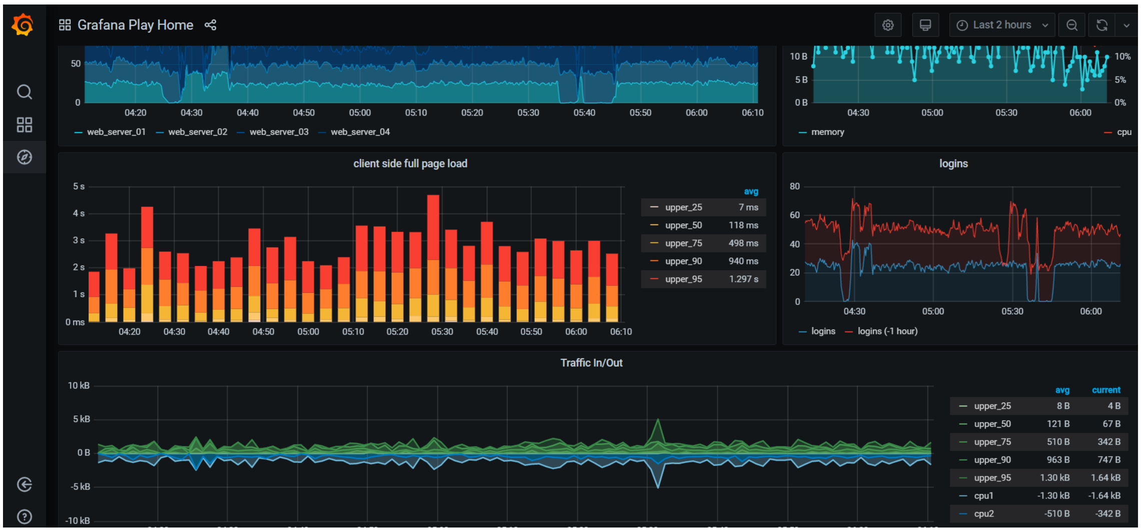
Grafana is another open-source alternative for visualisation. Unlike Kibana, Grafana can operate with diverse data sources without further tools [49].The preview of Grafana presents Figure 9.

Both - Kibana and Grafana allows logs exploration. Kibana does not have implemented alerting, although it is possible to set up an additional open-source solution, such as Watcher, OpenDistro or ElastAlert [50].

Alerta is a simple monitoring tool with the need for minimal configuration for any data source, easily scalable and designed to provide details of events if needed [51].

4.2.4 Log processing

Enterprise system needs an effective solution for pipelining a vast amount of data, such as user activity or operational metrics [52]. The position of log processors is between endpoint and applications for further analysis.



	avg
upper_25	7 ms
upper_50	118 ms
upper_75	498 ms
upper_90	940 ms
upper_95	1.297 s

	avg	current
upper_25	8 B	4 B
upper_50	121 B	67 B
upper_75	510 B	342 B
upper_90	963 B	747 B
upper_95	1.30 kB	1.64 kB
cpu1	-1.30 kB	-1.64 kB
cpu2	-510 B	-342 B

Figure 9. *Demonstraion of Grafana*

Kafka

LinkedIn developed Kafka for processing and real-time usage of a massive amount of log data. Kafka is an open-source solution running on Java, and it is horizontally scalable, fault-tolerant, wicked fast, and runs in production in thousands of companies [52]. High throughput and stable performance are kept even millions of logs at the same time are aggregated [52].This distributed streaming platform allows applications to consume log events in real-time and store streams of records in a fault-tolerant durable way as well as process streams of records as they occur [53]. On top of that, Kafka provides an API similar to a messaging system and allows applications to consume log events in real-time [52].

Peek

Peek is an opensource tool developed at Frankencoding hackathon organised by CCDCOE as a lightweight alternative to LogStash or Rsyslog [54]. It is a central data normalisation engine for Frankenstack, precisely streaming pre-processor and enrichment tool for structured logs. Peek consumes events from Kafka cluster and emits messages enriched of inventory information from targets and additional details, such as directionality of an action or event description based on sigma rules [54, 55]. Additionally, some sensitive information could reveal game network architecture and degrade scenario [54]. Therefore original and processed messages are kept separated on the cluster level. The design of Peek allows replaying of event messages which is beneficial for analysis and research purposes [54]. Event record created by Peek presents Listing 4.2.4.

Listing 4.3. Example of omitted emitting from Peek

```
{
    "@metadata": {
        "beat": "winlogbeat",
        "type": "type",
        "version": "1.0.0"
    },
    "@timestamp": "2020−05−01T05:45:57.418Z",
    "GameMeta": {
        "Host": "host01",
        "Alias": "",
        "is_asset": false,
        "DirectionString": "Local",
        "Directionality": 1,
        "MitreAttack": {
            "ID": "T1036",
            "Name": "T1036:␣Masquerading",
            "Tactics": null,
            "Items": [
                "Masquerading"
            ],
            "Techniques": [
                {
                    "ID": "T1036",
                    "Name": "Masquerading",
                    "Tactics": null
                }
            ]
        },
        "EventData": {
            "ID": 3,
            "Key": "Microsoft−Windows−Sysmon/Operational",
            "Fields": [
                "winlogbeat.exe",
                "Network␣connection␣detected␣(rule:␣NetworkConnect)",
                "SYSTEM"
            ]
        },
    },
    "destination": {
        "ip": "10.0.0.1",
        "port": 80
    },
    "network": {
        "community_id": "",
        "direction": "outbound",
        "transport": "tcp",
        "type": "ipv4"
    },
    "source": {
        "domain": "host01.network",
        "ip": "10.0.0.2",
        "port": 11111
    }

}
```

4.2.5 Log management solutions

ELK

ELK stands for Elasticsearch, Logstash and Kibana. Elasticsearch is a NoSQL database with the implementation of the Lucene search engine [56]. Logstash is a log pipeline system consumes, transforms, and loads data into Elasticsearch [56]. ELK stack is an open-source solution maintained by Elastic company. Later on, Elastic introduced Beats. Beats are data collectors generating unified logs, thus simplifying the load process. ELK stack solution is used by companies such as Netflix, Facebook, Microsoft, LinkedIn, and

Cisco [56]. An example of an Elastic cluster architecture manifests Figure 10.

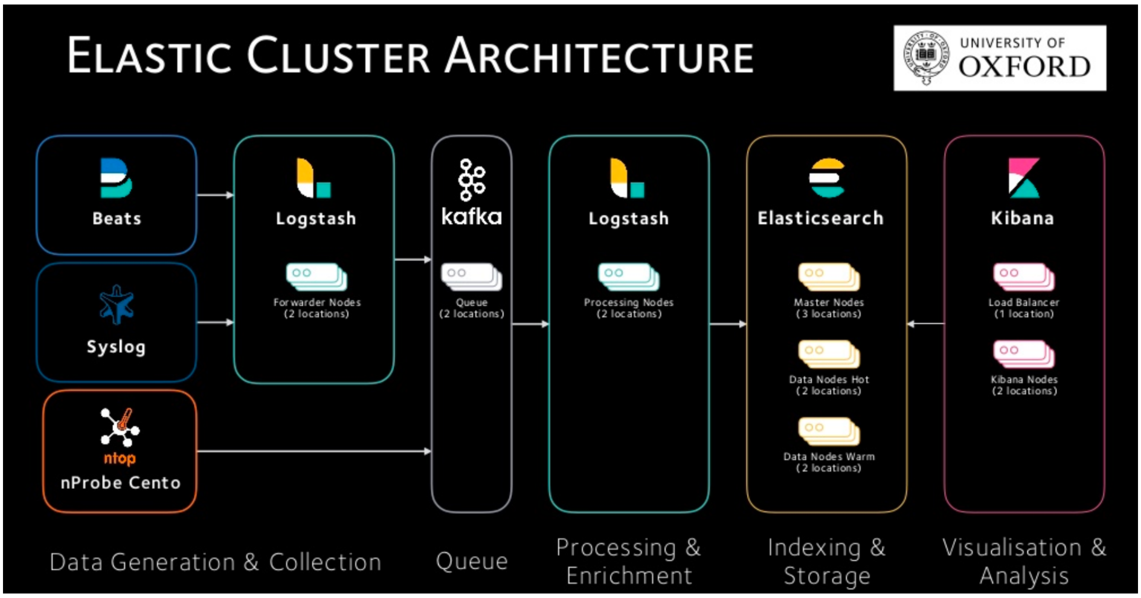


Figure 10. *Example of elastic cluter architecture using Kafka, from [57]*

Splunk

Splunk, know as "Google for logfiles", is a commercial solution developed by Splunk company [58]. As an equivalent of Elastic's Beats, Splunk uses Forwarders [59]. The indexer is processing, collects and serves the incoming data in real-time as well as collects and arranges the data on the disk [59]. For visualisation and analysis purposes is developed the Search Head [58]. More than 1000 applications and add-ons extend Splunk user experience [58]. Typical Splunk architecture displays Figure 11. Splunk might be an expensive solution for some use cases; on the other hand, ELK usually requires additional hardware costs, price of storage and professional services [58].

SIEM

Security information and event management (SIEM) helps to incident response team in reporting and forensic analysis of security incidents as well as alerting. The limitation of SIEM is a lack of context since it depends on the received data, which can lead to chasing false alarms and lead to desensitisation of the security team [61]. Therefore context is critical successful diagnosis and research of security events [61]. Splunk, often advertised as a SIEM, as well as ELK, are frequently used as SIEMs. Another example of SIEM solutions is IBM QRadar and LogRhythm.

Frankenstack

CCDCOE developed Frankenstack as an open-source framework for real-time Red Team feedback at cybersecurity exercises. Usage of commercial tools is often not reliable for

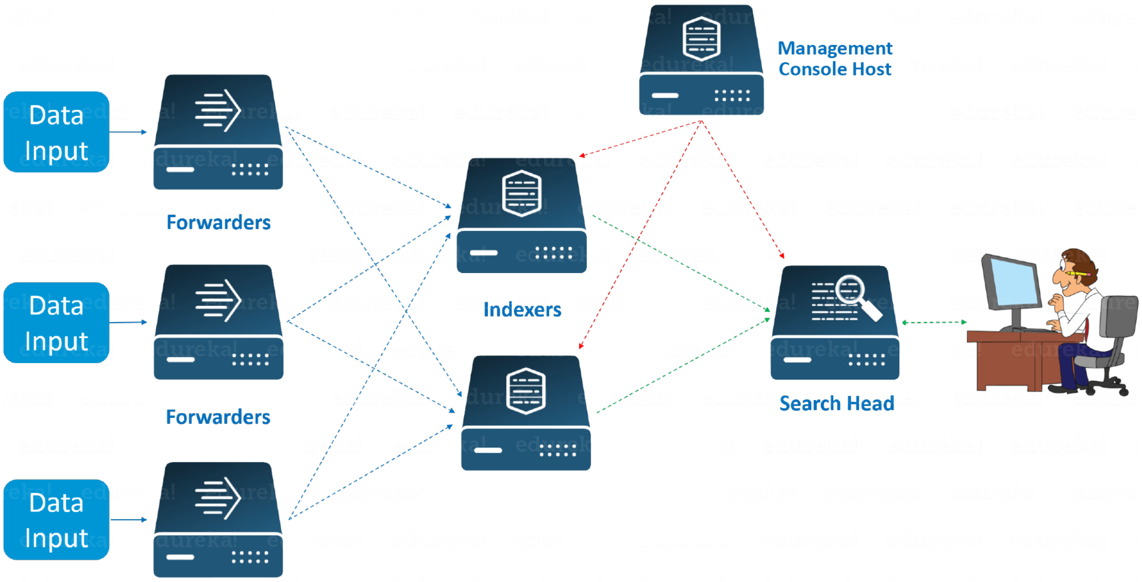


Figure 11. *Example of Splunk architecture, from [60]*

cybersecurity exercises because of license and hardware costs as well as requirements for specialists support from the field [62]. The full description of the detection logic of commercial tools might not be available to users which disallows detailed explanation of detected attack [62]. Frankenstack consists of open-source and self-developed tools and can be easily customised. Usage of Frankenstack met with mainly positive feedback by Crossed swords 2017 participants in the context of learning impact [62]. The Figure 12 represents implementation of Frankenstack at Crossed Swords 2017 cybersecurity exercise.

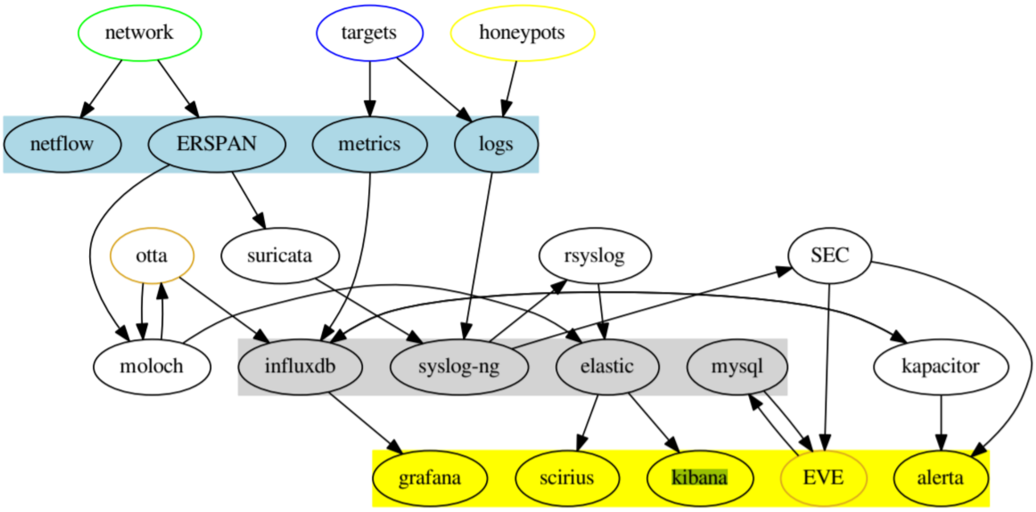


Figure 12. *Data flow between Frankenstack elements during XS17, from [62]*

4.3 Adversary behaviour

4.3.1 Advanced persistent threats (APTs)

Advanced persistent threats are sophisticated cyber attacks by hostile organisations or groups with the following goals:

- gaining access to targeted information from governments, corporations and individuals;
- maintaining a backdoor in order to enable future use and control;
- Modification of data to disrupt performance in their targets [63].

The major problem raises when attacker sneaks undetected into the network even when antivirus, firewalls and other tools were adequately implemented. On average, cybercriminals spend 191 days inside a network before being discovered, which is more than enough time to cause severe damage [64].

4.3.2 Adversary detection indicators

There are several indicators of malicious activity. Adversaries might easily evade some indicators, but it would take much effort to evade more sophisticated indicators. Following diagram (Figure 13) shows the relationship between types of indicators and adversary "pain" caused by evading them [65]. T

Width represents the uniqueness of indicated values and colour represents "pain" for evasion of the current indicator.

4.3.3 Indicators

Hash Values: Hashes provide unique references to specific samples of malware or file. Usually MD5 or SHA1 Hash value indicator is most accurate, easy to indicate and evade by changing a single bit [65].

IP Addresses: Numerical label to each device connected to computer network.

There is a massive number of IP addresses which might be easy to change, for example using ToR [65].

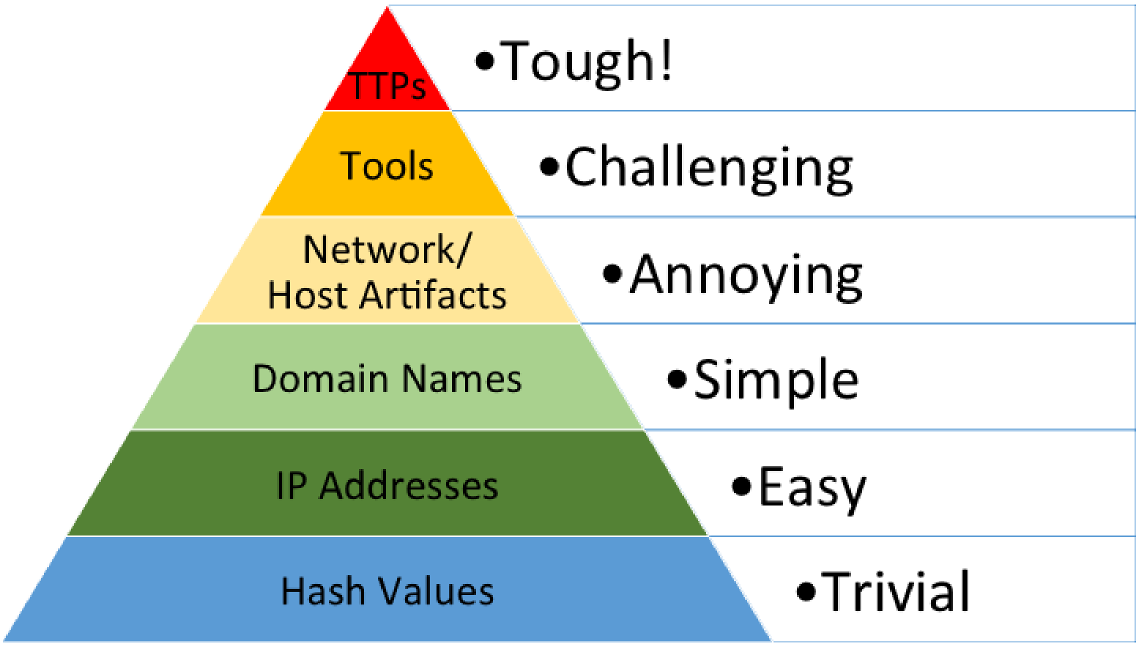


Figure 13. *Pyramid of pain, from [65]*

Domain names: Regular domain or subdomain name.

Domain name indicator is easy to evade by changing to another domain name. The adversary must purchase a new domain, and DNS propagation delay might be not reliable [65].

Network and Host artefacts: URI patterns, HTTP User-Agent or SMTP Mailer values might be valuable network artefacts. Host artefacts can be represented by dropped folders and files or registry manipulation.

As an example of evasion, an adversary must reconfigure or recompile their tool in order to use another User-Agent [65].

Tools: In other words, software used by an adversary such as port scanners or password crackers.

Evasion of tools indicator is a more time-consuming process since an adversary must find another tool or develop one [65].

TTPs APTs are described by tactics, techniques, and procedures (TTPs) for reporting purposes and cyber threat intelligence (CTI).

TTP indicator operates straight again adversary behaviours, not against their tools. It is considered as most effective because adversary must learn new behaviours (most time consuming) if incident responder takes quick actions [65].

4.4 Adversary behaviour at CSE

Understanding of adversary behaviour is an essential skill for system administrators, engineers and SOC operatives [66]. Adversary mindset and up-to-date information about threats improve computer systems security. Cybersecurity exercises allow codifying attacker behaviour. There are different approaches to serving this purpose. Measured data from recordings, Data from observations and Self-reported data can serve as primary sources for adversary characteristics[67].

Authors of the paper [66] propose the method for characterising attackers behaviour according to MITRE ATT&CK leading to the creation of diagrams presenting sequential use of techniques and tactics (see Figure 14). Diagrams are created based on the dataset collected during the 2018 National Collegiate Penetration Testing Competition (CPTC'18) while requiring considerable manual work [66]. The attack narratives represent an approach of competition participants and do not necessarily represent the real adversary approach [66].

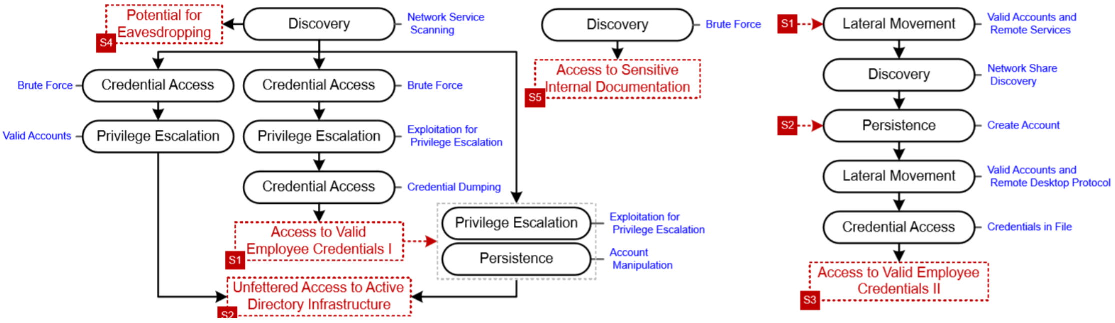


Figure 14. *Diagram of the adversary team behaviour based on MITRE ATT&CK during CPTC'18, from [66]*

Other framework [68] defines adversary narratives from the set of attack signatures extracted from network traffic, although it is not trivial to derive a useful analysis from attack signatures due to dataset complexity and white noise [68]. End-to-end sets of signatures are interpreted as narratives based on Mandiant's Attack Lifecycle Model [68]. Firstly step of this approach is reducing of dataset's white noise. Later on, attack signatures are extracted, which lead to the creation of narratives. Limitation of this framework is tracking non-ciphered protocols only [68]. The sample attack narrative represents Figure 15.

Another approach [69] demonstrates possible profiling of APT from network datasets using

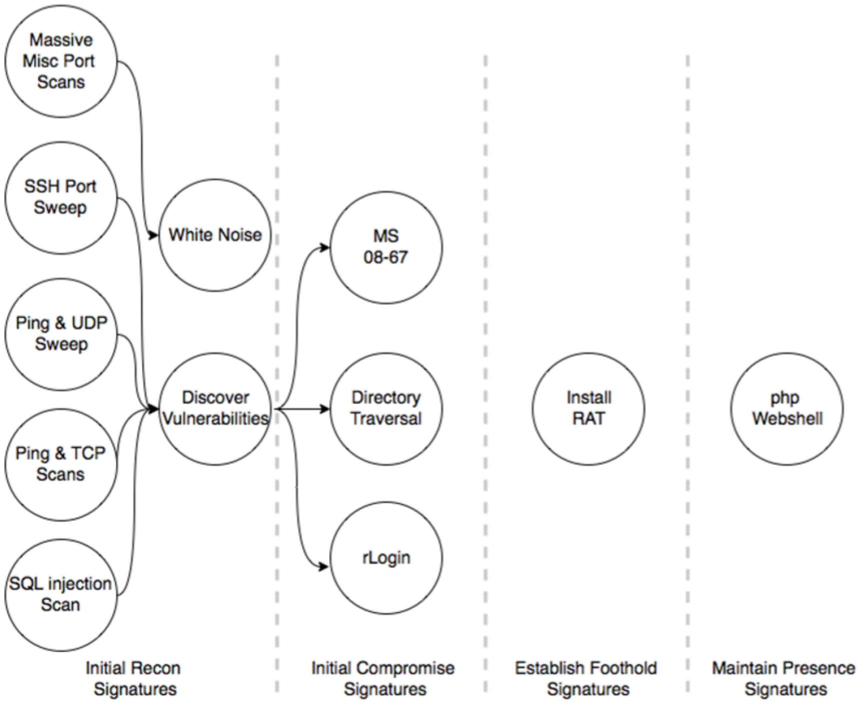


Figure 15. *An example of attack narrative, from [68]*

Zeek. Further research concludes that the most common techniques enable to detect the existence of APTs on a network while the least common techniques may help to specify APT [69].

5. Research and implementation of method for creating reports

This part will propose a method for the creation of the feedback report from cybersecurity exercises using Frankenstack (4.2.5). The report covers recognised techniques performed during the exercise against target domains, visualisations and methods for adversary testing as well as mitigating an impact of used techniques. The process of report creation uses only open-source tools and authors script, which is possible to deploy fully offline in order of data-leakage prevention.

5.1 Action research

The objective of this thesis is an improvement of exercise report based on adversary behaviour represented in ATT&CK for a feedback session at cybersecurity exercises organised by CCDCOE. These steps conducted this action research:

5.1.1 Planning

The planning stage starts by understanding the process of datasets creation of the cybersecurity exercise, in this case, Crossed Swords 2020 (XS20). Access to internal collaboration system for Green Team and private GitLab together with a paper describing data collection allows a deep understanding of this problematics. Documents listed in the collaboration system precisely describes Gamenet zones and their systems. Green team representatives provided datasets and permit to use them for this research.

A brief inspection of the full packet capture and EMIT dataset from XS20 led to the assumption of further steps. Two actions were planned:

1. An attempt to extract ATT&CK techniques from full packet capture (>1 TB of data).
2. Extraction of techniques and targets from EMIT dataset (568.5 MB, 174751 events) for projection to ATT&CK Navigator.

5.1.2 Initial intervention

There are two approaches for extracting data from EMIT logfile:

1. upload of data into ElasticSearch and preview using Kibana and further extraction and
2. an implementation of a python script for parsing techniques into a JSON format (Navigator input).
   Moloch, ElasticSearch and CASCADE establish the setup for the extraction of ATT&CK techniques from the full packet capture.

5.1.3 Evaluation and reflection of the initial intervention

Import of full packet capture into Moloch and further analysis processed by CASCADE throws undefined error respectively gets stuck for an unspecified reason. The process of techniques extraction is terminated at this moment. The upload of EMIT logfile to ElasticSearch is unsuccessful since it is in JSONl format. Another script would solve this issue, but this approach is also terminated at this point.

The first version of initial python script (16) parses loaded data into desired output for Navigator tool but occupies a significant part of RAM. It is possible to import and preview of all used techniques in Navigator at this moment. Since the process is inefficient, and the outcome is very general, a second intervention must take place.

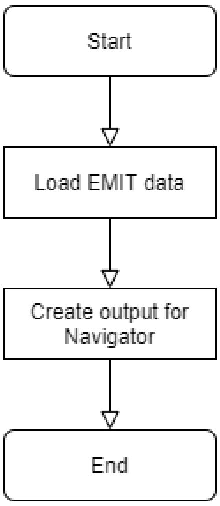


Figure 16. *Flowchart diagram of the script after initial intervention*

5.1.4 Second intervention

According to the reflection of the previous stage, it is necessary to optimise the script by the Pandas library. There is a need for separation of techniques used within domains. Therefore the script requires additional data about hosts. Loading of additional data from CSV files generated from the XS20 collaboration environment and concatenating newly obtained details with data from EMIT logfile solves this issue. Concatenation of these two datasets results in the correlation of IP addresses of all hosts. Events obtained from Linux machines are missing IP addresses; thus, this creates a new option of division by IP address. The newly created data frame allows the creation of output for Navigator divided by domains.

5.1.5 Evaluation and reflection of the second interventions

The second intervention brings the adequate opportunity of data separation according to domains but nothing more. An objective of this research is the feedback enrichment of atomic tests, mitigations of techniques, detections of techniques and data sources for monitoring each technique. Further interventions use the first 100 events of EMIT dataset in order to simplify the research process.

5.1.6 Third intervention

There is a need to add the library of the Atomic test and CTI interface at this stage. Followingly, optimisation of the script for proper usage of newly added inputs from Atomic tests and CTI is performed.

5.1.7 Evaluation and reflection of the third interventions

The script generates an output of techniques separated by domains, their mitigations, detections and data sources necessary for monitoring. The output consists of structured yet hard to read data. In order to satisfy an objective of the research, a user-friendly representation of data should take place.

5.1.8 Fourth intervention

The objective of this stage is the creation of a human-readable report. The output is parsed into MarkDown document format enhanced of the menu of used tactics within a domain. Newly added outputs for the production of graphs allows the actual conception of graphs using RawGraphs tool. Once graphs are created, they must be manually added into the report. Additionally, a scoring formula enriches data output for the Navigator and later on, the matrix is exported to image format and concatenates the final report as well.

5.1.9 Evaluation and reflection of the fourth interventions

Fourth intervention reaches the objective of the research. The ATT&CK-based report consisting of reading instructions, graphs, the menu of used techniques and techniques with their description, atomic tests, mitigations, detections and data sources for monitoring is created. Another intervention is not necessary, and action research terminates at this stage. The final version of the script represents the flowchart diagram 17.

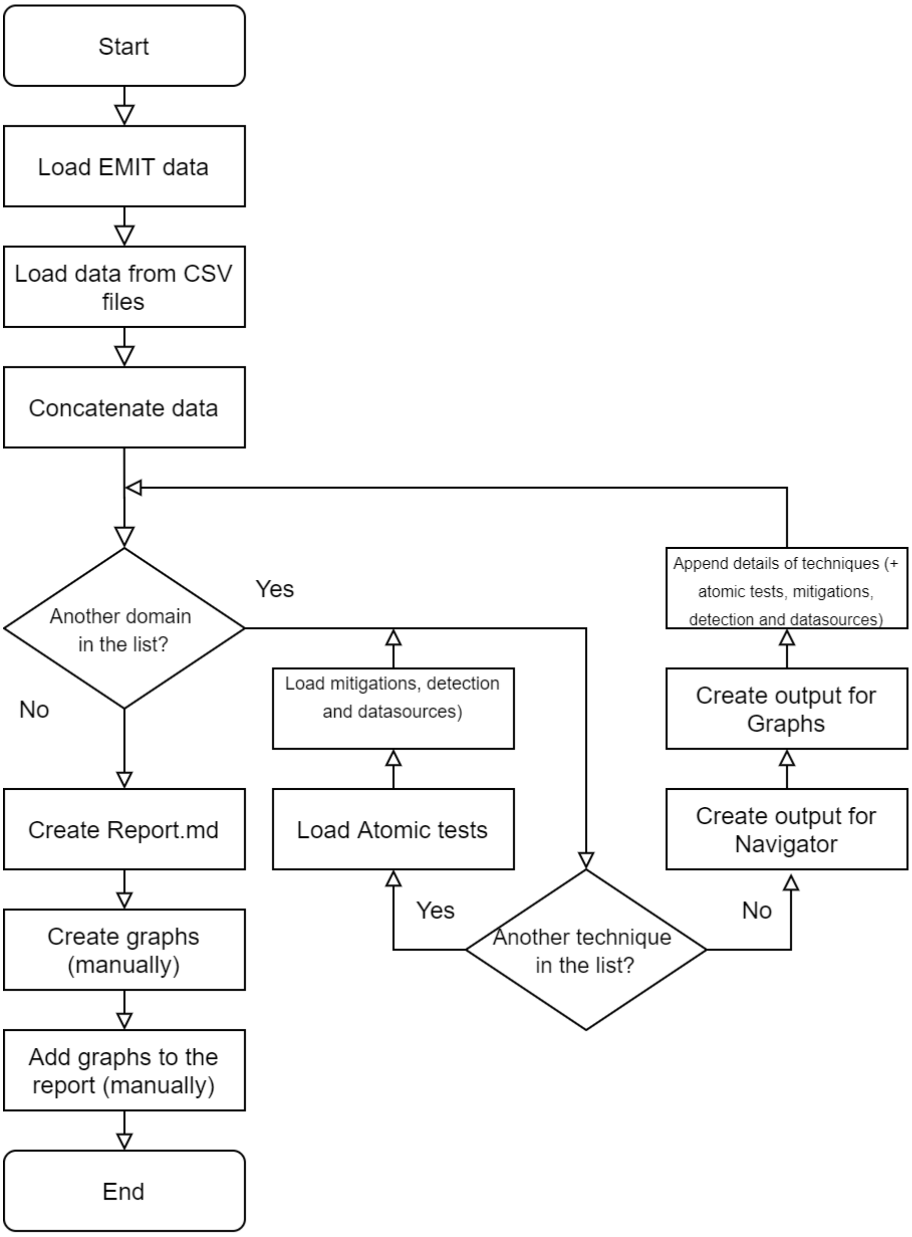

Start
Load EMIT data
Load data from CSV files
Concatenate data
Another domain in the list?
Yes
No
Append details of techniques (+ atomic tests, mitigations, detection and datasources)
Load mitigations, detection and datasources)
Create output for Graphs
Create Report.md
Load Atomic tests
Create output for Navigator
Yes
No
Another technique in the list?
Create graphs (manually)
Add graphs to the report (manually)
End

Figure 17. *Flowchart diagram of the report creation after fourth intervention*

5.2 Implementation

5.2.1 Implementation of report creation method

In this section, I am going to demonstrate the usage of the proposed method on data gathered during Crossed Swords 2020 cybersecurity exercise organised by NATO CCDCOE.

Action research leads to the statement that it is not sufficient to work with full capture dataset. The loading of full PCAPs larger than 1 TB to Moloch takes over 70 hours and secondly because it is not possible to run an analysis via CALDERA. Additionally, uploading to ElasticSearch and Kibana throws errors since the file format is JSONl and ElasticSearch requires JSON file format. Therefore implementation consists of a method using a script created based on action research.

5.2.2 Dataset

Proposed method operates with EMIT, the logfile formatted as JSON Lines and size approximately 500 MB. Omitted structure of EMIT is stated in 4.2.4. Furthermore, the script requires input files describing used systems within the exercise. It must consist of hostname and IP address; other parameters are not necessary at the moment.

5.2.3 The script

Although there is a python script for the report creation process, it still requires additional user's interaction described below. A user has to ensure dataset quality and has to create graphs manually based on output data of the script.

Installation prerequisites:

For correct usage, the user must manually clone two open-source projects:

1. Atomic Red Team 3.2.1,
2. MITRE CTI 3.2.1.

The recommended project structure:

```
/
├─ cti
├─ atomic-red-team
├─ Peek2Report.py
├─ input
│  ├─ systems
│  │  └─ [domain_x].csv
│  └─ emit.log
└─ output
   ├─ navigator
   │  └─ [domain_x].json
   ├─ graphs-data
   │  ├─ mitigations.csv
   │  └─ [domain_x]_timeline.csv
   └─ REPORT.MD
```

GitHub repository (`https://github.com/cmelakmartin/Peek2Report`) provides an updated installation guide and usage guide.

The script description

The python script consumes 2 types of input data - EMIT logfile and CSV files with information about systems in each domain. Files describing systems are consisting of the following information in this implementation:

1. Hostname,
2. operation system,
3. IPv4,
4. IPv6 and
5. description.

The script recognises each line as a unique host; therefore, the user must reshape the file structure in case hosts are summarised. User must keep on mind, that name of domain's CSV file will be used for further analysis and purposes and therefore should have a meaningful title (for example factory.csv). EMIT logfile loads as a next step.

The diversity of endpoint sensors results in missing information in multiple cases. Also, a notable number of events have the localhost IP address (listed as the empty string), which leads to an unclear representation of the actual event. Thus, the script replaces the IP of localhost with local IP address according to the description in domain-describing CSV files. The Pandas library enables effective implementation of operations with data.

Once the script reshapes and updates obtained data using Pandas library and additional functions, it provides essential outcomes. It generates a **JSON file as a data source for Navigator** (3.2.1) tool consisting of technique marker and score.

The formula for scoring within a domain:

$$(occurrence\_in\_domain/total\_in\_domain) * 100 = score$$

Scoring abates further operations with Navigator. First of all, it is possible to sort techniques based on score. Secondly, colour-based marking enables user to understand the occurrence or seriousness of used techniques. It is possible to use an online instance of Navigator or run local offline one.

Furthermore, the script communicates with CTI (3.2.1), which provides available mitigations, detection and data sources for monitoring of used techniques. CTI gives the possibility to render a list of threat groups using the current technique as well as known software for performing the technique. This data are significantly useful as a source for the graphs deliberated to enhanced after exercise analysis.

When the script successfully finishes, it generates the report file in MarkDown format consisting of the menu of used techniques and techniques details (description, atomic tests, mitigations, detections and data sources for monitoring). The script also generates data for ATT&CK matrix coverages and data for graphs creation. There are two types of files - the first consists of data for graphs showing mitigations and data sources for monitoring, and the second type supports the creation of timeline graphs.

The whole script is saved on GitHub: `https://github.com/cmelakmartin/Peek2Report`

When the script successfully finishes, it generates the report file in MarkDown format consisting of the menu of used techniques and techniques details (description, atomic tests, mitigations, detections and data sources for monitoring). There are three types of files - the first consists of data for graphs or diagrams showing mitigations and data sources for monitoring, and the second type supports the creation of timeline graphs. As a third type of output file, the script generates data for ATT&CK matrix coverages and data for graphs creation.

Graphs and diagrams should be created using the open-source tool named RawGraphs. An online instance is available as well as offline deployment. RawGraphs can process CSV files, JSON files and Microsoft Excel data formats as well as it can communicate through API. On the other hand, the manual generation of graphs is feasible since it provides more straightforward customisation and real-time preview. RawGraphs offers several types of graphs for diverse purposes.

The first type is a cluster diagram which shows the hierarchy of mitigations and their technique separated.

The second type of graph, inspired by [70], describes the relationship between used techniques, total usage of each technique and mutuality of mitigations. The idea of alluvial diagram proffers an advantage to look at performed attacks from - Red Team and Blue team perspective.

The third type of graph, named as Sunburst, offers a weighted-hierarchy perspective of the technique-mitigation relationship.

The last type of graph manifests the occurrence of techniques in time scale, in other words, time chunks. Utilised graph for "timeline" is termed GanttChart.

In order to create ATT&CK matrix, the user must upload an output JSON file to Navigator tool, sort techniques if necessary, and create bitmap or SVG file.

Additionally, a user must add graphs, diagrams and ATT&CK matrix manually. The final report shall consist of four sections:

1. Introduction and reading instructions
2. ATT&CK matrix, graphs and diagrams
3. The menu of used techniques.
4. A detailed description of used techniques, associated atomic tests, mitigations and data sources for monitoring in case they are present.

An example of the final report shows Appendix 6.1.

5.2.4 Tests and results

There were two tests of the created method demonstrating the following measurements:

- UniqueTechniques at all domains,
- UniqueTechniques at Domain 1,
- The period of the dataset,
- Total number of Events in Domain 1.

Both tests were run on two machines. The first machine (Machine 1) was TalTech university-owned development server equipped by 16 logical CPUs and 64 GB RAM with Ubuntu 18.04.4 LTS operation system. The specifications of machine 2, a personal computer, were 8 logical CPUs, 16 GB RAM running on Kali Linux 2019.4.

Test 1

The first test was deployed with EMIT log file shorten to first 100 events. The script worked without any problems while running the test since the development of the script insisted of hundred events dataset.

Test 2

The second test operated with the original full dataset. Initially, the script threw an error of wrong input technique. The additional investigation provided findings of inconsistent titling of techniques (for example, "T1053: Scheduled Task", "t1053: Scheduled Task" and "1053: Scheduled Task"). Some techniques had corrupted format ("T", "NaN" and an empty string). Therefore additional input sanitation was performed.

5.2.5 Tests outcomes

Results of both tests are shown in Table 1 (5.2.5). Both tests used the full capacity of one logical CPU Test 2, operating with full dataset needed approximately 8 GB of memory temporarily.

Table 1. Comparison of dataset outcomes

Test	**Unique Techniques (all domains)**	**Unique Techniques (Domain 1)**	**Time Span [dd:hh:mm:ss]**	**Events in Domain 1**
100 events	10	7	00:00:17:21	87
all events	59	40	05:23:48:13	116992

Time comparison of script runtime shows Table 2 (5.2.5). Ultimately, there was common bottleneck transpiring in both tests - communication with CTI over Stix2 server. Every interaction (gathering mitigations, data sources for monitoring and detections of techniques) took approximately 8 seconds. Also, atomic tests were not present in several cases due to ongoing development of Atomic Red Team.

Table 2. Comparison of script runtime on two different machines

Test	**Machine 1 runtime [s]**	**Machine 2 runtime [s]**
100 events	70.95	84
all events	574	697.7

Graphs and diagrams

Graphs of Domain 1 generated from the small dataset (Test 1) consisted of seven unique techniques and 87 event hits; hence, they were subjectively easy to follow. The outcome of Test 2 on Domain 1 had about 40 unique techniques and more than a hundred thousand event hits which made graphs untidy and hard-to-follow. It was not possible to create a timeline graph for test two using RawGraphs online tool. The input data had 5.9 MB, and the site stopped responding in every attempt.

The first graph shows the absolute occurrence of each ATT&CK technique, followed by the technique's name and its mitigations. Source data for the graph (Figure 18) are the first one hundred events in EMIT logfile while showing 7 techniques. The graph (Figure 22) generated on full dataset shows 40 techniques.

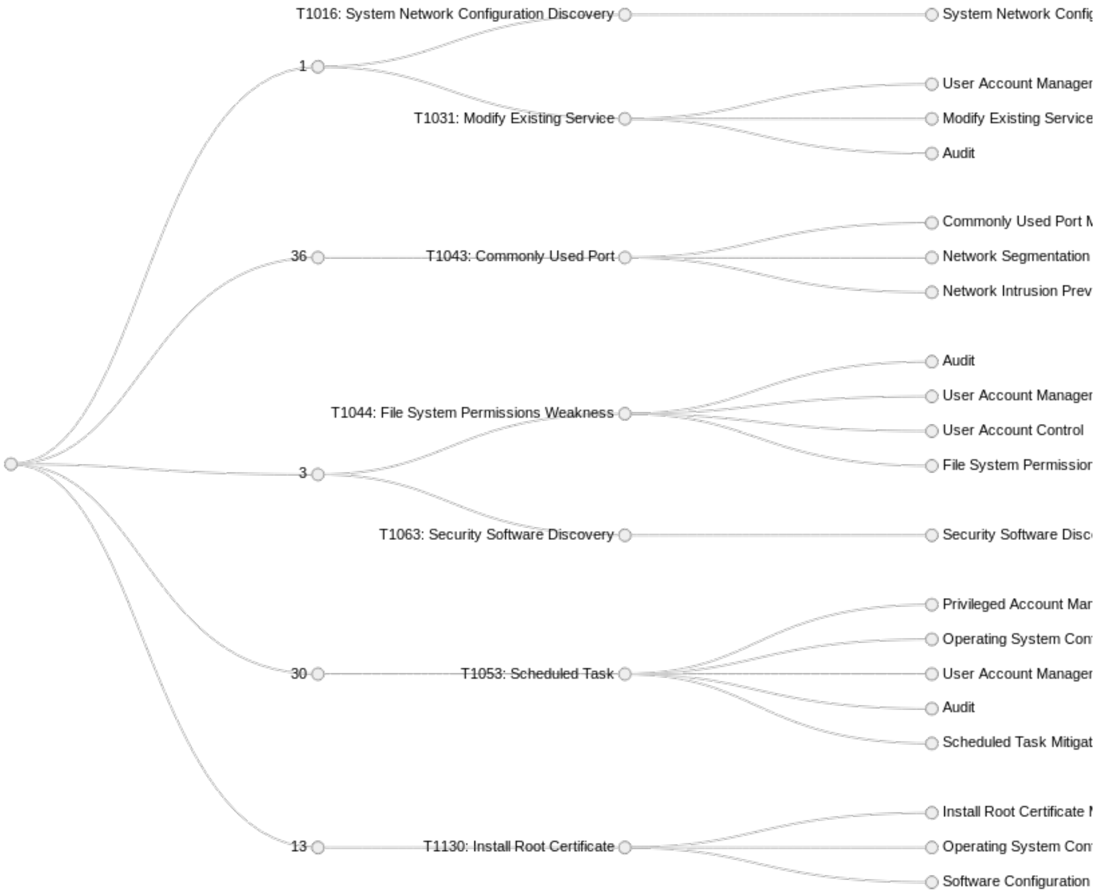


Figure 18. *Test 1: Cluster graph of first 100 events in EMIT logfile*

Alluvial diagram (Figure 19) based on first one hundred events of EMIT logfile and full dataset (Figure 23) shows used adversary techniques in the middle column, connecting with the left column showing possible mitigations and data sources for technique detection at the right column. The size of techniques means the relative occurrence of each technique (score, 5.2.3). In other words, Red Team used the technique T1053: Scheduled Task in the majority of cases within the first one hundred events against Domain 1. Accordingly, the size of mitigations and data sources represents significance based on technique occurrence.

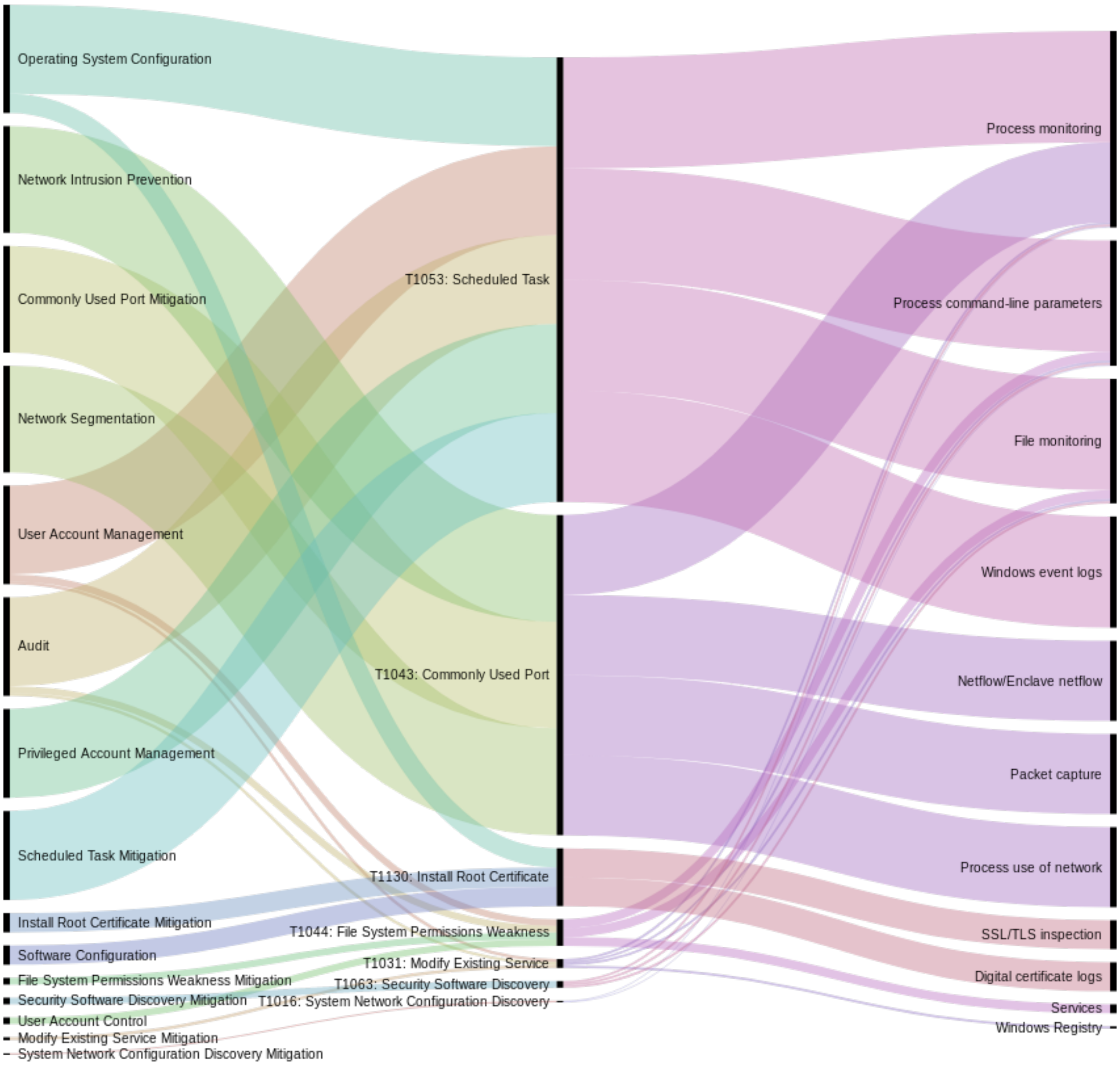

Operating System Configuration
Network Intrusion Prevention
Commonly Used Port Mitigation
Network Segmentation
User Account Management
Audit
Privileged Account Management
Scheduled Task Mitigation
Install Root Certificate Mitigation
Software Configuration
File System Permissions Weakness Mitigation
Security Software Discovery Mitigation
User Account Control
Modify Existing Service Mitigation
System Network Configuration Discovery Mitigation
T1053: Scheduled Task
T1043: Commonly Used Port
T1130: Install Root Certificate
T1044: File System Permissions Weakness
T1031: Modify Existing Service
T1063: Security Software Discovery
T1016: System Network Configuration Discovery
Process monitoring
Process command-line parameters
File monitoring
Windows event logs
Netflow/Enclave netflow
Packet capture
Process use of network
SSL/TLS inspection
Digital certificate logs
Services
Windows Registry

Figure 19. *Test 1: Alluvial diagram mitigation-technique-datasource of first 100 events in EMIT logfile in Domain 1, score based*

Sunburst diagram shows the proposed mitigations of techniques within the inner circle and techniques which are mitigated by these mitigations. The size of the inner blocks represents the total amount of proposed mitigation. Respectively, it shows which mitigation should be established by the Blue Team at first. Outer blocks present used techniques associated with the mitigation. In case of one mitigation might be used for multiple techniques, the size of blocks is calculated by the occurrence ratio of these techniques. Blocks have a different colour in order to improve readability. The graph (Figure 20) shows the outcome generated from one hundred events long EMIT logfile, while graph (Figure 24) represents the outcome of the full dataset.

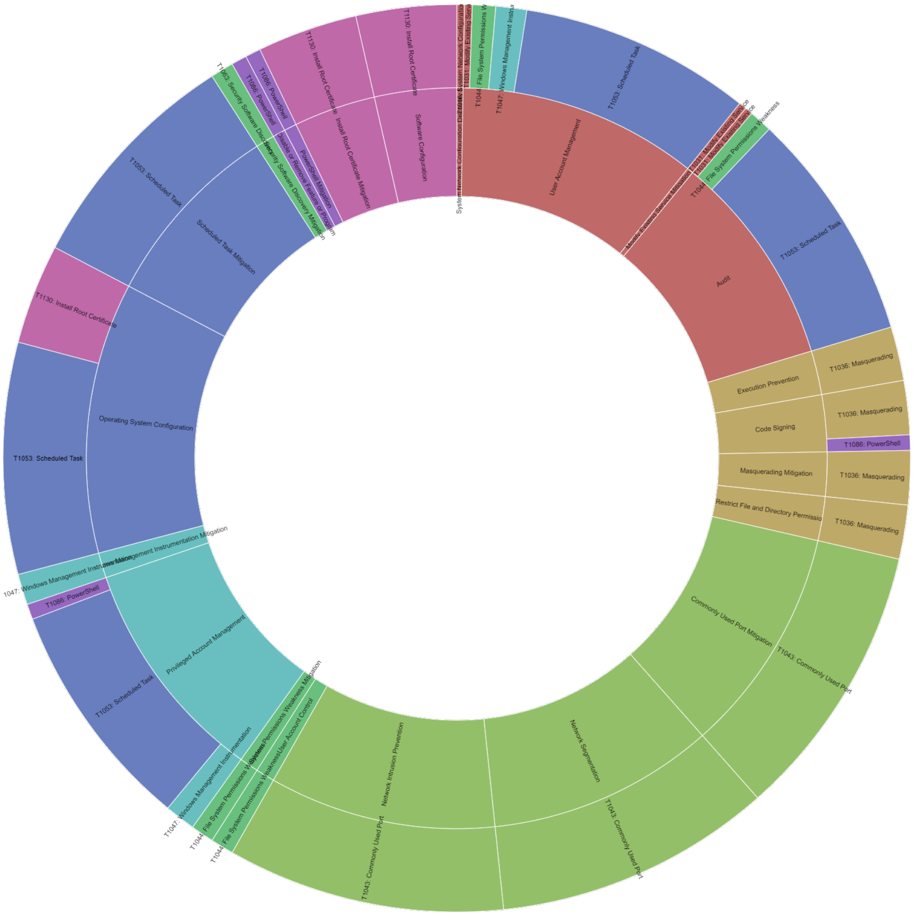


Figure 20. *Test 1: Sunburst diagram of first 100 events in EMIT logfile in Domain 1, score based*

Timeline graph shows the occurrence of techniques in time. Techniques are sorted from top to bottom by the first occurrence of the technique. The graph for one hundred events is presented in Figure 21, while the graph for the whole dataset is not present due to the need for a large format.

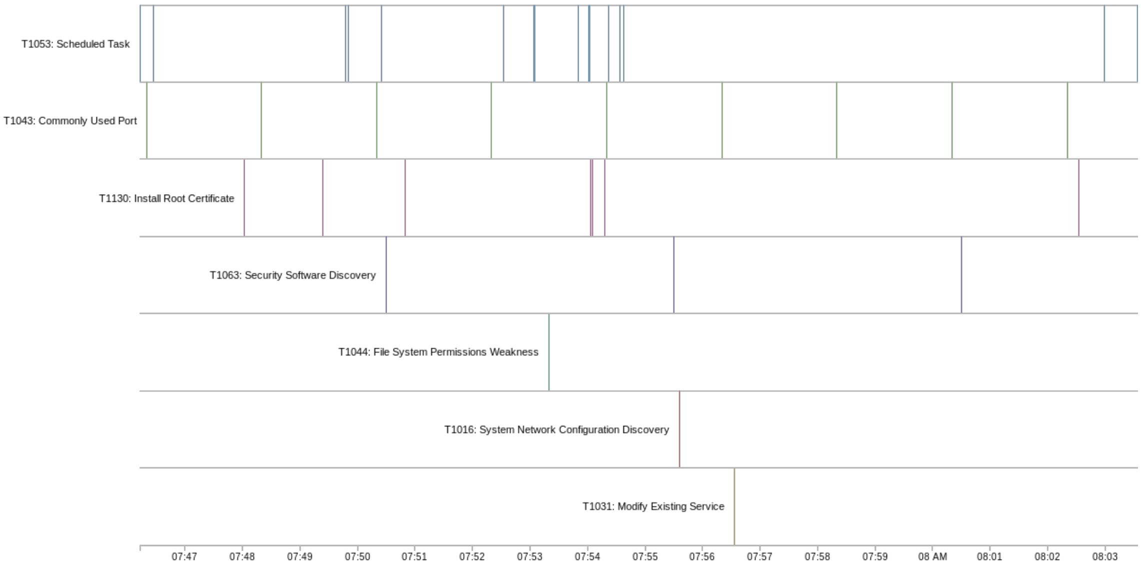


Figure 21. *Test 1: Timeline diagram within one zone of first 100 events in EMIT logfile in Domain 1*

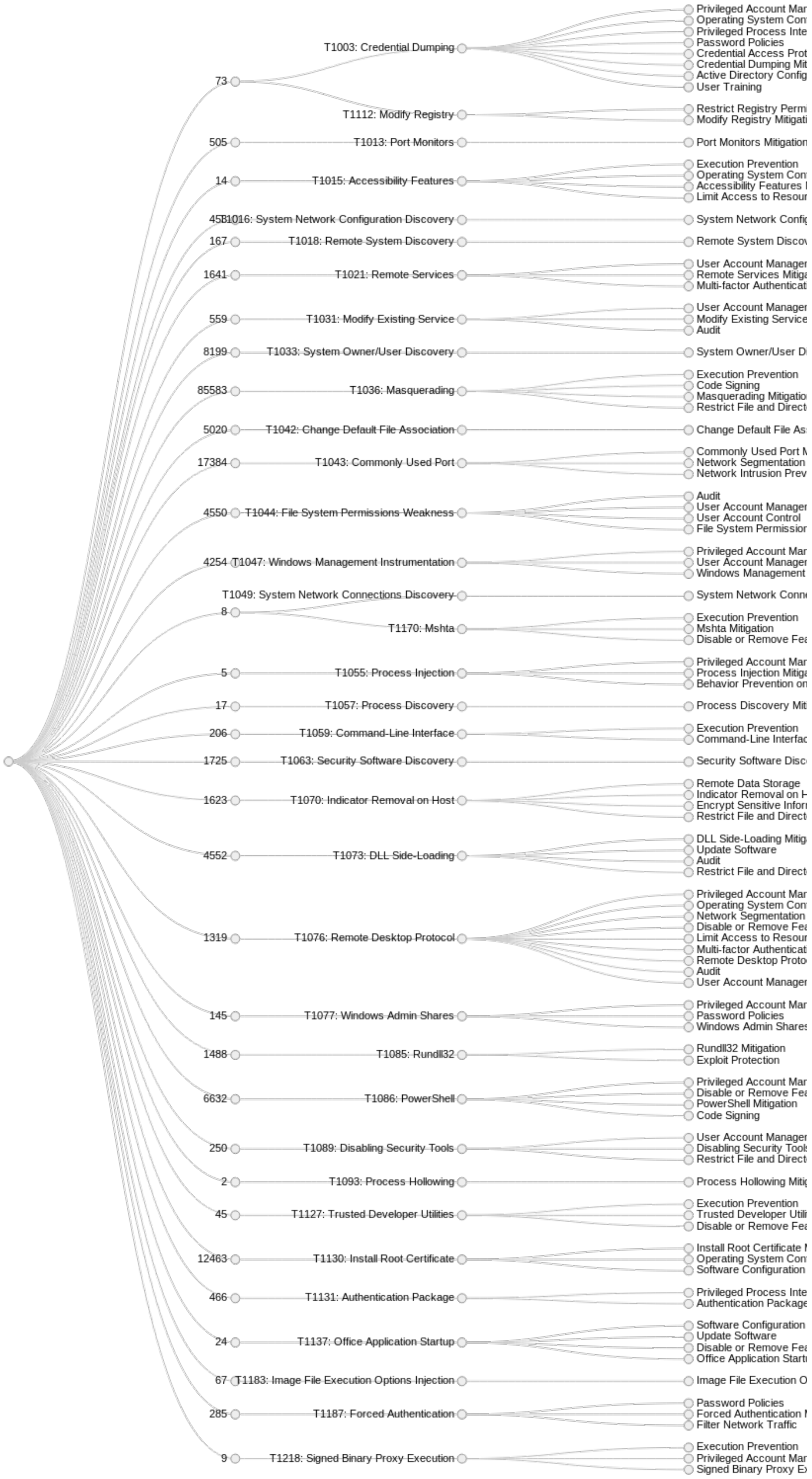


Figure 22. *Test 2:Cluster graph of full EMIT logfile in Domain 1*

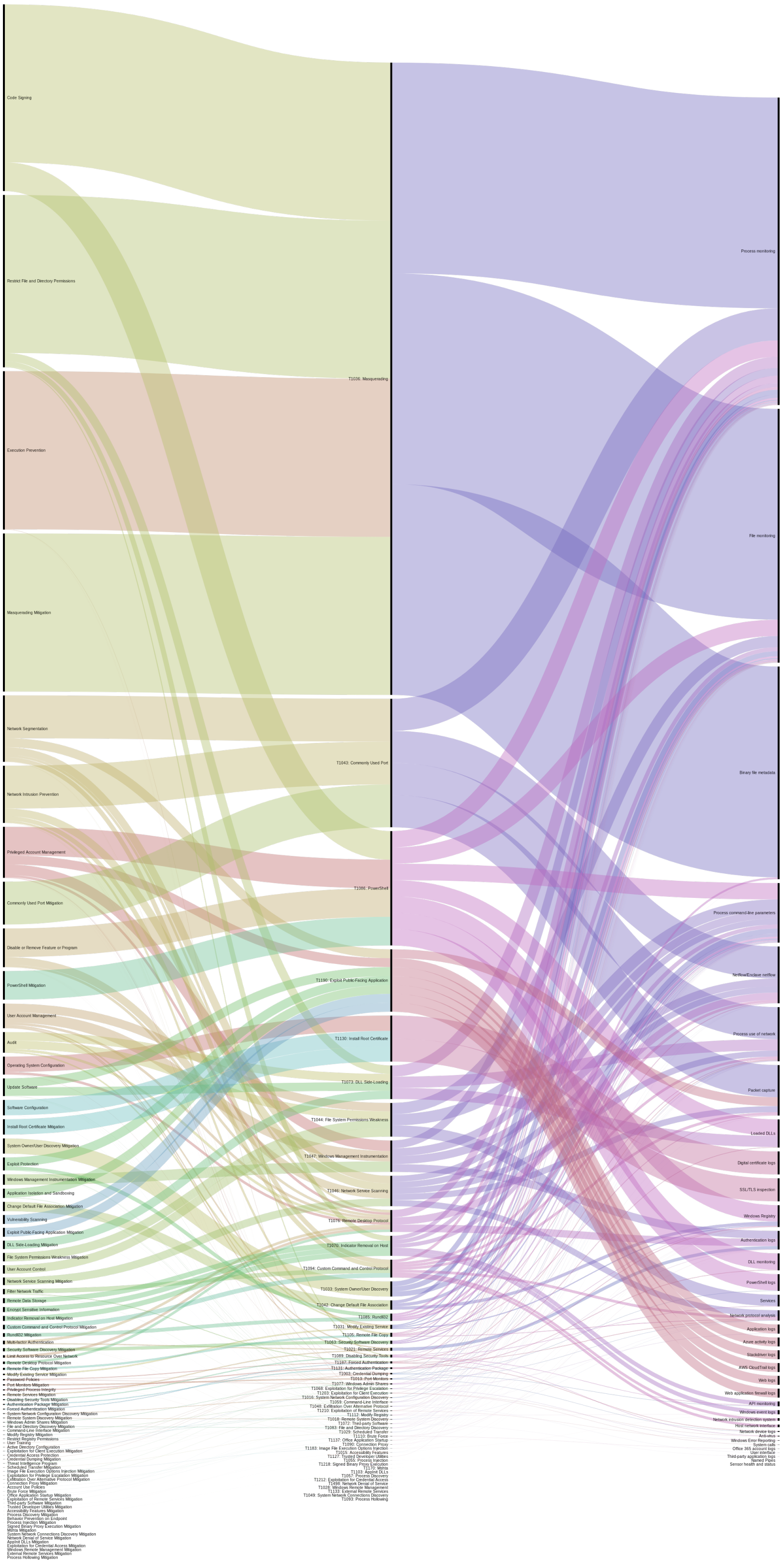


Figure 23. *Test 2: Alluvial diagram mitigation-technique-datasource of full EMIT logfile in Domain 1, score based*

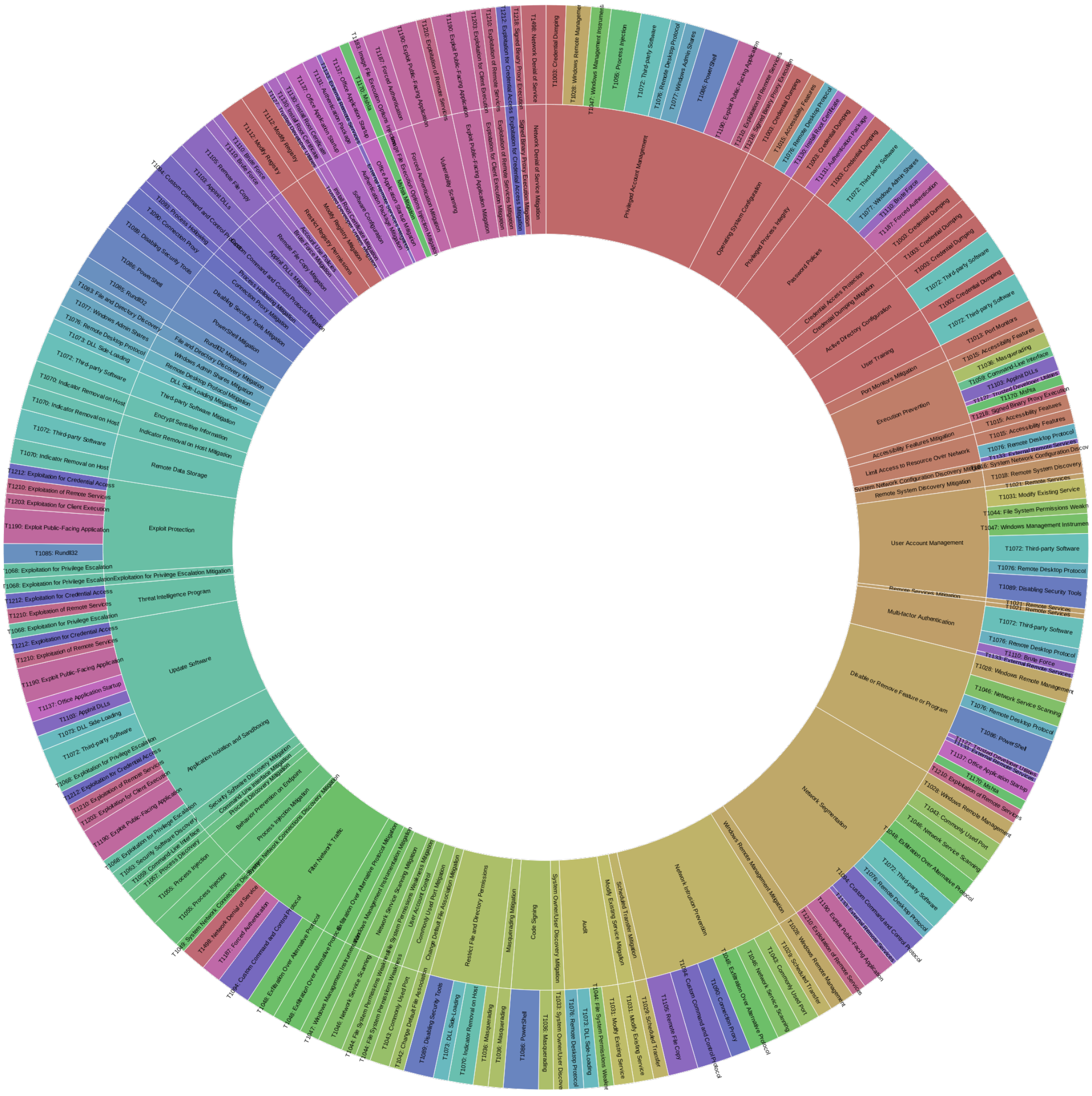


Figure 24. *Test 2: Sunburst diagram of full EMIT logfile in Domain 1, score based*

5.3 Implications

The action research demonstrates the process of method creation for creating reports based on adversary actions during Crossed Swords 2020 EMIT log file generated by Peek (part of Frankenstack) and CSV files describing topology served as an input.

Proposed method produced by interventions, evaluations and reflection is tested on a dataset consisting of the first one hundred events and full dataset. The results of test operating with short dataset provide a subjectively meaningful and tidy outcome. On the other hand, results of test operating with the full dataset were too broad and harder to follow.

In order to answer the question of the problem statement, one can find out that both datasets provide an overview of techniques used by the Red Team and techniques used in time. Furthermore, the final report presents Red Team progress using ATT&CK matrix, timeline graph, cluster diagram, sunburst diagram, alluvial diagram and techniques description. Precisely, what techniques are used in which time against target domain (Domain 1) and techniques occurrence.

Therefore participants, especially Red and Blue Teamers, of XS20 or other cybersecurity exercises may benefit from a deeper understanding of used ATT&CK techniques, their mitigations, detections, data sources and atomic tests. Ideally, improvement of this method should be based on participants feedback on the results of the proposed method - for instance, whether the report shall be created for the timespan of each attacker's phase, targeted point of view and whether the visualisation tools are understandable and lastly if scoring of techniques does make sense.

What is concerned with practical experience of using the proposed method, some manual work is requested for report creation. The creation of graphs might be technically more automatic, but feedback report creator can benefit from manual creation by the author's point of view. Also, the future report creator must observe whether there are any errors in input files. Errors found in given input files were caused by typos in sigma rules and CSV files describing systems. There were two bottlenecks in python script - the script used only one logical CPU, and the communication with CTI through Stix2 server suffered significant response time for an unknown reason.

An example of a report derived from the short dataset is listed in the Appendix 1 (6.1). This example shows events in Domain 1 and has only one technique shown for lucid purposes.

6. Summary

First three chapters conclude literature review and related work. Chapter "Cyber threat modelling" presents an overview of existing methods and frameworks on a low and high level of abstraction. The second chapter describes the philosophy and design of MITRE ATT&CK, its use cases and related projects. A chapter devoted to cybersecurity exercises describes the chief differences of CSE types and organisation problematics. It lists and describes commonly used tools for the whole process from sensors, through event processing, situation awareness visualisation, log processing and log management tools. Moreover, it provides an overview of adversary behaviour and different approaches for characterising adversary (Red Team) behaviour at cybersecurity exercises.

The aim of the fourth chapter is the creation of a method for generating adversary behaviour methods based on MITRE ATT&CK. Action research consisting of planning, four interventions, their evaluations and reflections supports this objective. The proposed method is applied on a short version of the dataset as well as full dataset obtained from Green Team representatives of Crossed Swords 2020. The application results in the following findings - it is possible to generate report consisting of used ATT&CK techniques, their atomic tests, mitigations, detections and data sources for monitoring. Red, Blue, White Teamers can get a more profound understanding of attackers actions taken during the exercise. Although python script generates a significant part of the report, manual work needs to be done. There are two test cases - shorten dataset and full dataset. Both tests led to the creation of a report consisting of graphs (timeline, mitigation-technique-data source relationship) and a detailed description of each recognised technique. Furthermore, tests show two bottlenecks - the python script was using only one CPU and communication with CTI through STIX server caused a significant delay for yet unknown reason. An omitted example (Domain 1, detailed description of 1 technique) of report derived from shortening dataset conjoined in the Appendix 1 (6.1).

6.1 Future work

Dozen of ideas appeared while working on this thesis what is concerned future work. Firstly, participants shall provide feedback on whether the report is suitable for them or they need the report constructed from another point of view - according to a target machine or all Red Team campaigns or other preferred paradigms. Unfortunately, the proper validation based on participants experience was not possible, since the initial plan was testing the report creation method at Locked Shields, another cybersecurity exercise organised by CCDCOE. Locked Shields must have been cancelled because of COVID-19 pandemic. I was testing this method on data gathered during Crossed Swords 2020 as a backup plan. The report in this thesis describes Red Team operations against one domain.

Another proposal of future work is diverse outputs for other visualisation tools. ATT&CK is becoming more and more popular, which leads to the creation of new tools. For instance, the playbook viewer created by Unit 42, threat intelligence team of Palo Alto Networks, or interactive timeline graph (for example, `https://www.amcharts.com/demos/serpentine-timeline/`) might improve participants feedback experience. Comparison of Red Team progress with existing threat groups or score formula based on the severity of the used technique can be implemented if desired.

Lastly, there is still a gap in events correlations and validation. Therefore it might be beneficial to compare the report with internal communication of Red Team (manually and using TRAM (3.2.1)) and compare to results from 5-Timestamps method (4.2.2). Ultimately, a combination of the report created based on the method in theses thesis, and 5-Timestamp method might bring new feedback opportunities.

Bibliography

[1] Steve Mansfield-Devine. “The best form of defence–the benefits of red teaming”. In: *Computer Fraud & Security* 2018.10 (2018), pp. 8–12.

[2] Kirk Hayes. *Penetration Test vs. Red Team Assessment: The Age Old Debate of Pirates vs. Ninjas Continues*. Aug. 2017. URL: https://blog.rapid7.com/2016/06/23/penetration-testing-vs-red-teaming-the-age-old-debate-of-pirates-vs-ninja-continues/.

[3] Blake E Strom et al. “Mitre att&ck: Design and philosophy”. In: *MITRE Product MP* (2018), pp. 18–0944.

[4] D Bodeau, C McCollum, and D Fox. “Cyber threat modeling: survey, assessment, and representative framework”. In: *HSSEDI, The Mitre Corporation* (2018).

[5] Nicole.keller@nist.gov. *New to Framework*. Nov. 2019. URL: https://www.nist.gov/cyberframework/new-framework.

[6] *The CBEST Framework - Cyber Threat Intelligence*. URL: https://www.secalliance.com/services/cyber-threat-intelligence/cbest-threat-assessment.

[7] *COBIT | Control Objectives for Information Technologies*. URL: https://www.isaca.org/resources/cobit.

[8] *Who is the OWASP Foundation?* URL: https://owasp.org/.

[9] SecurityTrails Team. *SecurityTrails: What is CVE? - Common Vulnerabilities and Exposures*. Dec. 2019. URL: https://securitytrails.com/blog/what-is-cve#top-4-cve-databases.

[10] SecurityTrails Team. *SecurityTrails: Top 8 Exploit Databases for Security Researchers*. Feb. 2020. URL: https://securitytrails.com/blog/top-exploit-databases.

[11] *Common Weakness Enumeration*. URL: https://cwe.mitre.org/about/index.html.

[12] *The MITRE Corporation*. URL: http://www.mitre.org/.

[13] *Mitre Corporation*. [Accessed: 19-04-2020]. Apr. 2020. URL: https://en.wikipedia.org/wiki/Mitre_Corporation.

[14] *Enterprise Matrix*. URL: https://attack.mitre.org/matrices/enterprise/.

[15] *Overview*. URL: https://collaborate.mitre.org/attackics/index.php/Overview.

[16] *Mitigations*. URL: https://attack.mitre.org/mitigations/.

[17] *Groups*. URL: https://attack.mitre.org/groups/.

[18] *Software*. URL: https://attack.mitre.org/software/.

[19] *Contribute*. URL: https://attack.mitre.org/resources/contribute/.

[20] *MITRE ATT&CK® EVALUATIONS*. URL: https://attackevals.mitre.org/.

[21] Jeong Do Yoo et al. “Cyber Attack and Defense Emulation Agents”. In: *Applied Sciences* 10.6 (2020), p. 2140.

[22] Mitre. *mitre/cascade-server*. Nov. 2018. URL: https://github.com/mitre/cascade-server.

[23] Mitre. *mitre/caldera*. Apr. 2020. URL: https://github.com/mitre/caldera.

[24] Redcanaryco. *redcanaryco/atomic-red-team*. Apr. 2020. URL: https://github.com/redcanaryco/atomic-red-team.

[25] Mitre. *mitre/cti*. Mar. 2020. URL: https://github.com/mitre/cti.

[26] *Sharing threat intelligence just got a lot easier!* URL: https://oasis-open.github.io/cti-documentation/.

[27] Sarah Yoder. *Automating Mapping to ATT&CK: The Threat Report ATT&CK Mapper (TRAM) Tool*. Dec. 2019. URL: https://medium.com/mitre-attack/automating-mapping-to-attack-tram-1bb1b44bda76.

[28] Christopher A. Korban et al. “APT3 Adversary Emulation Plan”. In: *The MITRE Corporation, Tech. Rep.* (Sept. 2017), 2–1–2–1. URL: https://attack.mitre.org/docs/APT3_Adversary_Emulation_Plan.pdf.

[29] Tim MalcomVetter. *Red Team Use of MITRE ATT&CK*. Oct. 2018. URL: https://medium.com/@malcomvetter/red-team-use-of-mitre-att-ck-f9ceac6b3be2.

[30] Blake E Strom et al. “Finding cyber threats with ATT&CK-based analytics”. In: *The MITRE Corporation, Tech. Rep.* (2017).

[31] Sagar Samtani et al. “AZSecure Hacker Assets Portal: Cyber threat intelligence and malware analysis”. In: *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. Ieee. 2016, pp. 19–24.

[32] Bilyana Lilly et al. “Applying Indications and Warning Frameworks to Cyber Incidents”. In: *2019 11th International Conference on Cyber Conflict (CyCon)*. Vol. 900. IEEE. 2019, pp. 1–21.

[33] Siddharth Chowdhury. “PERCEPTIONS OF PURPLE TEAMS AMONG CYBER-SECURITY PROFESSIONALS”. PhD thesis. Purdue University Graduate School, 2019.

[34] *Purple Teaming with Vectr, Cobalt Strike, and MITRE ATT&CK^{TM}*. May 2019. URL: `https://www.digitalshadows.com/blog-and-research/purple-teaming-with-vectr-cobalt-strike-and-mitre-attck/`.

[35] Jason Kick. *Cyber exercise playbook*. Tech. rep. MITRE CORP BEDFORD MA, 2014.

[36] Mika Karjalainen, Tero Kokkonen, and Samir Puuska. “Pedagogical Aspects of Cyber Security Exercises”. In: *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2019, pp. 103–108.

[37] David B Fox et al. “Cyber Wargaming: Framework for Enhancing Cyber Wargaming with Realistic”. In: *HSSEDI, The MITRE Corporation* (2018).

[38] Ensar Seker and Hasan Huseyin Ozbenli. “The concept of cyber defence exercises (cdx): Planning, execution, evaluation”. In: *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE. 2018, pp. 1–9.

[39] Mark Russinovich and Thomas Garnier. *Sysmon - Windows Sysinternals*. URL: `https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon`.

[40] Mark Russinovich. *Autoruns for Windows - Windows Sysinternals*. URL: `https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns`.

[41] a2o. *a2o/snoopy*. Dec. 2018. URL: `https://github.com/a2o/snoopy`.

[42] Carson Zimmerman. “Ten strategies of a world-class cybersecurity operations center”. In: *MITRE corporate communications and public affairs. Appendices* (2014).

[43] Aol. *aol/moloch*. Apr. 2020. URL: `https://github.com/aol/moloch`.

[44] Risto Vaarandi. *SEC - simple event correlator*. URL: `https://simple-evcorr.github.io/`.

[45] Richard C Schaeffer. “CNSS instruction No. 4009: National Information Assurance (NIA) Glossary”. In: *Maryland: Committee on National Security Systems* (2010).

[46] Francisco Jesús Rubio Melón, Teemu Uolevi Väisänen, and Mauno Pihelgas. “EVE and ADAM: Situation Awareness Tools for NATO CCDCOE Cyber Exercises”. In: *Systems Concepts and Integration (SCI) Panel SCI-300 Specialists’ Meeting on ‘Cyber Physical Security of Defense Systems’*. 2018, STO–MP.

[47] Kaie Maennel, Rain Ottis, and Olaf Maennel. “Improving and measuring learning effectiveness at cyber defense exercises”. In: *Nordic Conference on Secure IT Systems*. Springer. 2017, pp. 123–138.

[48] Nancy J Cooke et al. “Cyber situation awareness and teamwork”. In: *EAI Endorsed Transactions on Security and Safety* 1.2 (2013).

[49] *Grafana vs. Kibana: The Key Differences to Know*. Apr. 2020. URL: `https://logz.io/blog/grafana-vs-kibana/`.

[50] BigData Boutique. *Alerting with Elasticsearch and the Elastic Stack - BigData Boutique’s Ask Me Anything*. Youtube. 2020. URL: `https://youtu.be/hjQcON_HZAs?t=1317`.

[51] Alerta. *alerta/alerta*. Apr. 2020. URL: `https://github.com/alerta/alerta`.

[52] Jay Kreps, Neha Narkhede, Jun Rao, et al. “Kafka: A distributed messaging system for log processing”. In: *Proceedings of the NetDB*. Vol. 11. 2011, pp. 1–7.

[53] *Introduction*. URL: `https://kafka.apache.org/intro`.

[54] Ccdcoe. *ccdcoe/go-peek*. Feb. 2020. URL: `https://github.com/ccdcoe/go-peek/tree/master`.

[55] Markus Kont. *markuskont/go-sigma-rule-engine*. URL: `https://github.com/markuskont/go-sigma-rule-engine`.

[56] Dan Barker and Dan Barker. *3 open source log aggregation tools*. URL: `https://opensource.com/article/18/9/open-source-log-aggregation-tools`.

[57] *University of Oxford: building a next generation SIEM*. Apr. 2020. URL: `https://www.elastic.co/elasticon/tour/2019/london/oxford-university-building-a-next-generation-siem`.

[58] UpGuard. *Splunk vs ELK: Which Works Best For You?* Nov. 2019. URL: `https://www.upguard.com/articles/splunk-vs-elk`.

[59] *What is Splunk - Splunk Meaning and Splunk Architecture*. Feb. 2020. URL: `https://intellipaat.com/blog/what-is-splunk/`.

[60] *Splunk Architecture: Forwarder, Indexer & Search Head Tutorial*. May 2019. URL: `https://www.edureka.co/blog/splunk-architecture/`.

[61] Jeff Petters. *What is SIEM? A Complete Beginner's Guide - Varonis*. Mar. 2020. URL: `https://www.varonis.com/blog/what-is-siem/`.

[62] Markus Kont et al. "Frankenstack: Toward Real-time Red Team Feedback". In: *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE. 2017, pp. 400–405.

[63] Thoufique Haq, Jinjian Zhai, and Vinay K Pidathala. *Advanced persistent threat (APT) detection center*. US Patent 9,628,507. Apr. 2017.

[64] Louise Byrne. *A Beginner's Guide to Threat Hunting*. Oct. 2019. URL: `https://securityintelligence.com/a-beginners-guide-to-threat-hunting/`.

[65] David Bianco. *The Pyramid of Pain*. Mar. 2013. URL: `http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html`.

[66] Nuthan Munaiah et al. "Characterizing Attacker Behavior in a Cybersecurity Penetration Testing Competition". In: *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE. 2019, pp. 1–6.

[67] Joel Brynielsson et al. "Using cyber defense exercises to obtain additional data for attacker profiling". In: *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE. 2016, pp. 37–42.

[68] Jose David Mireles, Jin-Hee Cho, and Shouhuai Xu. "Extracting attack narratives from traffic datasets". In: *2016 International Conference on Cyber Conflict (CyCon US)*. IEEE. 2016, pp. 1–6.

[69] Benjamin Bornholm. "Network-based APT profiler". In: (2019).

[70] The Mitre Corporation. *MITRE ATT&CK® : APT29 Techniques Mapped to Mitigations and Data Sources*. Accessed: 7-5-2020.

Appendices

Appendix 1 provides an example of the final record generated from the first one hundred events in Domain 1. The report shows a description, atomic tests, mitigations, detections and data sources for monitoring of one technique, the rest of the techniques is omitted in this Appendix.

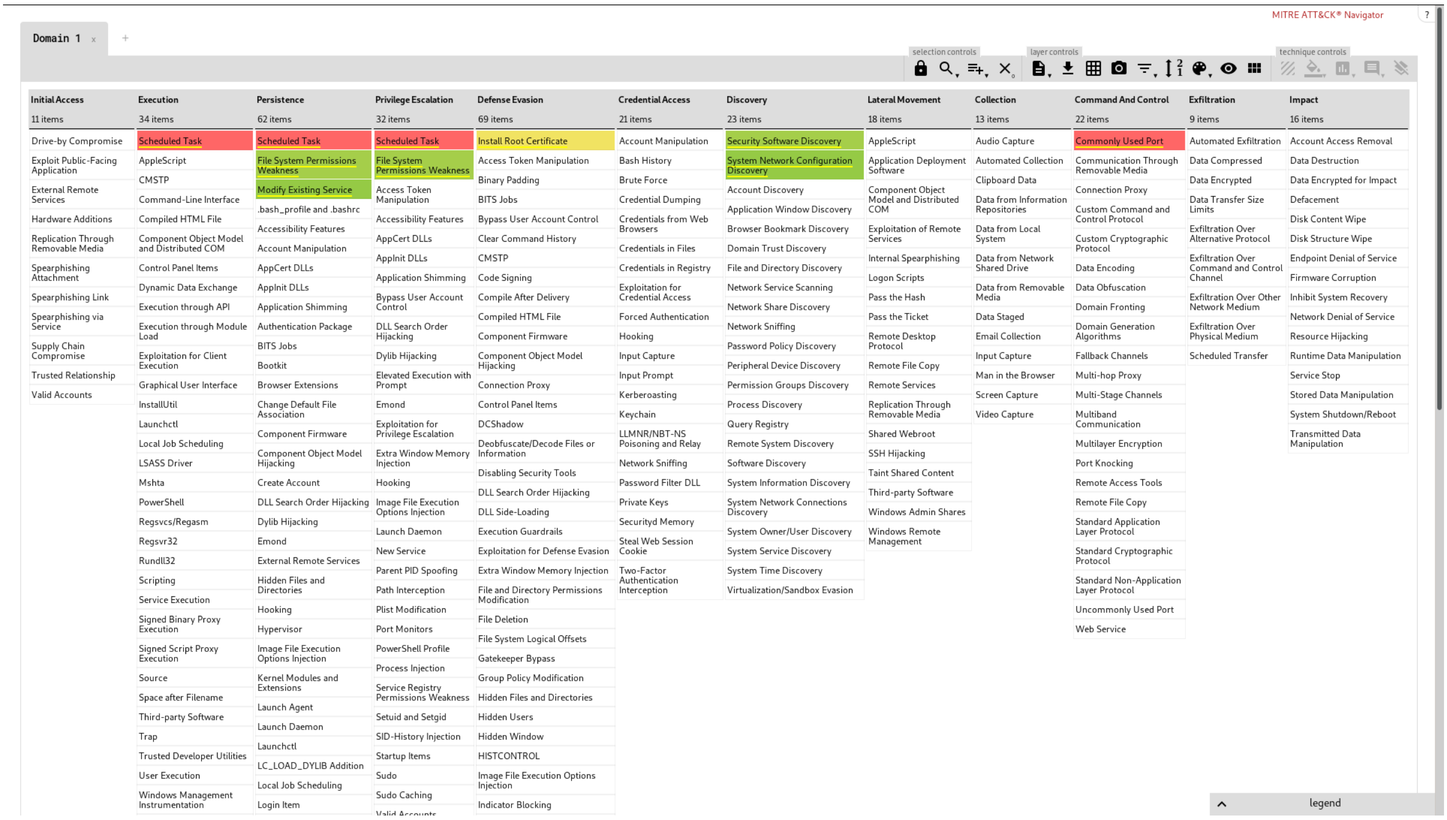
Appendix 1 - Report

report.md

XS 2020 Report

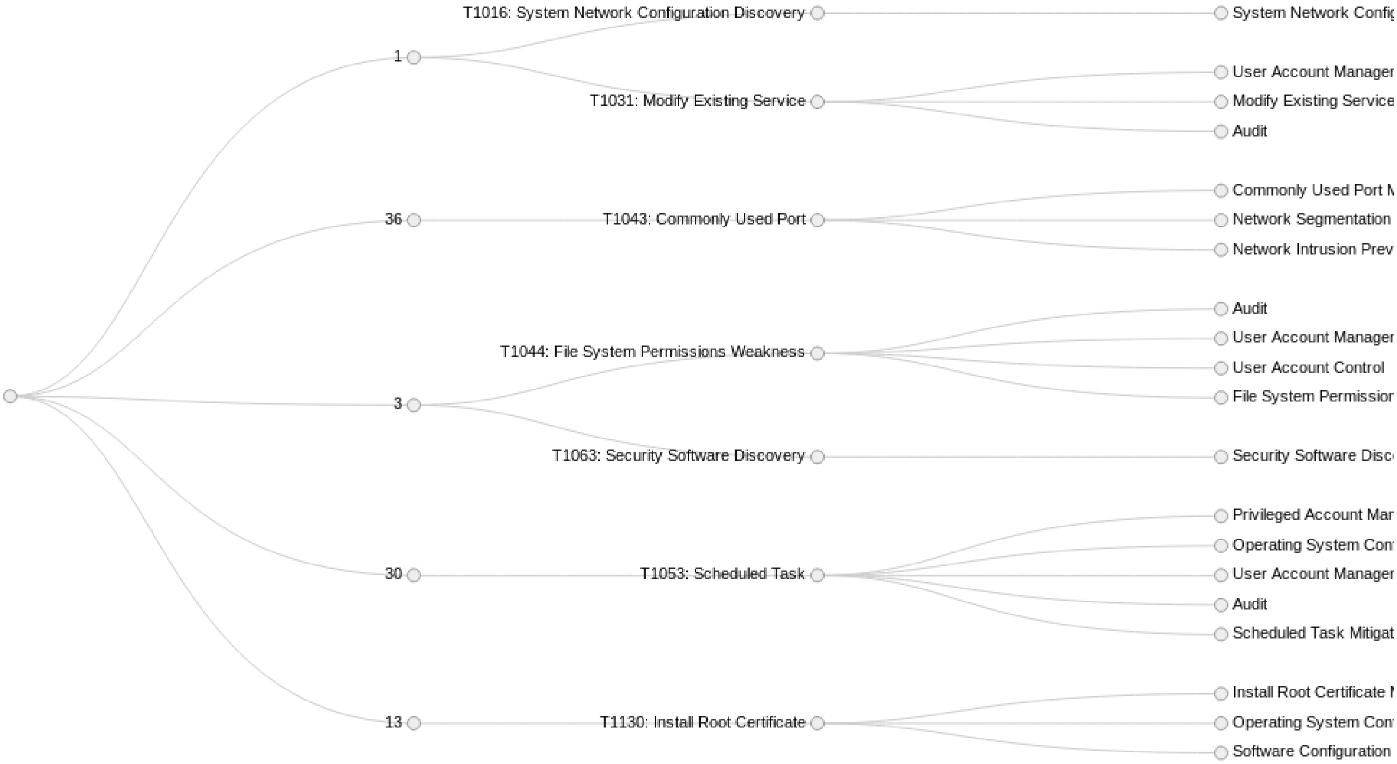
> This report summarises techniques used by Red Team against [Domain 1]. The first part consists of graphs representing technique, mitigations and data sources for monitoring technique occurrence. There is a brief description of each adversary technique used against this domain, an example of atomic tests which can validate of implementation of monitoring/alerting tools, proposed mitigations and data sources necessary for monitoring. It is recommended to use this report along with five timestamps method for feedback purposes.

Graph 1: Used techniques visualised in ATT&CK matrix

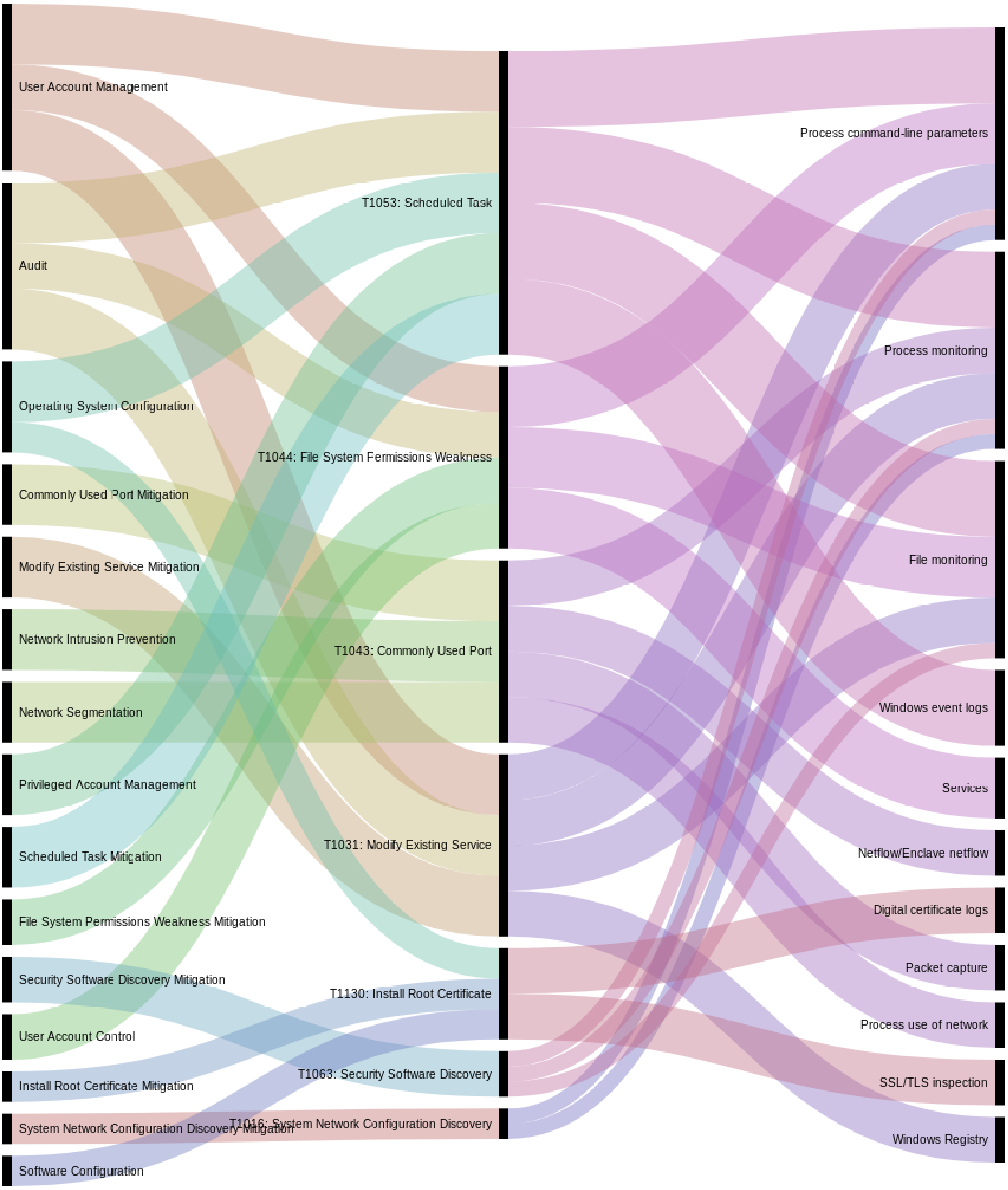


Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	34 items	62 items	32 items	69 items	21 items	23 items	18 items	13 items	22 items	9 items	16 items
Drive-by Compromise	Scheduled Task	Scheduled Task	Scheduled Task	Install Root Certificate	Account Manipulation	Security Software Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	AppleScript	File System Permissions Weakness	File System Permissions Weakness	Access Token Manipulation	Bash History	System Network Configuration Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	CMSTP	Modify Existing Service	Access Token Manipulation	Binary Padding	Brute Force	Account Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Command-Line Interface	.bash_profile and .bashrc	Accessibility Features	BITS Jobs	Credential Dumping	Application Window Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Compiled HTML File	Accessibility Features	AppCert DLLs	Bypass User Account Control	Credentials from Web Browsers	Browser Bookmark Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Component Object Model and Distributed COM	Account Manipulation	AppInit DLLs	Clear Command History	Credentials in Files	Domain Trust Discovery	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Control Panel Items	AppCert DLLs	Application Shimming	CMSTP	Credentials in Registry	File and Directory Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Dynamic Data Exchange	AppInit DLLs	Bypass User Account Control	Code Signing	Exploitation for Credential Access	Network Service Scanning	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through API	Application Shimming	DLL Search Order Hijacking	Compile After Delivery	Forced Authentication	Network Share Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Execution through Module Load	Authentication Package	Dylib Hijacking	Compiled HTML File	Hooking	Network Sniffing	Remote File Copy	Input Capture	Fallback Channels		Network Denial of Service
Valid Accounts	Exploitation for Client Execution	BITS Jobs	Elevated Execution with Prompt	Component Firmware	Input Capture	Password Policy Discovery	Remote Services	Man in the Browser	Multi-hop Proxy		Resource Hijacking
	Graphical User Interface	Bootkit	Emond	Component Object Model Hijacking	Input Prompt	Peripheral Device Discovery	Replication Through Removable Media	Screen Capture	Multi-Stage Channels		Runtime Data Manipulation
	InstallUtil	Browser Extensions	Exploitation for Privilege Escalation	Connection Proxy	Kerberoasting	Permission Groups Discovery	Shared Webroot	Video Capture	Multiband Communication		Service Stop
	Launchctl	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Keychain	Process Discovery	SSH Hijacking		Multilayer Encryption		Stored Data Manipulation
	Local Job Scheduling	Component Firmware	Hooking	DCShadow	LLMNR/NBT-NS Poisoning and Relay	Query Registry	Taint Shared Content		Port Knocking		System Shutdown/Reboot
	LSASS Driver	Component Object Model Hijacking	Image File Execution Options Injection	Deobfuscate/Decode Files or Information	Network Sniffing	Remote System Discovery	Third-party Software		Remote Access Tools		Transmitted Data Manipulation
	Mshta	Create Account	Launch Daemon	Disabling Security Tools	Password Filter DLL	Software Discovery	Windows Admin Shares		Remote File Copy		
	PowerShell	DLL Search Order Hijacking	New Service	DLL Search Order Hijacking	Private Keys	System Information Discovery	Windows Remote Management		Standard Application Layer Protocol		
	Regsvcs/Regasm	Dylib Hijacking	Parent PID Spoofing	DLL Side-Loading	Securityd Memory	System Network Connections Discovery			Standard Cryptographic Protocol		
	Regsvr32	Emond	Path Interception	Execution Guardrails	Steal Web Session Cookie	System Owner/User Discovery			Standard Non-Application Layer Protocol		
	Rundll32	External Remote Services	Plist Modification	Exploitation for Defense Evasion	Two-Factor Authentication Interception	System Service Discovery			Uncommonly Used Port		
	Scripting	Hidden Files and Directories	Port Monitors	Extra Window Memory Injection		System Time Discovery			Web Service		
	Service Execution	Hooking	PowerShell Profile	File and Directory Permissions Modification		Virtualization/Sandbox Evasion					
	Signed Binary Proxy Execution	Hypervisor	Process Injection	File Deletion							
	Signed Script Proxy Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	File System Logical Offsets							
	Source	Kernel Modules and Extensions	Setuid and Setgid	Gatekeeper Bypass							
	Space after Filename	Launch Agent	SID-History Injection	Group Policy Modification							
	Third-party Software	Launch Daemon	Startup Items	Hidden Files and Directories							
	Trap	Launchctl	Sudo	Hidden Users							
	Trusted Developer Utilities	LC_LOAD_DYLIB Addition	Sudo Caching	Hidden Window							
	User Execution	Local Job Scheduling	Valid Accounts	HISTCONTROL							
	Windows Management Instrumentation	Login Item		Image File Execution Options Injection							
				Indicator Blocking							

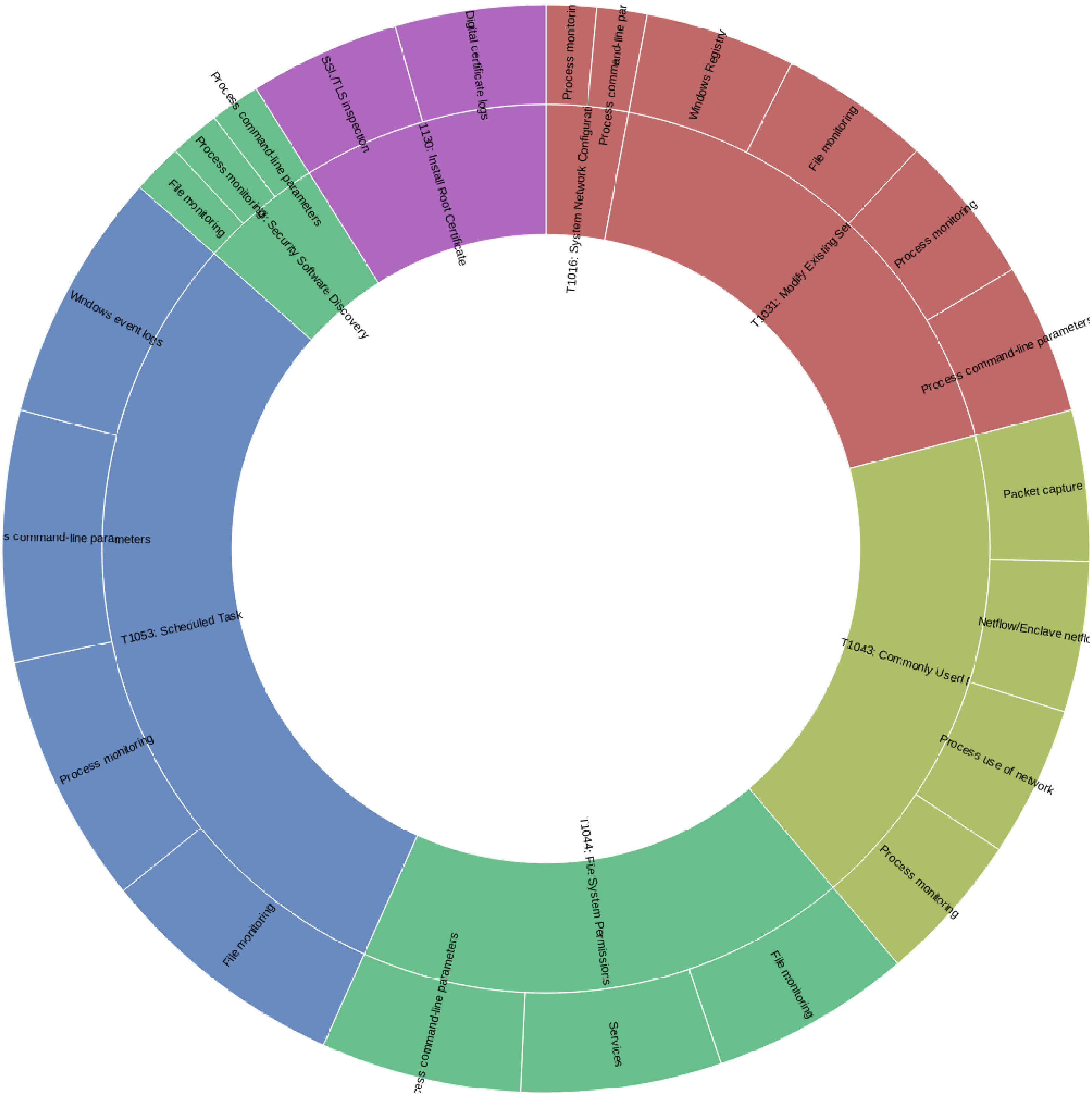
Graph 2: Used techniques and their mitigations based on technique occurance



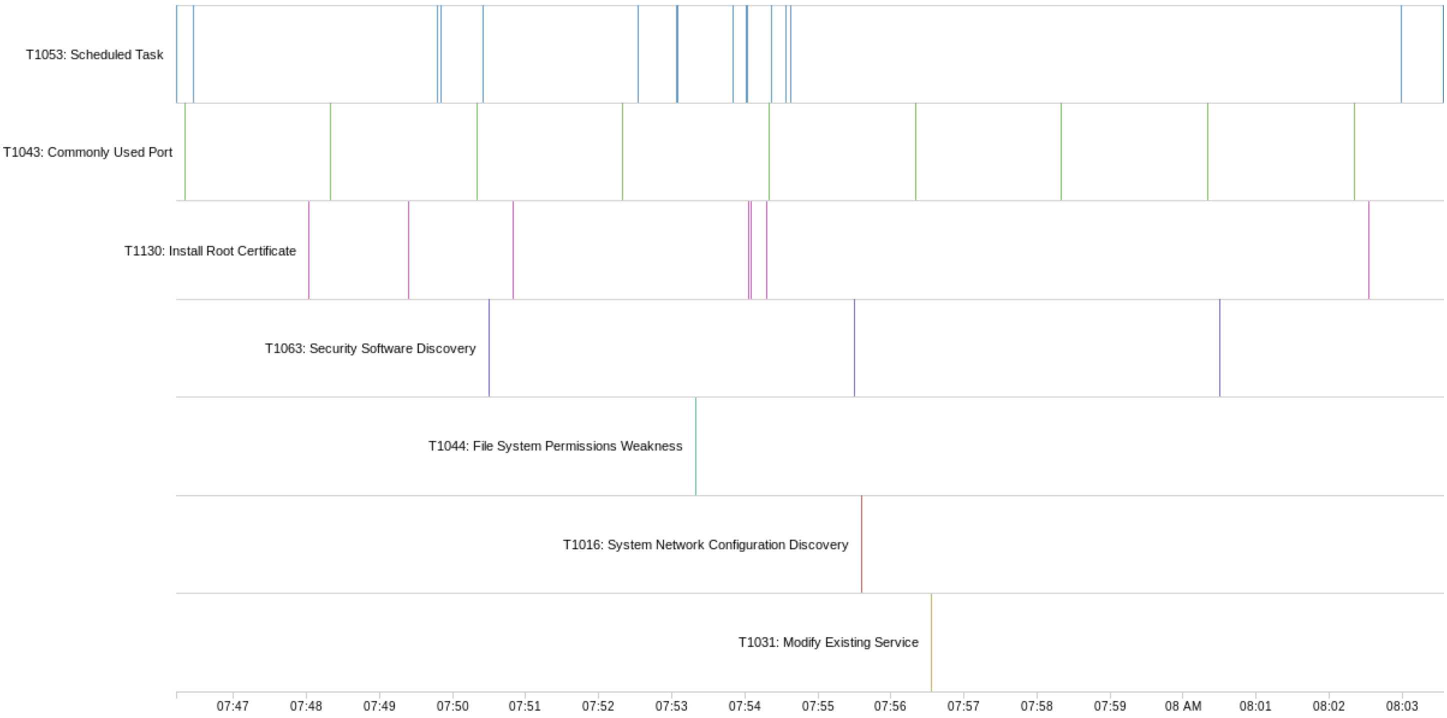
Graph 3: Mitigations, used techniques and sources for monitoring based on technique occurance


User Account Management
Audit
Operating System Configuration
Commonly Used Port Mitigation
Modify Existing Service Mitigation
Network Intrusion Prevention
Network Segmentation
Privileged Account Management
Scheduled Task Mitigation
File System Permissions Weakness Mitigation
Security Software Discovery Mitigation
User Account Control
Install Root Certificate Mitigation
System Network Configuration Discovery Mitigation
Software Configuration
T1053: Scheduled Task
T1044: File System Permissions Weakness
T1043: Commonly Used Port
T1031: Modify Existing Service
T1130: Install Root Certificate
T1063: Security Software Discovery
T1016: System Network Configuration Discovery
Process command-line parameters
Process monitoring
File monitoring
Windows event logs
Services
Netflow/Enclave netflow
Digital certificate logs
Packet capture
Process use of network
SSL/TLS inspection
Windows Registry

Graph 4: Used techniques and data sources for monitoring, based on technique occurance



Graph 5: Used techniques in time



Used techniques sorted by descending occurrance

- T1031: Modify Existing Service

T1031: Modify Existing Service

Description from ATT&CK

> Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Registry. Service configurations can be modified using utilities such as sc.exe and [Reg] (https://attack.mitre.org/software/S0075).
> Adversaries can modify an existing service to persist malware on a system by using system utilities or by using custom tools to interact with the Windows API. Use of existing services is a type of Masquerading that may make detection analysis more challenging. Modifying existing services may interrupt their functionality or may enable services that are disabled or otherwise not commonly used.
>
> Adversaries may also intentionally corrupt or kill services to execute malicious recovery programs/commands. (Citation: Twitter Service Recovery Nov 2017) (Citation: Microsoft Service Recovery Feb 2013)

Atomic Tests

- Atomic Test #1 - Modify Fax service to run PowerShell

Atomic Test #1 - Modify Fax service to run PowerShell

This test will temporarily modify the service Fax by changing the binPath to PowerShell and will then revert the binPath change, restoring Fax to its original state.

Upon successful execution, cmd will modify the binpath for `Fax` to spawn powershell. Powershell will then spawn.

Supported Platforms: Windows

Attack Commands: Run with `command_prompt` ! Elevation Required (e.g. root or admin)

```
sc config Fax binPath= "C:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -noexit -c \"write-host 'T1031 Test'
sc start Fax
```

Cleanup Commands:

```
sc config Fax binPath= "C:\WINDOWS\system32\fxssvc.exe" >nul 2>&1
```

Mitigations for T1031:

User Account Management

Manage the creation, modification, use, and permissions associated to user accounts.

References

- mitre-attack:: https://attack.mitre.org/mitigations/M1018

Modify Existing Service Mitigation

Use auditing tools capable of detecting privilege and service abuse opportunities on systems within an enterprise and correct them. Limit privileges of user accounts and groups so that only authorized administrators can interact with service changes and service configurations. Toolkits like the PowerSploit framework contain the PowerUp modules that can be used to explore systems for Privilege Escalation weaknesses. (Citation: Powersploit)

Identify and block potentially malicious software that may be executed through service abuse by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown programs.

References

- mitre-attack:: https://attack.mitre.org/mitigations/T1031

- Powersploit::PowerSploit. (n.d.). Retrieved December 4, 2014.

https://github.com/mattifestation/PowerSploit

- Beechey 2010::Beechey, J. (2010, December). Application Whitelisting: Panacea or Propaganda?. Retrieved November 18, 2014.

http://www.sans.org/reading-room/whitepapers/application/application-whitelisting-panacea-propaganda-33599

- Windows Commands JPCERT::Tomonaga, S. (2016, January 26). Windows Commands Abused by Attackers. Retrieved February 2, 2016.

http://blog.jpcert.or.jp/2016/01/windows-commands-abused-by-attackers.html

- NSA MS AppLocker::NSA Information Assurance Directorate. (2014, August). Application Whitelisting Using Microsoft AppLocker. Retrieved March 31, 2016.

https://www.iad.gov/iad/library/ia-guidance/tech-briefs/application-whitelisting-using-microsoft-applocker.cfm

Audit

Perform audits or scans of systems, permissions, insecure software, insecure configurations, etc. to identify potential weaknesses.

References

- mitre-attack:: https://attack.mitre.org/mitigations/M1047

Detection of T1031:

Look for changes to service Registry entries that do not correlate with known software, patch cycles, etc. Changes to the binary path and the service startup type changed from manual or disabled to automatic, if it does not typically do so, may be suspicious. Tools such as Sysinternals Autoruns may also be used to detect system service changes that could be attempts at persistence. (Citation: TechNet Autoruns)

Service information is stored in the Registry at `HKLM\SYSTEM\CurrentControlSet\Services` .

Command-line invocation of tools capable of modifying services may be unusual, depending on how systems are typically used in a particular environment. Collect service utility execution and service binary path arguments used for analysis. Service binary paths may even be changed to execute cmd commands or scripts.

Look for abnormal process call trees from known services and for execution of other commands that could relate to Discovery or other adversary techniques. Services may also be modified through Windows system management tools such as Windows Management Instrumentation and PowerShell, so additional logging may need to be configured to gather the appropriate data.

Data Sources for T1031:

Windows Registry

File monitoring

Process monitoring

Process command-line parameters