Incorporating S.P.I.C.Y. DICOM Polyglot Threats into Cyber Warfare Exercises

Danny "Danhammer" Hetzel[1,2], Xavier Palmer[1,3], Lucas Potter[3] and Janine (Nina Alli) Medina[1]

[1]Biohacking Village, USA
[2]Husar Security LLC, USA
[3]BiosView Labs, USA

danny@hetzelsecurity.com (Corresponding Author)

Abstract: The technical feasibility of crafting polyglot files, objects valid under multiple file formats, has been repeatedly demonstrated in various contexts, including the DICOM medical imaging standard. While prior work has established that DICOM polyglots are technically constructible, a significant operational gap exists between this known vulnerability and the preparedness of healthcare organizations to detect and mitigate such data-centric threats. This paper addresses this critical oversight by shifting the focus from how to construct DICOM polyglots to why they pose an unmitigated, systemic risk in clinical workflows. We argue that this operational gap stems from three primary issues: non-uniform inspection across heterogeneous security tools, misplaced trust in the structural compliance of medical data, and insufficient threat modelling for automated pipelines. Our feasibility analysis, using a controlled DICOM/Portable Executable (PE) polyglot, demonstrates that mainstream security tools exhibit substantial inconsistencies in detection, with different systems interpreting the same object according to varying, often shallow, parsing assumptions. This non-uniform inspection allows the polyglot to remain stealthy and polyglot. To provide a structured framework for analyzing and mitigating this threat, we introduce the S.P.I.C.Y. framework, defined by five operational characteristics: Stealthy, Polyglot, Infiltrative, Cascading, and Yield. S.P.I.C.Y. helps practitioners measure the operational impact of malicious data artifacts that propagate through trusted, automated workflows, such as Picture Archiving and Communication Systems (PACS) and Vendor Neutral Archives (VNAs). The Infiltrative and Cascading characteristics highlight how an object, once past initial inspection, maintains its dual functionality across multiple automated processes. Finally, we redefine Yield to include not just execution, but the operational value gained from persistence within trusted infrastructure. By applying the S.P.I.C.Y. framework, organizations can move beyond traditional security models and strategically address the critical blind spots created by operational complexity and the assumed trust placed in medical data standards.

Keywords: DICOM, Polyglot, Healthcare, Cybersecurity, S.P.I.C.Y., Operational gap, Automated pipeline

1. Introduction: The Dual Nature of Medical Data

Cybersecurity increasingly exploits ambiguities in trusted digital systems. As automation and usability take their toll, these trust boundaries begin to waver. Antiquated standards and automation create assumed trust boundaries as files are pushed through increasingly complex pipelines. Adversarial techniques increasingly take advantage of these structural ambiguities. Polyglots take advantage of these ambiguities. Polyglots are defined as a single byte sequence that is syntactically valid under multiple file format specifications and can be meaningfully processed by more than one independent parser. Prior research has identified numerous file formats capable of supporting polyglot constructions, such as PDF, Jpeg, ZIP and many others. (Albertini, 2015). Through polyglot techniques, it becomes possible to construct objects that expose weaknesses in assumptions about file structure and interpretation.

The DICOM standard is a foundational fixture in medical imaging. Security validation has not matured alongside security threats, leaving a gap in operational security. The DICOM Part 10 file structure is pushed through automated pipelines repeatedly. Practitioners face attack surface gaps formed from inconsistent detection and overlapping weakened trust boundaries. The DICOM file structure has led researchers to discover a critical example of a threat that develops from such ambiguity (Ortiz, 2019). Polyglot-capable conditions resulting from the asymmetry between extensibility in the DICOM standard and validation gaps in detection tools have created an attack vector that relies on opaque file structures and avoids most inspection.

Current frameworks do not provide a means to assess the risk of data-centric threats and their potential operational blast radius. While previous research has demonstrated feasibility, the operational impact and implications for automated medical systems remain insufficiently explored (Ortiz, 2019). We propose the S.P.I.C.Y. framework to assist practitioners in analysing these polyglot threats and expose defensive gaps created by operational assumptions and complexity. Traditional threat modelling frameworks, such as the Cyber Kill Chain and MITRE ATT&CK, focus on specific vulnerabilities or attacker techniques and tactics

(Hutchins et al., 2011; Strom et al., 2018). S.P.I.C.Y. instead measures the operational impact of malicious data artifacts that propagate through automated workflows and trusted pipelines.

2. Theoretical Framework: S.P.I.C.Y. and the Polyglot Taxonomy

2.1 S.P.I.C.Y. Framework Introduction

Polyglots as a threat form an ambiguous, even hard to quantify, gap in the standard assessments for exploits and vulnerabilities. Traditional threat models evaluate threats through software flaws or execution paths. Polyglots function as a result of data interpretation and structural format ambiguity. To help define and describe such behavior this paper introduces the S.P.I.C.Y. framework, which is defined through five operational characteristics: Stealthy, Polyglot, Infiltrative, Cascading, Yield.

Stealthy is used to understand how polyglots are positioned to abuse ambiguous structural elements of formats, standards, and specifications to evade consistent detection. The ability of a polyglot to leverage these ambiguities depends on the gap in analysis and inconsistency of security tools in their respective environments (Albertini, 2015). Stealthy asks: How consistently is the polyglot object recognized across the different layers of inspection in a workflow?

Polyglot emphasizes the object's ability to be valid under different interpretations and contexts. Simultaneously able to conform to multiple formats and remain valid. As an example, different DICOM polyglots conform to distinct dual states (Ortiz, 2019). Polyglot asks: How many separate unique file formats can the object conform to while retaining core functionality?

Infiltrative describes the object's ability to pass different layers of ingestion without raising alarms. DICOM polyglots are accepted by typical ingestion services like DICOMweb STOW and then flow into PACS systems (Pianykh, 2012; Kahn et al., 2016). Infiltrative asks: Can this object successfully be ingested at point of entry without failing to validation or other inspection tools?

Cascading alludes to the polyglot's ability to successfully retain its multifaceted functionality despite moving through different transformations and systems as it moves through workflows. DICOM files move through PACS, VNAs, and AI workflows. The ability of polyglot to contain data within private elements and preamble show these structures may endure (Mirsky et al., 2019). Cascading asks: does the polyglot retain functionality through these transfers and transformations applied by the pipeline?

Yield identifies the operation value that is achieved once a structurally stable polyglot enters and persists in the workflow. Execution is not the only valid operational goal. The presence of executable code embedded in trusted medical data allows for a form of persistent, deferred activation. Downstream handling and reuse of images allow for exposure and embedding of code further than reliant on immediately executable code could achieve.

2.2 Polyglot Taxonomy

We have defined the taxonomy of polyglots into three sets of definable characterizations-structural, semantic, and operational - to provide a systematic framework for understanding how polyglot files are constructed, evade defensive controls, and are weaponized in operational workflows. This taxonomy is intended to address the concerns of a distinct but overlapping audience involved in the security and governance of data systems: the structural taxonomy is relevant to software developers, standard authors, and tool builders, allowing for an understanding of byte-level construction that remains syntactically valid under multiple formats; the semantic taxonomy is directly applicable to security engineers, detection vendors, or incident responders and focuses on inspection and validation mechanisms that fail when polyglots are encountered in practice; and the operational taxonomy is directed toward system architects, risk managers, and executive stakeholders with the goal of mapping functional roles in attack chains and TTPs (Tactics, Techniques, and Procedures). Understanding how these malicious polyglots are constructed is key to developing effective defenses. What follows are several taxonomy categories, their definitions, and examples to facilitate understanding. These include structural taxonomy, semantic taxonomy, and operational taxonomy.

Structural taxonomy defines the organizational structure of a polyglot file. This includes sequential polyglots, where two or more complete files are placed back-to-back, relying on parsers to ignore trailing data, which can bypass file type validation and allow lists. Another type is embedded or container-based polyglots, where one valid file format is contained within the data fields of another while both remain compliant. This allows for data transmission by surviving compression and metadata rewriting. Finally, interleaved polyglots involve the

same bytes being interpreted differently by two parsers due to overlapping semantic meaning. The technique can be used to evade or break parser-based detection, static analysis, or machine learning extraction.

Semantic taxonomy defines how definitive security measures fail or are evaded by the polyglot. Type-confusion polyglots are characterized by a mismatch between the declared file type and the actual parser behavior, serving as an evasion vector for magic-byte identification or file upload validation and enabling downstream abuse. Inspection-evasion polyglots occur when security tools inspect only one semantic interpretation of the file while ignoring others, allowing the polyglot to achieve persistence even in inspected environments. Context-switch polyglots are benign in one operational context but become malicious in another, which allows for delayed or cascading exploitation while evading shallow assessments without breaking workflows.

Operational taxonomy maps polyglots into specific attack usage scenarios. Ingress polyglots are used specifically for smuggling across security boundaries. A persistence polyglot has the purpose of surviving archival, sharing, or long-term storage without triggering alerts. Lastly, activation polyglots are designed with the specific purpose of being activated through a pipeline, workflow, or some form of triggered automation.

Together, these provide perspectives on how polyglots should be analysed-not as mere anomalies in file formatting but as intentional malicious instruments used to pry exploitable gaps between file semantics, defensive tooling, and system trust boundaries.

3. Feasibility Demonstrated: The Operationalized DICOM Polyglot

3.1 Methodology and Research Environment

For the purpose of this work, a table of components used in the investigation is provided. These help the audience to have a sense of what is needed to replicate findings and what to look for. Parameters are default.

Table 1: Components used for this investigation

Item	Definition	Purpose	Version
Malware Bytes	Commercial endpoint anti-malware solution	Evaluated local static detection behavior of constructed artifacts	5.4.7.229(Component Package 148.0.5470)
Orthanc server	Open-source DICOM server implementing DICOM storage and query/retrieve services	Used to simulate PACS ingestion, storage behavior, send studies via STOW, and reserialization of DICOM objects	1.12.9
Weasis Medical Viewer	Open-source DICOM viewer	Used to validate image rendering and observe viewer handling of modified DICOM objects	4.6.3
HxD Hex Editor	Low-level binary file editor	Used for byte-level inspection and structural modification of file regions for crafting the exploit	2.5.0.0
Windows 11	Host operating system environment	Used to evaluate file association behavior, invocation context, and endpoint inspection responses	24H2 (OS Build 26100.7623)
MinGW-w64 GCC 13.2.0	C code compiler	Used to compile executable Windows PE code	13.2.0

This paper looks at the potential uses of polyglot file techniques in relation to the DICOM Part 10 file structure, as discovered by d00rt (Ortiz, M.O.,2019). Our research was executed in three phases. The first phase defined a taxonomy of the structural, semantic, and operational attributes for polyglot files. This phase also placed the category of DICOM-based polyglot files within the existing body of research concerning different types of file formats.

The second phase of the work involved an analysis of the DICOM Part 10 file structure, specifically examining the preamble, file meta information, dataset encoding, and private data elements. These components were

utilized to assess the susceptibility to potential polyglot exploitation and to identify areas where dual-valid constructs could exist while maintaining compliance with the DICOM Data Dictionary.

In the third phase, controlled laboratory DICOM polyglots were created. Payload artifacts were written in C/C++ and compiled into Windows Portable Executable (PE) binaries, using MinGW-w64 GCC 13.2.0 on Windows 11, to be embedded as test payloads. The payloads were embedded into anonymized, functioning DICOM part 10 image files while maintaining structural validity using the tools and environment from Table 1.

Evaluation of the polyglot behavior across ingestion, archival, inspection, and detection workflows was completed to monitor how polyglots would interact with common medical imaging environments. Using a fully patched and updated Windows 11, the objects were scanned by Defender and Malwarebytes to observe end point detection. They were then placed in a directory and uploaded as a study via Orthanc using STOW to be stored in a receiving Orthanc DICOMweb server for storage. This step was done to observe possible alteration to the file that may occur during ingestion. The items were then rescanned by Defender and Malwarebytes, checked for executability on windows or Linux machines, and lastly, tested for the ability to be viewed as an image in Weasis.

The paper includes a description of the research environment, tools used for construction and validation, and static analysis results to illustrate detection variability. Findings are interpreted through the S.P.I.C.Y. model-Stealthy, Polyglot, Infiltrative, Cascading, Yield-to formalize how such artifacts function in offensive simulations. The goal is not to promote exploit development, but to provide a structured framework for evaluating data-centric threat surfaces that may be overlooked in conventional security training focused primarily on network and credential compromise.

3.2 DICOM Vulnerability Context

A DICOM file structured according to the Part 10 file specification is constructed of four primary components: a preamble, a file prefix, the file meta information, and the actual data set. The preamble, the set of initial 128 bytes, which for the DICOM Standard is unutilized space, often filled with null bytes, and is reserved for third party or application specific purposes. This section is designed to maintain compatibility between DICOM part 10 files and software or legacy systems that may rely on a specific header offset. This is a characteristic that will later become significant for exploitation.

Following the preamble, a four-byte prefix with the string 'DICOM' provides the file signature, a particular set of characters used to explicitly identify the file as a valid DICOM Part 10 file. The third main section is the File Meta Information. This section exclusively contains data elements from the group (0002, xxxx) and is mandatorily encoded with an Explicit VR Little Endian transfer syntax, irrespective of the encoding applied to the subsequent data set. Important elements within the File Meta Information include the Media Storage SOP Class UID, Media Storage SOP Instance UID, Transfer Syntax UID, and Implementation Class UID. The Transfer Syntax UID specifies the encoding method for the remainder of the data set. This includes byte ordering (Little-endian or Big-endian) and whether Value Representations are explicit or implicit. The final and largest component of the DICOM file is the dataset. This segment of the file contains the data elements representing metadata and the pixel data of the images. It is a collection of all the data elements encoded given instructions in the Transfer Syntax. Every Data element may be broken down further into four sub-components or fields. The first is the Tag, a numeric identifier consisting of the group number and element number. Next is the Value Representation, which is only included with explicit VR encoding. Third, is the Value Length, a parameter that specifies the length in bytes of the value field. Finally, the Value Field which contains the stored image data.

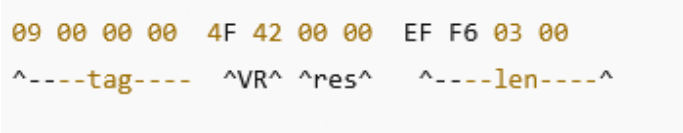

Figure 1: The Private Tag

DICOM files are structured using a series of Data Elements, each uniquely identified by a tag comprising a group number and an element number (Figure 1). Each Data Element possesses a Value Representation (VR), which specifies the data type (e.g., text, numeric, sequence), and a Value Length, indicating the size in bytes. The encoding, including byte ordering, is governed by the negotiated Transfer Syntax. This rigid yet flexible layout ensures reliable parsing across different systems while accommodating the wide variety of medical data stored in DICOM files. The format distinctly separates public and private Data Elements. Public Data Elements

are formally defined within the DICOM standard and utilize even-numbered Group Values. Conversely, private Data Elements employ odd-numbered Group Values, reserved for vendor-specific or third-party usage. Prior to utilizing private Data Elements, a vendor must first define a Private Creator Data Element to reserve a range of Data Element Numbers for proprietary use. This mechanism facilitates the embedding of proprietary metadata without compromising adherence to the DICOM standard. While compliant systems typically include and transmit private Data Elements as-is, they are not necessarily interpreted unless the receiving application can access the associated private dictionary. Consequently, when this access is unavailable, the private data can appear as an opaque entity to external parties, intended for another system. This phenomenon can occur throughout the entire processing pipeline, including ingestion, archival, forwarding, and viewing, meaning that private fields, regardless of their source or format, may pass through without detailed examination.

The customizable attributes permitted by the DICOM format, while offering flexibility for interoperability, introduces an inconsistency in the level of scrutiny of compliant data elements. Public data elements are typically well-defined and normally validated, but private fields may contain data in an arbitrary format. The private field formats may only be known to the vendor creating an issue where the code is opaque to any DICOM parser and may not be analysed. Given the increasingly mixed applications and development of clinical environments where images traverse multiple platforms, a valid private field can persist unchanged for an indefinite amount of time and processing. This allows private Data Elements to potentially embed tertiary data structures while remaining a valid DICOM object. The associated risk stems not from DICOM's specifications but from the asymmetry between the application of the extensibility and the failure to consistently validate private data elements throughout the medical image pipeline.

3.3 Proof of Concept: Constructing the Dual-Valid Artifact (Technique Validation)

Here in Figure 2, we can see the first 128 bytes of a Dicom file with the first 26 bytes over-written with a Bash script. By using the hex editor application HxD, we are able to alter the hex code of the file to add the script. It contains the '#!' of the bash script followed by a basic whoami command and a proper exit from the script. This allows the DICOM file to now function as a DICOM image collection and a bash script. This is our first example of a DICOM polyglot proof-of-concept. Its simplicity in creation means that automation and prolific generation would take small amounts of resources, time, or effort.

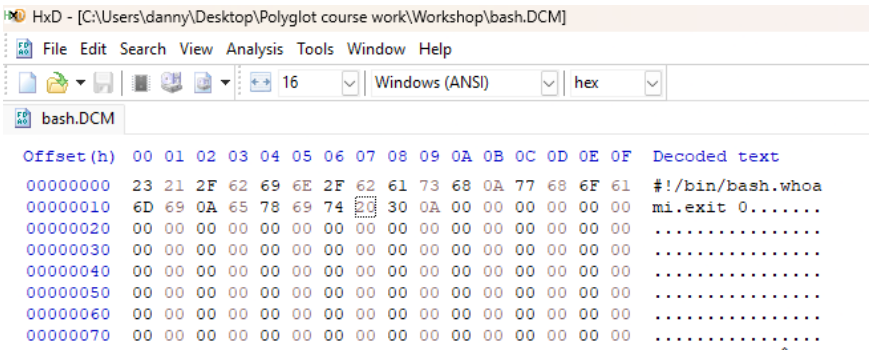

Figure 2: HxD table: An initial example illustrating the exploitability of DICOM and its utility as an ingress vector involves abusing the preamble. The first 128 bytes of the DICOM preamble are commonly ignored by data parsers. This vulnerability permits the injection of any file up to 128 bytes, which most DICOM parsers will subsequently disregard

The next part of the DICOM part 10 File we will attempt to demonstrate as exploitable is the Private tag. In this example we will use both the 128-byte preamble and the private tag data element in conjunction to create a working DICOM/PE (Portable Executable) file.

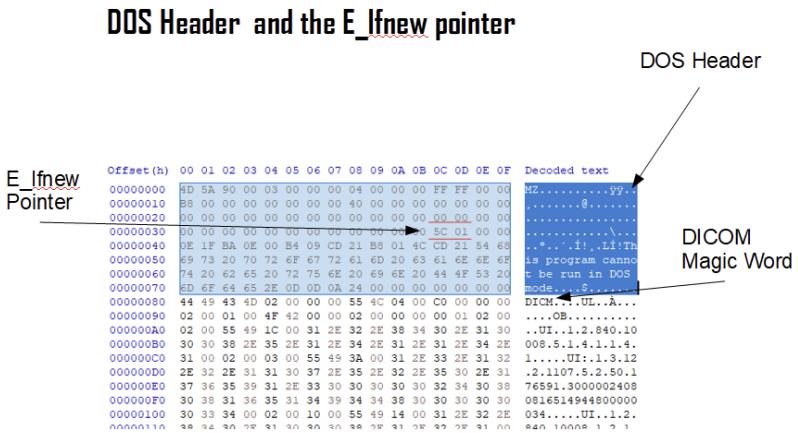

Figure 3: DICOM Magic Word, (File Signature), DOS Header, and E_lfnew Pointer locations in the Hex code.

In Figure 3, we can see how the DOS header of the PE perfectly fits into the blank Preamble of the DICOM. Leveraging the PE e_lfnew Pointer, we can redirect the PE flow to the PE file signature. If the pointer is used to point further into the DICOM file toward data embedded in a private element identified with a private tag, we can place the remainder of the PE inside this Data element and with a high likelihood it will not be parsed by most DICOM viewers. Further alteration of the private tag to match length, adding proper padding to the PE to ensure alignment, and altering the remaining PE pointers to the correct offsets will now allow a PE to function while not disrupting the DICOM data structure (Figure 4).

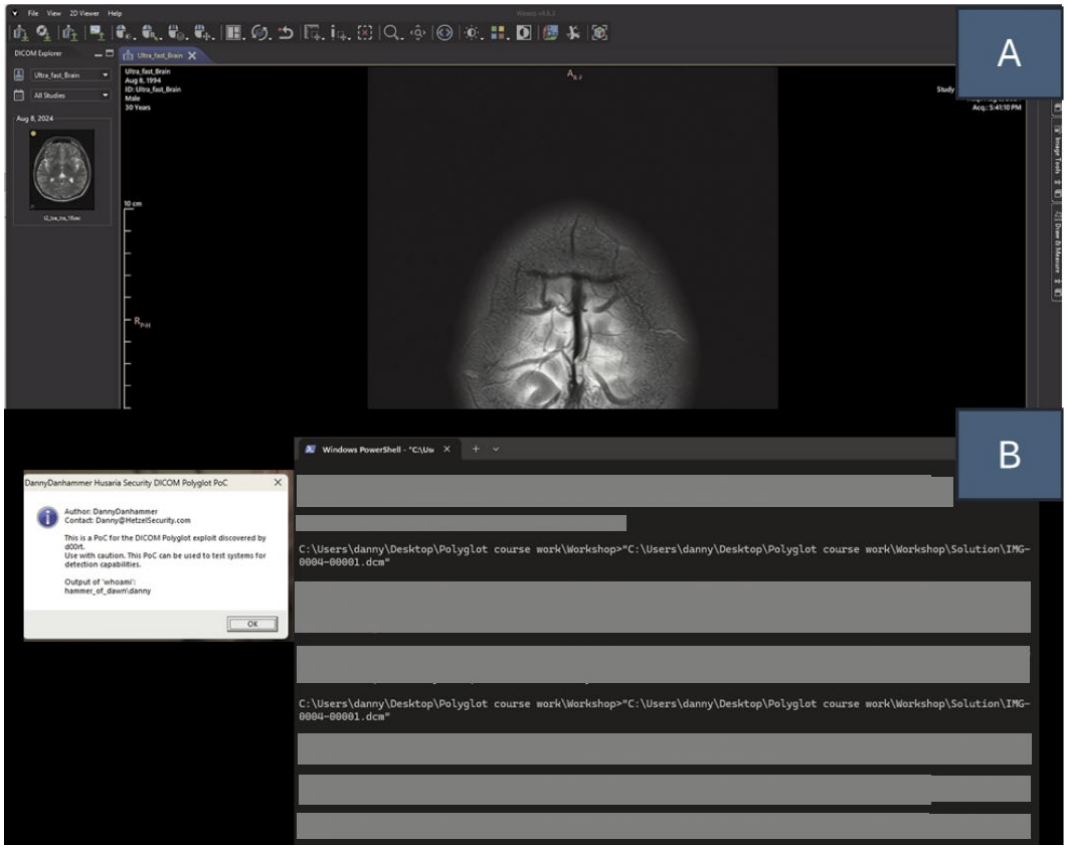

Figure 4: DICOM Image and Execution: A) The DICOM image opens with the associated DICOM parser with a double click. B) But when opened via the terminal it executes out payload PE. (Unnecessary terminal commands redacted provide focus.)

4. The Operational Gap: Why Feasibility Translates to Risk

4.1 Non-Uniform Inspection and the Stealthy/Polyglot Gap

A key defining characteristic of polyglots is their chameleon-like ability to remain syntactically valid under multiple file formats simultaneously (Albertini, 2015). Their dual, and sometimes ternary, validity introduces a challenge for security tools that assumes a file conforms to a single format. In healthcare imaging environments, where DICOM objects are commonly treated as clinical data rather than untrusted adversarial inputs, prioritization of schema conformance and renderability by inspection mechanisms is seen rather than full byte-stream analysis. As such, data artifacts containing multiple semantic interpretations may not be consistently classified across security tools.

Observations from the feasibility phase of this study illustrate this asymmetry in inspection. The polyglot proof-of-concept developed in Section 3 was analysed using the multi-engine malware analysis tool VirusTotal and the endpoint detection software Malwarebytes. VirusTotal had a mixed reporting of 11 out of 71 security tools reporting as possibly malicious (Figure 5). While a score of 11/71 would raise suspicions to many, the inability to run DICOM through public scanners and requiring the choice of a local antivirus software, creating an environment where evasion only needs to succeed in 1 or the 71 presented. This places a higher importance on the failure to detect proportion from 84.51% of assessed antivirus products. When executed within heuristic-capable sandboxes on VirusTotal, the polyglot triggered only seven low-severity and three informational MITRE signatures, despite performing indirect command execution and process injection. Although VirusTotal was able to identify the polyglot through aggregated results, many mainstream security tools failed to detect it independently.

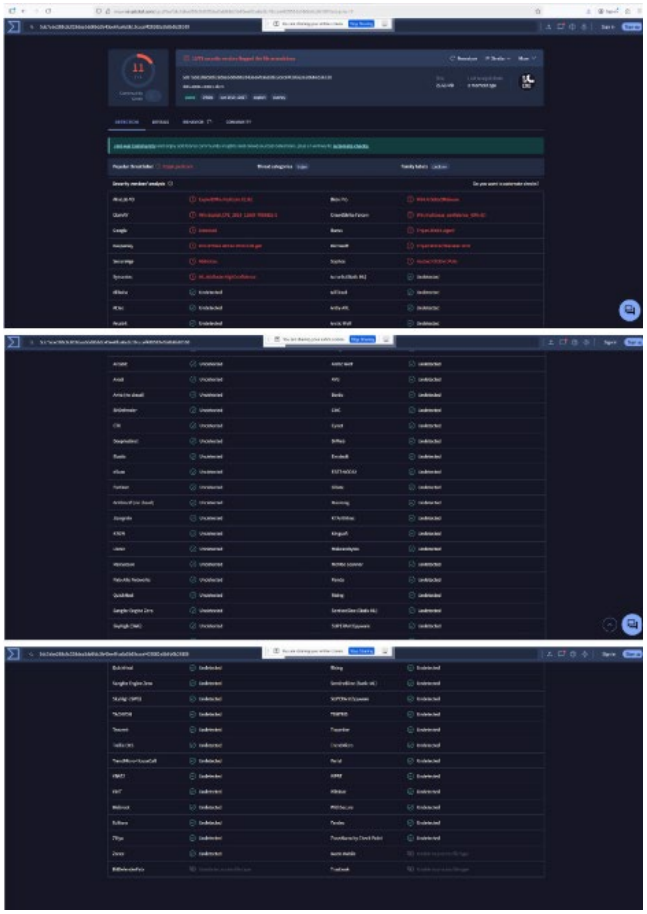

Figure 5: A) Analysis of the artifact using VirusTotal. The VirusTotal report indicates that 11 security providers (highlighted in red) successfully identified the artifact, while numerous other providers failed to detect the exploit, despite its relationship to a widely recognized vulnerability

To further assess the local endpoint detection behavior, the polyglot was scanned using Malwarebytes (version 5.4.6.227) in a controlled environment. At the time of the test, Malwarebytes reported no detection (Figure 6).

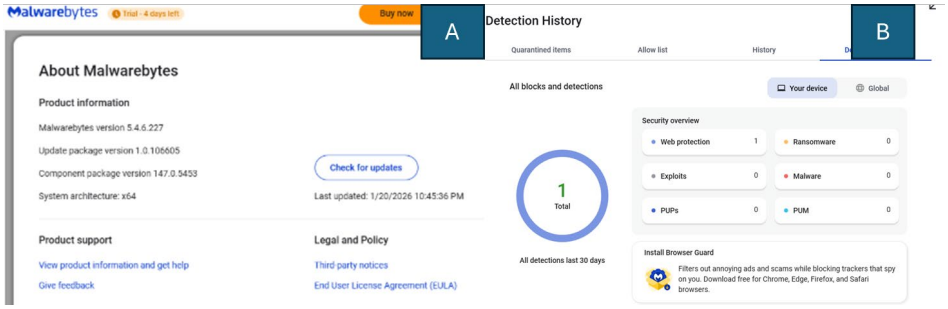

Figure 6: A) Malwarebytes, a consumer anti-malware program. B) Diagnostic screen demonstrating failure to detect the artifact

This gap indicates that different security products interpreted the polyglot file according to different assumptions and approaches to parsing. Some products readily identified byte patterns associated with executable structures embedded in the file and within the preamble. Others accepted the format presented as a simple DICOM object and did not do further analysis to uncover the patterns of possibly obfuscated code. This is not attempting to illustrate a failure of this specific product but rather the lack of uniformity in detection. Previous research has highlighted product to product inconsistencies are deeply ingrained (Mohaisen and Alrawi, 2014). Antivirus products fundamentally disagree with each other on both detection and labelling, to a degree that makes their outputs, by themselves, unreliable as ground truth. Classification of multi-format or polyglot files may not be reliably detected by a single security product. Polyglots exploit this ambiguity by presenting different semantic interpretations depending on the parsing logic applied. A polyglot does not need to evade all security tool detections to succeed operationally. It must only pass through enough stages of inspection to enter the trusted workflow. In this sense, stealth arises not from sophisticated obfuscation techniques but from classification ambiguity. Within the S.P.I.C.Y. model, this represents the intersection of the Stealthy and Polyglot characteristics: the polyglot file remains structurally compliant while different systems disagree about its interpretation. This inconsistency can be operationally significant. In the complex enterprise environment, security controls rarely rely on a single security tool or inspection method. As files move through different systems and scanners, each applying its own parsing logic, the likelihood of inconsistent classification increases.

4.1.1 The standards non-operational state

After 7 years of existing as a known vulnerability, the persistence of this gap reflects a broader systemic issue within the healthcare ecosystem, as opposed to a flaw in DICOM standard itself. DICOM was designed as a tool to offer interoperability across diverse imaging devices and software platforms. To accomplish this feat, the standard permits extensibility. Extensibility is achieved through features of the standard such as vendor specified private tags and flexible metastructures. (National Electrical Manufacturers Association (NEMA), 2026). These features enable vendors to embed proprietary information while maintaining compliance with the standard. This extensibility also introduces ambiguous areas within the file that creates issues for uniformity in parsing.

In practice, many implementations of "DICOM validation" focus on verifying structural conformance or the ability of imaging software to render the imaging data correctly (Mirsky et al., 2019). Parsers and viewers, such as Weasis, only focus on the imaging and ignore unnecessary sections of data. While these checks are essential for interoperability, they do not necessarily enforce deep comprehensive inspection of the entire object. The result is what can be described as a non-operational security state. The format's extensibility is fully implemented but the corresponding security validations are not consistently enforced across tools or environments. Several factors contribute to this condition further. Complete comprehensive semantic and structural inspection of every DICOM object can be computationally expensive and could result in latency in critical clinical workflows and pipelines. Next, traditional healthcare security architectures acknowledge the dangers of ingress points of email attachments, web downloads, and others, but imaging data pipelines are often treated as trusted channels (Mahler et al., 2018).

To properly address this operational gap requires moving beyond the assumption that format compliance alone guarantees safety within trusted workflows. Standards bodies, vendors, and healthcare operators must consider approaches that apply more thorough validation practices. Potential directions include establishing baseline expectations for byte-stream inspection of DICOM objects, defining clearer policies for handling

private data elements, and developing security-focused conformance profiles that can be implemented consistently across heterogeneous systems. Such efforts would not eliminate the flexibility that makes DICOM interoperable but would help ensure that this flexibility does not translate into systemic blind spots within clinical imaging workflows.

4.2 Automated Pipelines and the Infiltrative/Cascading Gap:

While the inconsistent inspection and parsing in security tools has shown how such objects can evade detection, feasibility alone cannot translate into operational risk, the polyglot must have the ability to traverse real workflows. Modern medical Imaging infrastructure in healthcare environments are not static storage. They are interconnected data pipelines in which DICOM objects move through multiple automated systems over the course of its lifecycle (Pianykh, 2012). A single image study may originate from an imagining source ,such as a MRI scanning system, pass through ingestion services (Store-SCP or DICOMweb STOW), enter a Picture Archiving and Communication System (PACS), be transferred into a long term archival system such as Vendor Neutral Archive (VNA) and finally be accessed by diagnostic viewers , exported to research sets, or even processed by artificial intelligence and machine learning pipelines (Kahn, 2016). Throughout the course of its lifecycle, the DICOM object is repeatedly parsed, indexed, modified, and redistributed by automated systems (Pianykh, O.S., 2012).

For the S.P.I.C.Y. model, this corresponds to the Infiltrative and Cascading phases of the lifecycle. This infiltrative phase occurs when a polyglot object enters the pipeline through legitimate ingestion. Because the polyglot follows the syntax and structural requirements for the DICOM File part 10 specification, systems that rely primarily on structural validation may accept the object without detecting its secondary semantics (Pianykh, 2012; Huang, 2018). The cascading occurs when the polyglot is parsed and pushed through multiple steps of the pipeline while being able to maintain its malicious capabilities (Mirsky et al., 2019; Mahler et al., 2018). Within this framework, the infiltration and persistence become properties and assets of the workflow rather than of a single exploit. Using the ambiguity of the heterogenous workflow that due to assumed trust, the accepted structurally valid polyglot may propagate throughout the environment, through multiple systems, that independently interpret the same data, allowing the Stealthy and Polyglot characteristics of the S.P.I.C.Y. model to persist across the pipeline.

5. Discussion: Activation, Yield, and Strategic Implications

5.1 Activation Context in Automated Workflows

The execution and activation of a DICOM polyglot does not require manual user double-click. In healthcare environments, as discussed in section 4, there are numerous contexts in which a DICOM object might be parsed. Dicom objects are routinely processed by automated systems outside simple graphical viewers. These interactions can occur through command-line utilities, batch processing scripts, research pipelines, and more recently into machine learning and AI contexts. At each step the possibility of execution based on varying interpretation may occur because each system may apply different parsing assumptions to the same byte stream.

This inconsistent nature of the different forms of interpretation creates the conditions necessary for execution or activation. A viewer may see a plain DICOM image to be displayed, while another subsystem or script may interact with the same object and interpret the same data differently by the embedded syntax. This disparity forms the path to polyglot execution. A file, now embedded deep within the architecture and having crossed multiple trust boundaries, may be invoked through command-line tools, scripting environments, or automated data transformation processes.

Recent practitioner work suggests that the process to create such polyglots will become increasingly automated. Aguilar (2026) and Davis (2026) describe a proof-of-concept toolkit of embedding crafted malicious Windows Portable Executables (PE) binaries within valid DICOM images through an automated process and leveraging and merging further advanced evasion tactics. Invoked through command-line contexts while remaining undetected through automated creation and built in features including beaconing. Although the toolkit has not been formally released or subjected to peer review, its existence presents an important operation leap forward from the manual creation presented in this paper: The technical barrier to constructing DICOM polyglots is decreasing and automation of the threat is becoming a significant reality. As tooling matures, the threshold for adversarial use is decreased.

5.2 Redefining Yield: The Scope of Damage:

Conventional exploit analysis often measures success primarily in terms of payload execution. Within data-centric attack surfaces, yield must be understood more broadly. Operational value from a polyglot can extend beyond execution by successfully entering and persisting within trusted workflows. Successful ingestion and persistence demonstrate that the artifact can survive inspection boundaries and remain structurally intact. These conditions create the opportunity for later activation through lateral vulnerabilities within the system.

Under the S.P.I.C.Y. model, yield asks what the operational value is gained once a polyglot artifact successfully traverses trusted workflows. The most direct outcome is achieving execution when the polyglot is interpreted in a specific execution context. Yield may also arise from the persistence described previously, as the ability of the polyglot to remain embedded within trusted workflows enables adversaries to position executable objects deep inside clinical infrastructure. From a defensive perspective, this broader definition of yield is critical. It repositions analysis of the threat from a binary condition of exploit success or failure into a graduated model of system exposure.

5.3 Strategic Implications and Mitigation

The use of DICOM polyglots in threat exercises challenges the assumption that medical data is inherently non-executable; this finding fits within the understanding of biocentric data, media, and platforms being relevant for attack chains (Mueller, 2021; Potter and Palmer, 2023; Pope, 2026). Medical images were designed to ensure reliability and interoperability between software, platforms, and devices. As a result, security controls have often developed behind these design principles in prioritization. Business needs and attempting to support legacy systems has left an operational gap in detection. When a file satisfies the structural requirements of the DICOM standard, generally, it is treated as a valid medical object and allowed to proceed through the workflow as a trusted object. This design philosophy has operational consequences and, in those consequences, lies the gaps that researchers and adversarial actors will eventually discover and leverage; further commentary follows in Figure 7.

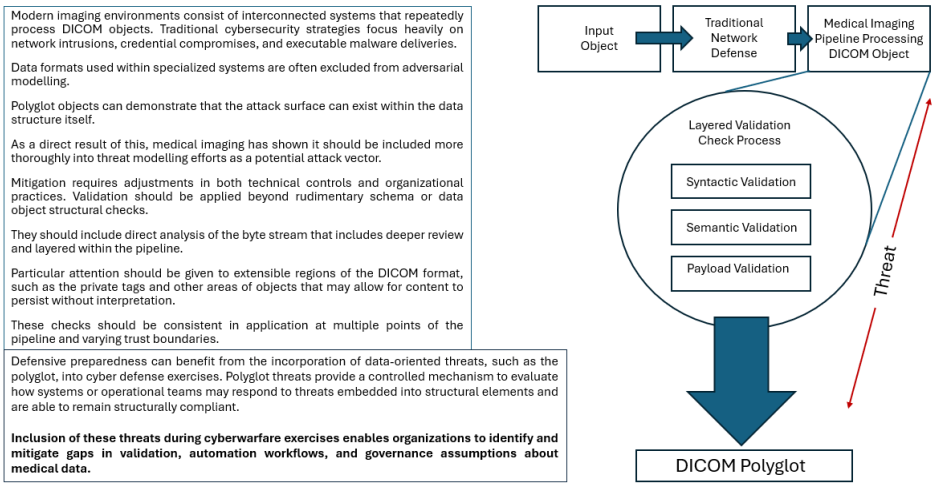

Figure 7: Design and preparedness philosophy commentary around DICOM Polyglots

The S.P.I.C.Y. framework offers a guided approach for evaluating these scenarios. Through examining how an artifact remains stealthy, infiltrates trusted workflows, cascades across connected systems, and produces operational yield, defenders can better understand how data-layer threats affect healthcare infrastructure. Using S.P.I.C.Y. as a guide creates an avenue for organizations to overcome the gaps presented by the traditional exploit models and help them move toward a broader understanding of how trusted data structures and standards can be abused to become attack surfaces. Avenues created through the use of guides such as these potentially create opportunities for healthcare organizations to improve perceptions of patient and provider sense of security, within a world that is increasingly subject to healthcare-centric cyber-attacks (Griffin et al, 2026).

6. Limitations and Future Works

The paper lacks empirical experimental data to aid exact reproducibility. The choice to limit this data is due to the nature of harm that could be weaponized and data hazards that could be produced with replication of the

experiment. Upon responsible request and discussion, a guided walkthrough to aid in reproducibility can be obtained from the author. While it may appear that the tools used were rather limited, VirusTotal comprises 70+ tools, thus mitigating this concern. Future work involves generation of a larger proof-of-concept dataset which can be tested for detection by security analysis tools. This will result in the ability to create a more quantified and firmer consensus to the state of DICOM polyglot detection in industry tools.

7. Conclusion

This paper has shifted the discussion around DICOM polyglots from technical feasibility to unmitigated operational risk within healthcare workflows, highlighting a critical security gap stemming from non-uniform inspection, misplaced trust in data compliance, and insufficient threat modelling. Our feasibility analysis, using a controlled DICOM/PE polyglot, demonstrated that mainstream security tools exhibit substantial inconsistencies in detection, allowing these dual-valid artifacts to remain stealthy. To address this, we introduced the S.P.I.C.Y. framework as a tool for practitioners to measure the operational impact of malicious data propagation through trusted, automated pipelines like PACS and VNAs. The framework emphasizes that yield extends beyond execution to include the operational value gained from persistence deep within clinical infrastructure. Ultimately, mitigating this systemic risk requires moving beyond the assumption that format compliance guarantees safety, urging standards bodies, vendors, and operators to apply consistent, byte-stream-level validation across the entire medical image lifecycle to close the operational blind spots exposed by data-centric threats.

Ethics Declaration: Clearance for the research within this paper was not required and no humans were affected.

AI Declaration: Grammarly and Google Docs and AI tools within were used for formatting and overall re-organization.

References

Aguilar, M. ("Vega"), 2026. DICOM Polyglot Attack Kit. Unpublished proof-of-concept toolkit.

Albertini, A. (2015). File Formats Polyglots: Multiple file formats in a single file. Black Hat / Corkami research.

Davis, C., 2026. Oh my vibe hack! SolaSec. https://www.solasec.io/blog/oh-my-vibe-hack/ (Accessed: 12 March 2026).

Griffin, B., Barnett, M., VanNorman, T., Medina, N.J., Potter, L. and Palmer, X. (2026) Ontological security and threat mitigation in healthcare: A descriptive exploration. In: Proceedings of the 21st International Conference on Cyber Warfare and Security (ICCWS 2026). Academic Conferences International Limited. doi: 10.34190/iccws.21.1.4541

Huang, H.K., 2018. PACS and Imaging Informatics: Basic Principles and Applications. Hoboken: Wiley.

Hutchins, E., Cloppert, M. and Amin, R., 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In: Proceedings of the 6th International Conference on Information Warfare and Security (ICIW). Washington, DC, USA, pp. 113–125.

Kahn, C.E., Langlotz, C.P., Burnside, E.S., Carrino, J.A., Channin, D.S., Hovsepian, D.M. and Rubin, D.L., 2016. Toward best practices in radiology reporting. Radiology, 274(2), pp. 473–487.

Mahler, T., Mirsky, Y., Shabtai, A. and Elovici, Y., 2018. Know your enemy: Characteristics of cyber-attacks on medical imaging devices. IEEE Security and Privacy Workshops.

Mirsky, Y., Mahler, T., Shelef, I. and Elovici, Y., 2019. CT-GAN: Malicious tampering of 3D medical imagery using deep learning. Proceedings of the 28th USENIX Security Symposium.

Mohaisen, A. and Alrawi, O., 2014. AV-Meter: An Evaluation of Antivirus Scans and Labels. Proceedings of the 11th International Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA). Cham: Springer, pp. 112–131. Doi: 10.1007/978-3-319-08509-8_7

Mueller, S., 2021. Facing the 2020 pandemic: What does cyberbiosecurity want us to know to safeguard the future?. Biosafety and health, 3(01), pp.11-21.Doi: 10.1016/j.bsheal.2020.09.007

National Electrical Manufacturers Association (NEMA) (2026) DICOM PS3.5: Data Structures and Encoding. Available at: https://dicom.nema.org/medical/dicom/current/output/html/part05.html

Pianykh, O.S., 2012. Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide. 2nd ed. Berlin: Springer.

Pope, T. (2026) Sleeper code in protein data files as cyber adversarial vectors. In: Proceedings of the 21st International Conference on Cyber Warfare and Security (ICCWS 2026). Academic Conferences International Limited. Doi: 10.34190/iccws.21.1.4459

Potter, L. and Palmer, X.L., 2023. Mission-aware differences in cyberbiosecurity and biocybersecurity policies: Prevention, detection, and elimination. In Cyberbiosecurity: A new field to deal with emerging threats (pp. 37-69). Cham: Springer International Publishing. Doi: 10.1007/978-3-031-26034-6_4

Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G. and Thomas, C.B., 2018. Mitre att&ck: Design and philosophy.