THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

COLLEGE OF INFORMATION SCIENCES AND TECHNOLOGY

An Analysis of the SolarWinds Supply Chain Breach via Attack Graphs

RICHARD CHENG
SPRING 2024

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Cybersecurity Analytics and Operations
with honors in Cybersecurity Analytics and Operations

Reviewed and approved* by the following:

Peng Liu
Professor of Cybersecurity
Thesis Supervisor

Nicklaus A. Giacobe
Associate Teaching Professor
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

The 2020 SolarWinds supply chain cyberattack greatly contributed to the evolution of existing areas of study for cyber defense, such as machine learning, network theory, and malware analysis. Attack modeling techniques (AMTs), such as attack graphs, present novel visualizations to enhance the analysis of different security breaches. This paper contributes to the existing literature on the attack graph modeling of large cyberattacks by synthesizing approximately 100 indicators of compromise from a diverse range of sources to provide an intuitive and unfragmented model of the breach on SolarWinds. Subsequent analysis revealed different critical nodes and attack paths that may allow for more robust defensive metrics applicable to other cyber threats. Exploring the utility of attack graphs for cyber threat modeling may offer valuable insights for informed defense efforts.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. Peng Liu for his invaluable time, mentorship, and support throughout the thesis process. I would also like to thank Dr. Nicklaus Giacobe for his guidance and teachings throughout my undergraduate career. Finally, I would like to give special thanks to my family and friends for their continued support. I could not have done it without you all.

Chapter 1

Introduction to the SolarWinds Cybersecurity Breach

On December 13th, 2020, after conducting analysis on their own environment, Mandiant announced that an adversary, now known as APT 29, StellarParticle by CrowdStrike, Dark Halo by Volexity, and NOBELIUM by Microsoft, leveraged a trojanized SolarWinds Orion update to compromise an unprecedented number of organizations (FireEye, 2020). Many high-profile companies, such as Mandiant themselves, Microsoft, Cisco, as well as US and international agencies, such as the Department of Defense, Homeland Security, and the EU agencies, were reported to have been breached (Cimpanu, 2021; Jankowicz & Davis, 2020). Furthermore, an estimated 18,000 of the total 33,000 organizations are reported to have downloaded the weaponized update (Cimpanu, 2020), however, many more were indirectly impacted through the resources and hours needed to conduct post-forensic investigations. The financial implications of the attack cost SolarWinds a minimum of 44 million: 18 million from resolving the incident and another 26 million for a lawsuit (Kovacs, 2022; Satter 2021).

The attack catalyzed a surplus of information sharing from different security organizations; many whitepapers and firsthand accounts detailing indicators of compromise, hashes, and IP addresses helped companies immediately take defensive actions to contain the threat. However, as companies increase their reliance on technology, they often depend on third-party solution providers. Such reliance can create a cyberattack known as a supply chain attack, where adversaries typically alter existing trusted software for malicious purposes. This issue is not specific to third-party applications, but also as software dependencies or API calls (Lenaerts-

Bergmans, 2023b). Thus, the SolarWinds attack presents a unique challenge to the security community that goes against the typical security mantra of updating to the latest software. As supply chains continue to grow prevalent in today's technological landscape, a need to provide visibility and risk communication for the inherent black-box nature of technology grows imperative.

Turning the lens to academia, fields such as machine learning, network theory and attack graphs have been a few of the many fields of study for cyber defense. Attack graphs in particular present an attractive framework for providing clarity in the anatomy of a cyberattack. Post-forensic investigations in particular may greatly benefit from the inclusion of attack graphs, as such representations may assist forensic examiners in analyzing and including evidence of compromise for each path (Liu et al., 2012). Attack graphs also offer strong temporal relationships via the chronological ordering of nodes and edges, which increases its usage in tasks such as network topology modeling or communicating assets compromised (Zenitani, 2023a).

This thesis aims to model the SolarWinds supply chain breach as an attack graph by synthesizing various indicators of compromise. The attack graph and its analysis can help security analysts develop novel insights into defense via attack paths that can be generalized to other cyber incidents.

Timeline of the SolarWinds Attack

A timeline of the events that led up to the breach on SolarWinds is displayed below. Most events are focused on technical details, as activity diminished following public announcements.

August 6th, 2019	Command and control (C2) infrastructure setup (Unit 42, 2020)
September 4th, 2019	Attackers started accessing SolarWinds (CDOC & Microsoft Threat Intelligence, 2021; Ramakrishna, 2021b)
September 12th, 2019	Code injection tests by inserting blank classes (CDOC & Microsoft Threat Intelligence, 2021; Ramakrishna, 2021b)
October 26th, 2019	Earliest identified modification of SolarWinds' Orion code (Unit 42, 2020)
November 4th, 2019	Attackers stopped injecting test code (CDOC & Microsoft Threat Intelligence, 2021; Ramakrishna, 2021b)
December 2019	Earliest Cobalt Strike payload identified generated with Cobalt Strike 4.0 (Unit 42, 2020)
December 6th, 2019	DGA domain avsvmcloud.com acquired (Unit 42, 2020)
February 2020	First SSL certificate acquired (Unit 42, 2020)
February 20th, 2020	Backdoor compiled and deployed (CDOC & Microsoft Threat Intelligence, 2021; Ramakrishna, 2021b)

March 26th, 2020	Hotfix 5 DLL available to customers (Ramakrishna, 2021b). Compromised update released to public.
March - May	Backdoor distribution and profiling (CDOC & Microsoft Threat Intelligence, 2021; Ramakrishna 2021b)
May	Start of hands-on keyboard attacks (CDOC & Microsoft Threat Intelligence, 2021)
June 4th, 2020	Malware removed from SolarWinds build VMs (Ramakrishna, 2021b)
December 8th, 2020	Mandiant announced that their red team penetration testing tools was stolen as victims to a nation-state cyberattack (Baker, 2021)
December 11th, 2020	Mandiant discovers SolarWinds had been attacked (Kiuwan, 2021)
December 12th, 2020	Mandiant disclosed to SolarWinds the Solorigate supply chain attack (Kiuwan, 2021)
December 13th, 2020	Mandiant and SolarWinds announces breach (FireEye, 2020)
December 15th, 2020	SolarWinds releases software fix (Ramakrishna, 2021b), MS seizes C2 domain (Unit 42, 2020)

January 11th 2021	SUNSPOT findings (Ramakrishna, 2021b)

Table 1: Timeline of the SolarWinds breach

Effects and Impact

After the breach was announced, SolarWinds posted details regarding their plan moving forward. Such steps include improving the security of their products by ensuring compiled code matches against source code and resigning products with a new certificate; more interestingly however, SolarWinds took steps into securing their own internal environment through factors such as threat protection software for endpoints, MFA, and credential resets (Ramakrishna, 2021a). In a similar fashion, organizations were given the choice of continuing SolarWinds Orion use or to replace them. Attack graphs may aid these decision processes for both SolarWinds and impacted organizations by identifying overall attack graph traversal and subsequent risk analysis.

Furthermore, at the time of writing, Microsoft is undergoing new investigations into NOBELIUM under a password spraying attack that allowed access to the “company’s source code repositories” (MSRC, 2024). Such actions continue to showcase the importance of improved cyber defense practices that attack graphs may assist in.

Chapter 2

Literature Review

A wide variety of literature was produced as a result of the breach on SolarWinds. Different sectors, such as academia, private and public industries aimed to document and share information regarding the attack to provide visibility and countermeasures.

SolarWinds Related White Papers

Several technology companies posted detailed whitepapers outlining remediation advice and indicators of compromise, such as file hashes, IP addresses and YARA rules. The literature also provided technical details of the attack by providing a high-level analysis of tainted source code and malware reverse engineering results.

Mandiant was the first organization to alert about a possible intrusion campaign, and their reports focused on providing preliminary visibility into the SolarWinds attack (FireEye, 2020). Subsequent analysis from different industries showcased similar results on the attack chain, with some revealing new malware such as GoldMax, GoldFinder, and Sibot (Nafisi et al., 2021) or different techniques (Cash et al., 2020). Notably, these whitepapers outline a series of events that represent indicators of compromise and suspicious information flow.

One notable discrepancy in regard to forensic analysis procedure is to either preserve evidence or upgrade to the latest SolarWinds version. Specifically, the SolarWinds security advisory advocated for applying upgrades for affected versions (SolarWinds, 2021) while Mandiant's recommendations were the opposite to prevent the removal of forensic artifacts

(FireEye, 2020). If upgrade was not possible, Mandiant recommended several hardening measures to mitigate the risk of using the Orion platform.

The research is grounded in these white papers, specifically from organizations with first-hand analysis of the attack. Microsoft (CDOC & Microsoft Threat Intelligence, 2021; Microsoft Threat Intelligence, 2020a; Nafisi et al., 2021), Mandiant (Eckels et al., 2020; FireEye, 2020), CrowdStrike (CrowdStrike Intelligence Team, 2021), and Checkpoint (Check Point Research, 2020) provided technical details and events, and thus serve as a foundation for how the attack graph is created and organized. References were correlated with each other to produce a complete possible timeline of the attack, diving into specifics, such as functions called or domains contacted. Other references include specific pieces of information, such as specific Powershell commands (Cash et al., 2020) or decoded blacklist checks (Cohen, 2023).

Graph Theory in Cybersecurity

Attack Graphs

AMTs encompass a wide variety of visualization formats to analyze cyberattacks (Lallie et al., 2020; Pirca & Lallie, 2023). An attack graph is a type of AMT that is a directed acyclic graph used to model the flow of data (Zenitani, 2023b). They are also commonly used as visual representations for security analysis and can model different scales, such as large networks or individual hosts.

As with any graph, there are two common elements: nodes and edges. Nodes typically represent a host, vulnerability, or network device manipulated by an adversary (Zeng et al., 2019), and the edges between nodes represent the causal relationships, typically in chronological

order (Zenitani, 2023a). An attack path then represents a series of nodes and edges in chronological order that an attacker may take to achieve an objective. As a result, attack graphs capture causal relationships between malicious events and may allow for root cause identification.

These nodes are typically organized by prerequisites. In order for a node to pass to the next node, the previous nodes must have been passed. As a result, prerequisite nodes, or critical nodes, present attractive options for security analysts to identify gaps in networks or areas in need of patching (Liu et al., 2012).

History

There is an extensive history of the use of attack graphs in cybersecurity. Phillips and Swiler published a paper in 1998 regarded as the first graph-based model for network security analysis (Liu & Jiang, 2023; Zenitani, 2023), although other papers such as Dacier (1996) also make use of the same idea. They define nodes as a single state of attack, which may be a "combination of physical machine(s), user access level, and effects of the attack" (Phillips & Swiler, 1998). Edges then represent the change caused by the node before it.

Jha et al. (2002) however, address the attack-centric nature of Phillips & Swiler (1998) by defining attack graphs as models that present the steps, or nodes needed, to reach a goal. They argue that their definition of attack graphs does not account for benign nodes, whereas their model includes both, thus further generalizing its utility for security analysis. They also illustrate how attack graphs assist security analysts in identifying the effectiveness of their detection systems and enhancing correlation between events. Furthermore, the researchers address how the creation of such graphs is tedious and error-prone and offer a possible solution with off-the-shelf model checkers.

Zenitani (2023a) defines attack graphs as a model-based approach for network security, where nodes represent events, edges represent the causal relationships between nodes, and an attack path as steps taken to achieve an objective. Zenitani (2023a) also summarizes the history of the developments and advancements of attack graphs, highlighting how later studies built on Phillips & Swiler's (1998) paper with definitions in relation to categories such as state enumeration graphs, cyclic directed AND/OR Graphs, and special Bayesian attack graphs.

Liu et al. (2012) define attack graphs as directed graphs, where nodes represent exploits with pre- and post-conditions and edges exist if the source node is needed for the destination node to proceed. Their work builds on the use of attack graphs during post-forensics, especially in the case where adversaries use anti-forensic techniques.

Challenges

Overall, existing literature tends to agree on the definition of nodes and edges in attack graphs. However, there appears to be no widely accepted standard for attack graph representation (Lallie et al., 2020; Zenitani, 2023a). The general consensus among research papers focuses on the two core elements, nodes and edges, however, depending on the need for each analysis type, the definitions and format of the attack graph changes. Zenitani (2023b) uses mathematical functions called attack functions to centralize the source to which attack graphs are derived for different purposes, however, the paper focuses more on the underlying mathematical rules to analyze attack graphs rather than how attack graphs are formatted themselves.

Furthermore, one of the first papers researching the visualization of attack graph representations highlights the volume and corresponding inconsistency of attack graph standards Lallie et al. (2020). Most researchers use self-nominated visualization syntax, which has resulted in over 75 attack graph visualization standards, leading to fragmented research efforts and

inconsistency for visual representation. Lallie et al. (2020) also emphasizes how terminology, specifically attack graphs and attack trees, are both graph structures, with key differences being the way each represents preconditions, attacks, and event flows. This lack of standards makes it difficult for researchers to model well-known attacks via attack graphs, as selecting the specific standard is dependent on what the attack graph aims to achieve.

Expanding upon this concept, there appears to be a lack of attack graph examples for well-known cyberattacks. Research endeavors such as Nguyen (2017) present a partial attack graph to model the Stuxnet attack for the integration of uncertainty. While the main purpose of the paper aims to analyze the likelihood of which path(s) are taken and to make informed decisions on which security appliances need hardening, these attack graphs often serve as supplemental information rather than as the main focus. Even more so, attack graph modeling and analysis requires all known vulnerabilities to be known (Zenitani, 2023a), which may not be feasible if information is not publicly released.

Finally, the manual mapping of indicators of compromises (IoC) to represent nodes is error-prone and time-consuming (Jha et al., 2002), especially when modeling complex cyberattacks over long durations. Different tactics, techniques, and procedures (TTPs) listed in whitepapers or other gray literature are often written in human language, which may make it time-consuming to extract certain key action items not related to IoCs such as hashes or domain names. Additionally, Sun (2023) conducted cyber threat intelligence (CTI) mining from a variety of datasets and identified that its high-volume nature and quality control make it challenging for organizations to improve their threat posture.

While there has been works in automating CTI text extraction for automated attack graph creation (Li, 2022; Venkataraman & Drummonds, 2000; Zhu & Dumitras, 2018), most of these

tools abstract the indicators of compromise or specific technical details to address the issue of space and presentation. The more complex a given cyberattack is, the more difficult it is to model or account for all details contained within the attack itself. Large Language Models (LLM) such as OpenAI's ChatGPT also present attractive opportunities to address the limitations of previous natural language processing techniques for accurate cyber defensive measures. There may be improved graph readability and white paper data extraction based on these advancements.

Regardless, attack graphs continue to serve as an important tool to model the chronological causal relationships between events. They allow security analysts to accomplish a wide variety of tasks, such as modeling network intrusions, summarizing and implementing proactive security defensive recommendations, and assessing a given host or network's security. While there is a rich history of attack graph research endeavors and the definitions of an attack graph's overall elements are well understood, there appears to be a gap in modeling and analyzing well-known cyberattacks, such as the SolarWinds supply chain breach. Using attack graphs to model the SolarWinds breach may help security analysts improve their overall cyber posture.

Provenance Graphs

A provenance graph is a directed graph that provides causal modeling between subjects (nodes) in relation to events or operations (edges) on a given system (Li, 2021). Because these edges are directed, provenance graphs have strong temporal and spatial properties (Li, 2021), which enhances investigation for event causality analysis. Li (2021) also emphasizes that

because provenance graphs demonstrate causality between events, they are synonymous with causality graphs. Additionally, there are also two different analysis techniques: backward tracking and forward tracking. Analyzing how a node came to existence refers to backtracking, where all the steps causally create the node. In contrast, forward tracking starts out with a node, but analyzes how this node impacts other nodes within the graph.

DARPA's Transparent Computing program aimed to provide visibility into computer systems by developing technology to detect cyberattacks via different system events (DARPA, n.d.). This program arguably catalyzed the use of provenance graphs, and several implementation prototypes have been demonstrated to achieve high detection rates (Anjum et al., 2022; Han et al., 2020; Milajerdi et al., 2019). Milajerdi et al. (2019) specifically generates a high-level graph for analysts to respond to cyberattacks, where nodes represent TTPs, and edges represent the information flow between entities. Different symbols are also used to represent nodes. Furthermore, this field also showcases potential promise through its utilization of real-world datasets, thus making such graph-based applications from academia to industry valuable.

Al-Saraireh (2022) and Li et al. (2021) discuss issues and possible solutions related to provenance graphs. Notably, the lack of unified adversary datasets and data formats increases the barrier for researchers to develop novel detection solutions, as the only known, high-quality datasets are from the TC engagements. Another notable challenge is with node data reduction; generating and loading provenance graphs into a given system can be both space and time-consuming, as they are typically implemented with databases (Li et al., 2021; Xie et al., 2013). Research endeavors focus on compression algorithms, which typically remove unnecessary nodes and their corresponding edges, but arguably, the removal of information usually comes with the removal of causality.

Another challenge is the literature regarding the ineffectiveness of provenance graphs. Mimicry attacks in integrate benign subgraph(s) within the main provenance graph, such that a greater chance is derived for the detection ratings to move closer to the false positive decision boundary (Goyal et al., 2023). This attack method boasts a 100% success rate against different provenance-based detection systems. Coe (2014) also highlights that the way data is captured directly impacts how provenance graphs are created – in some cases some nodes are missed – thus calling for the use of different capture agents to increase overall coverage.

While provenance graph research arguably represents the next generation of detection technology for their ideal methods for modeling cyberattacks (Li et al., 2021), they may not be relevant for this thesis because of their focus as detection technology. Attack graphs can arguably be thought to offer visual representations of a cyberattack and aid security analysts in tasks, such as network hardening or cyberattack perception (Pirca & Lallie, 2023). Furthermore, as indirectly demonstrated with systems like HOLMES through high-level scenario graphs (Milajerdi et al., 2019), attack graphs could be used to represent attacks within a host or network, which may suggest that the two complement each other by serving different strategic purposes.

Industry Frameworks

There are several industry frameworks for modeling cyberattacks. Two of the most notable include the Lockheed Cyber Kill Chain and the MITRE ATT&CK matrix. Both model the techniques and procedures typically taken by adversaries and aid security analysts in mitigating cybersecurity threats. One key difference between the two is that the Cyber Kill Chain

focuses on a general sequence of events, while the MITRE ATT&CK matrix focuses on organizing TTPs into a table-like format.

While attack graph nodes also employ various TTPs listed in the attack framework and demonstrate strong temporal relationships similar to the Kill Chain, they differ in that attack graphs focus on modeling a specific campaign where an adversary chains together system events to form an attack (Mell & Harang, 2015); in contrast, the Kill Chain focuses on providing a high-level framework of the different stages of a cyberattack, and does not showcase technical events or details by itself. While the ATT&CK matrix does showcase TTP details through the corresponding techniques page, the level of specificity is often found within the corresponding tables, rather than immediately.

There is also research on how well attack modeling techniques (AMTs) aid in communicating cyberattacks. Pirca & Lallie (2023) recruited 157 participants, ranging from computer science students to C-suite executives, and revealed that not only did participants overwhelmingly prefer attack graphs when understanding different cyberattacks, they also performed empirically better when questioned on topics such as attack identification. This highlights the importance of information presentation especially when it comes to post-attack analysis. Moreover, while the ATT&CK matrix may provide more detail than attack graphs, it requires more time for analysis. In contrast, the information in attack graphs may be reduced for visualization, but ultimately help individuals understand complex cyber threats by providing a high-level, yet detailed, overview of how the attack executes and traverses across a given system. Pirca & Lallie (2023) also demonstrates how attack graphs are a growing area of research popular in academia, while the MITRE ATT&CK matrix is popular in industry. Exploring the

potential synergy between these two frameworks by leveraging their respective strengths may improve visibility and analysis.

The Diamond Model of Intrusion Analysis models four key components (adversary, capability, infrastructure, victim) under a relationship of a diamond shape, which uses meta-features to capture all aspects of a malicious event. It is a valuable tool for threat intelligence and complements the Cyber Kill Chain (Caltagirone et al., 2013) by providing granularity among complex relationships of events, but may not be relevant for this thesis as attack graphs are more suitable for visualizing paths within a given system or network.

Chapter 3

SolarWinds Supply Chain Breach Attack Graph

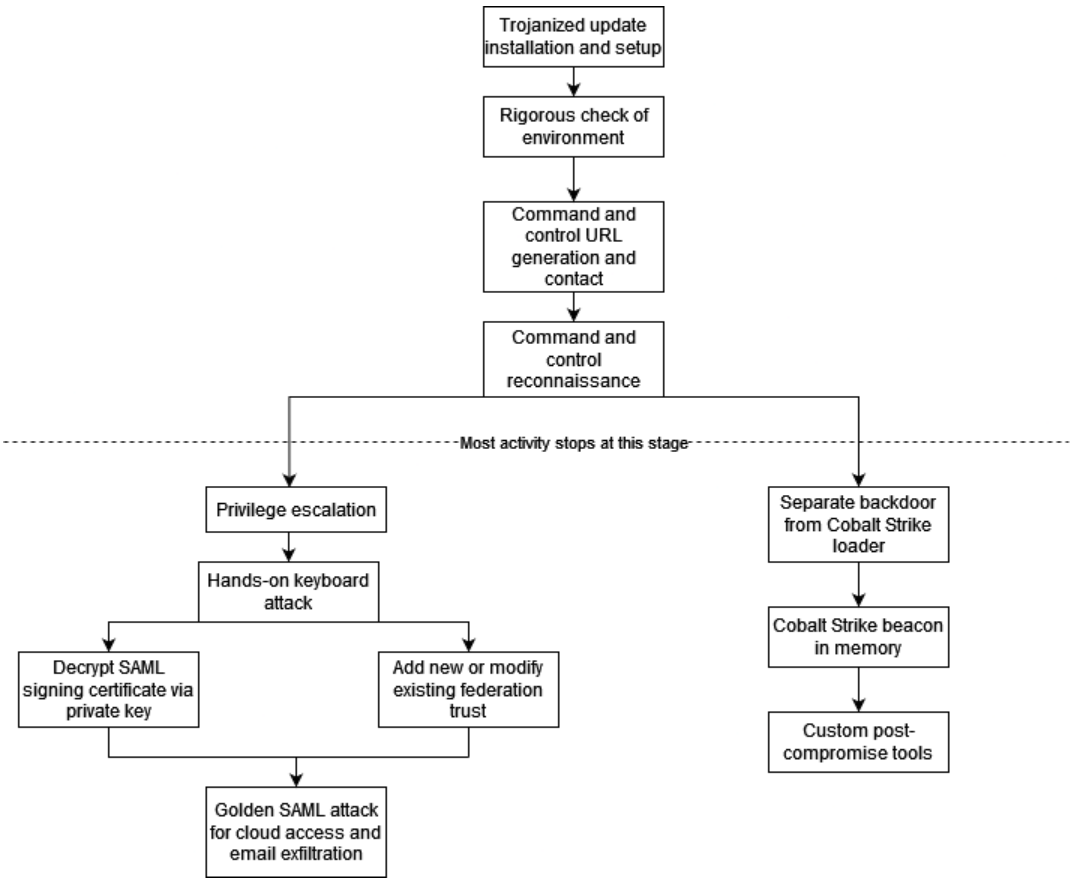


Figure 1: Overview of SolarWinds breach attack graph

Figure 1 illustrates a high-level overview of the attack graph, starting with the root node being the installation of the compromised SolarWinds Orion platform DLL (Microsoft Threat Intelligence, 2020a) and the end nodes typically representing the exfiltration of data. Nodes with high granularity are omitted for presentation purposes, as presenting the full attack graph negatively impacts visualization. This notion is expanded on in the limitations section.

Chapter 4

Causality Relationships between Attack Activities

This section aims to present the analysis of the generated attack graph on the SolarWinds breach through the various firsthand reports on technical analysis. Subgraphs and corresponding analysis are analyzed and used.

Attack Graph Definition of the SolarWinds Attack

The SolarWinds supply chain attack can be defined as an attack graph containing a series of ordered nodes defined by time. Figure 2 describes a snippet of an attack graph modeling the SolarWinds breach, which contains two main parts: nodes and relationships. Nodes represent computer events of different scales of granularity. Depending on granularity, some nodes represent simple events, such as the use of an XOR algorithm, while other nodes represent complex events, such as retrieving information from various sources to send to a command and control (C2) server. As commonly described in attack graphs, these nodes are organized by time, and thus, the previous nodes must have passed for the next to execute. Edges represent a relationship between node(s), which can be functions that are, but not limited to, reading, writing, or executing functions.

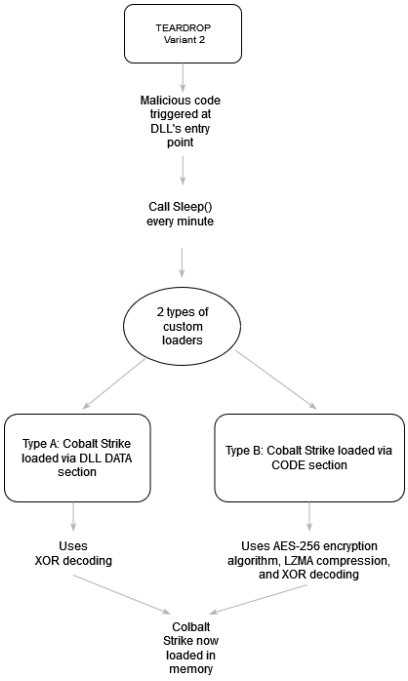


Figure 2: Snippet of attack graph format

Several nodes are organized and grouped into rectangles with a corresponding label at the top left. For example, the installation phase in Figure 3 presents how the backdoor requires several nodes to pass for the next node to execute. If any of the nodes fails, the attack does not continue. Subsequently, Figure 4 highlights the domain generation algorithm (DGA) group, which showcases three nodes that are responsible for the generation of a unique string of characters to be used for the subdomain of a C2 server. In the event the backdoor fails to retrieve or in any way cannot access these values, the C2 server may not respond properly, causing the attack path to deviate away from the created attack graph.

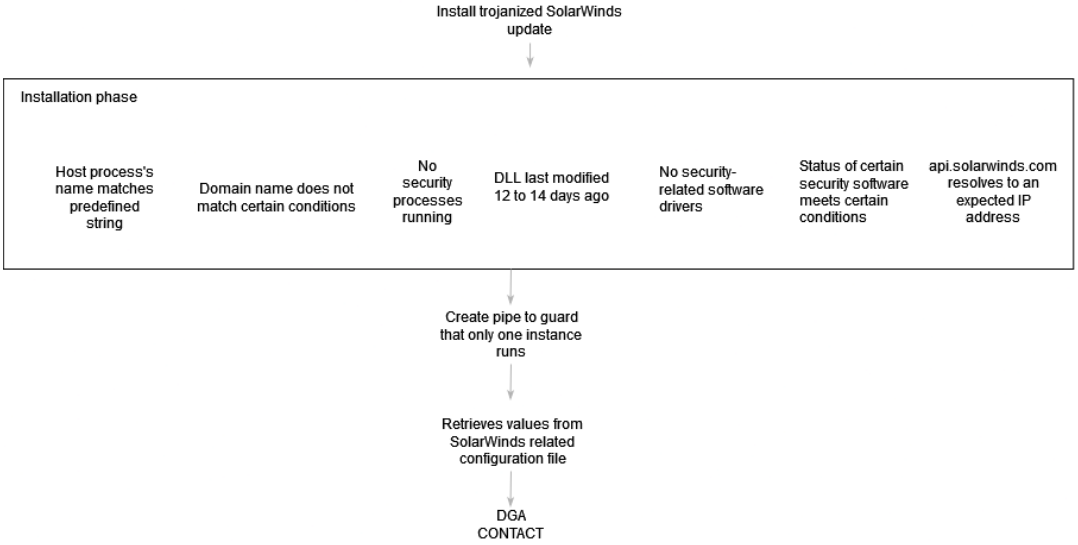


Figure 3: Initial installation of trojanized update

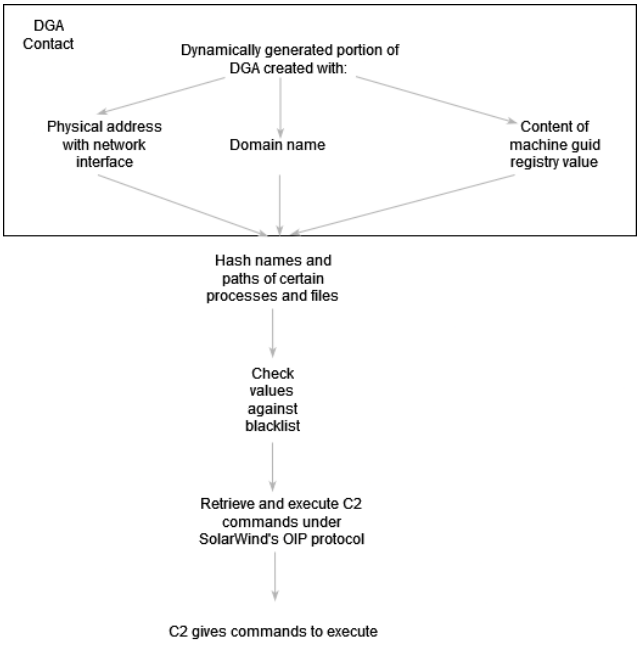


Figure 4: Dynamic generation of C2 URI

Furthermore, the adversaries used different variants of the same phase to progress to the same objective. These are represented with rounded rectangles, such as the one showcased in Figure 2. This is demonstrated with TEARDROP, where the adversaries were careful in

separating the SolarWinds backdoor from the Cobalt Strike beacon by using two variant loaders to load Cobalt Strike to memory (CDOC & Microsoft Threat Intelligence, 2021).

As a result, the SolarWinds breach is comprised of events, which are represented as nodes and organized such that each node is either a precondition for subsequent node(s), or a post-condition result from preceding node(s). Furthermore, a collection of nodes, or subgraphs, can be a precondition for two or more subgraphs.

Summary and Analysis

The genesis node of the attack begins with the installation of the trojanized SolarWinds update (Microsoft Threat Intelligence, 2020a), as showcased in Figure 3. Several nodes follow, which all primarily perform checks to ensure that the malware is not running in a test environment. Once these nodes have passed, the next stage of the attack begins. If any of these conditions are not met, the program terminates (Microsoft Threat Intelligence, 2020a).

The sample then follows a relatively linear set of nodes to ensure smooth operation. FireEye (2020) and Microsoft Threat Intelligence (2020a) present detailed analysis relevant to the early execution of the SolarWinds breach. The first node creates a pipe to guard that only one instance is running before reading a configuration file from disk and retrieving the XML field appSettings (FireEye, 2020). Such reading ensures that the malware operates as intended with the desired parameters.

The extensive use of blacklists against security related software is well documented (Check Point Research, 2020), however, such circumventions rooted from a newly installed program showcase an important precondition for the attack to continue, as ideally such analysis

directly addresses the question of the SolarWinds detection problem. Potentially, by quarantining or monitoring such calls – especially for newly installed programs or updates, – the toolkits for detecting such tactics can be strengthened.

More interestingly, after a relatively long period of delay, the sample then begins to check-in with a C2 server to send basic information about the environment and receive commands (Microsoft Threat Intelligence, 2020a). From this subgraph, we can derive that the postcondition of the installation of the trojanized update is the actual contact with the external server, and such postcondition occurs after blacklist verification. However, Figure 4 showcases how the adversaries employ unusual tactics to create the C2 domain by gathering information from a diverse range of sources, which is then used to dynamically generate the actual URI. The DGA contact group reveals the three core nodes used: the physical address of the NIC, the domain name, and the content of the MachineGuid registry value (Microsoft Threat Intelligence, 2020a).

When these conditions are met, the sample generates the domain and contacts the C2 server. If a response is received, the connection is successful, and the adversaries move to the next stage by performing hands-on keyboard attacks (Microsoft Threat Intelligence, 2020a). Most activities seen on compromised networks typically do not advance past reconnaissance, highlighted by the fact that fewer than 100 networks of the 18,000 total customers were impacted by SUNBURST (Ramakrishna, 2021c).

As a result, the main requirements can be derived from the beginning subgraph composing of the nodes from the installation of the trojanized SolarWinds update to the preliminary information sent to the C2 server. This subgraph represents the key prerequisite for the rest of the attack to continue. Additionally, the precondition subgraph splinters in two post-

conditional attack subgraphs: one aims to achieve off-premises access to cloud resources through an attack vector known as a Golden SAML attack, while the other attempts to load a Cobalt Strike beacon. This is a result of the attack graph's ability to persevere temporal relationships, and as such, two subgraphs can be derived from the main graph. Security analysts can learn from the attack graph's representation to view the attack from a different perspective through this rearrangement of data.

Further analysis reveals the first subgraph, presented in Figure 5, primarily focuses on forging a SAML token used to access resources. To achieve this, the adversaries aim to gain persistence on the system with various techniques; Microsoft Threat Intelligence (2020a) outlines the use of Powershell and rundll32.exe, with specific commands showcased.

Because persistence is achieved, the adversaries shift to post-conditional nodes of hands-on keyboard attacks to "obtain domain permissions" via on-premises intelligence gathering, lateral movement, and credential dumps (Microsoft Threat Intelligence, 2020b). These precondition nodes are required for the adversaries to acquire valid SAML tokens through two main methods: (1) add their own certificate(s) as trusted entities by adding or modify existing federation trust properties; or (2) steal the SAML signing certificate to sign their own SAML token (Microsoft Threat Intelligence, 2020b). Both paths forge valid SAML tokens to establish persistence, access cloud resources, or exfiltrate email.

While the abuse of Powershell and rundll.exe are not new techniques, they still represent the challenge of adversaries "living off the land" of binaries (Lenaerts-Bergmans, 2023a). By using existing tools on a given system, adversaries avoid the need to request and install malicious code, thus avoiding potential traffic or antivirus programs detecting malicious signatures. Lambert (2021) highlights how traditional IoCs, such as file names or hashes are too imprecise;

Lenaerts-Bergmans (2023a) instead argues for the shift away from traditional IoCs, where forensic artifacts of adversary activity are detected, and rather to IoAs (indicators of attack), which uses dynamic combinations of behaviors within a system to determine a breach in confidentiality.

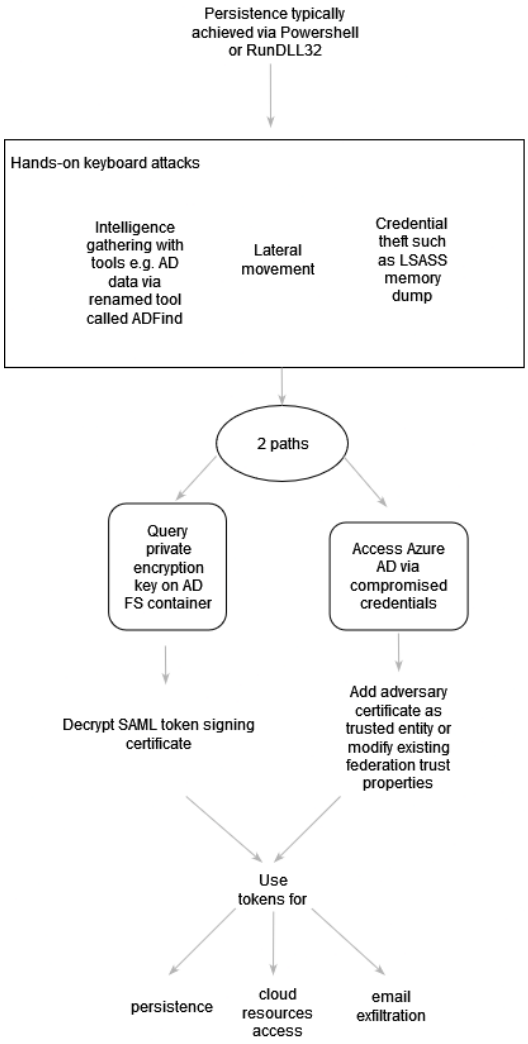


Figure 5: First subgraph detailing SAML abuse (Golden SAML attack)

Analysis of the second subgraph shifts to the installation of additional malware, specifically Cobalt Strike and custom malware for compromised networks. After the C2 server responds with encoded commands, the backdoor then begins the second-half portion of the attack to separate the SolarWinds backdoor to the execution of Cobalt Strike. Two key critical nodes

are in the form of files created by the backdoor: a VBScript and a DLL containing the Cobalt Strike beacon. An IEFO registry value triggers an executable (wscript.exe), which runs said VBscript, which finally runs the malicious DLL (CDOC & Microsoft Threat Intelligence, 2021).

Depending on the network or company in question, there are several known custom Cobalt Strike DLLs that load Cobalt Strike into memory. CDOC & Microsoft Threat Intelligence (2021) describe two main variants as showcased in Figure 6. The first outlines two DLLs that contain an export function that spawns a new thread and triggers malicious code in said thread. The code checks a variety of data, such as image files and registry values before decoding and running the Cobalt Strike loader (CDOC & Microsoft Threat Intelligence, 2021). The second variant, however, directly runs the malicious code from the DLL's entry point, which CDOC & Microsoft Threat Intelligence (2021) further break down into two types of DLLs that either decode and run Cobalt Strike from the DLL's DATA or CODE section. Both ultimately result in the execution of the Cobalt Strike Reflective Loader. Symantec also describes a third variant dubbed RAINDROP, which is a DLL built on 7-Zip source code. Upon loading, the DLL starts a new thread, performs unrelated tasks, finds the start of the encoded malicious code, and subsequently decodes and executes the malicious code (Threat Hunter Team, 2021).

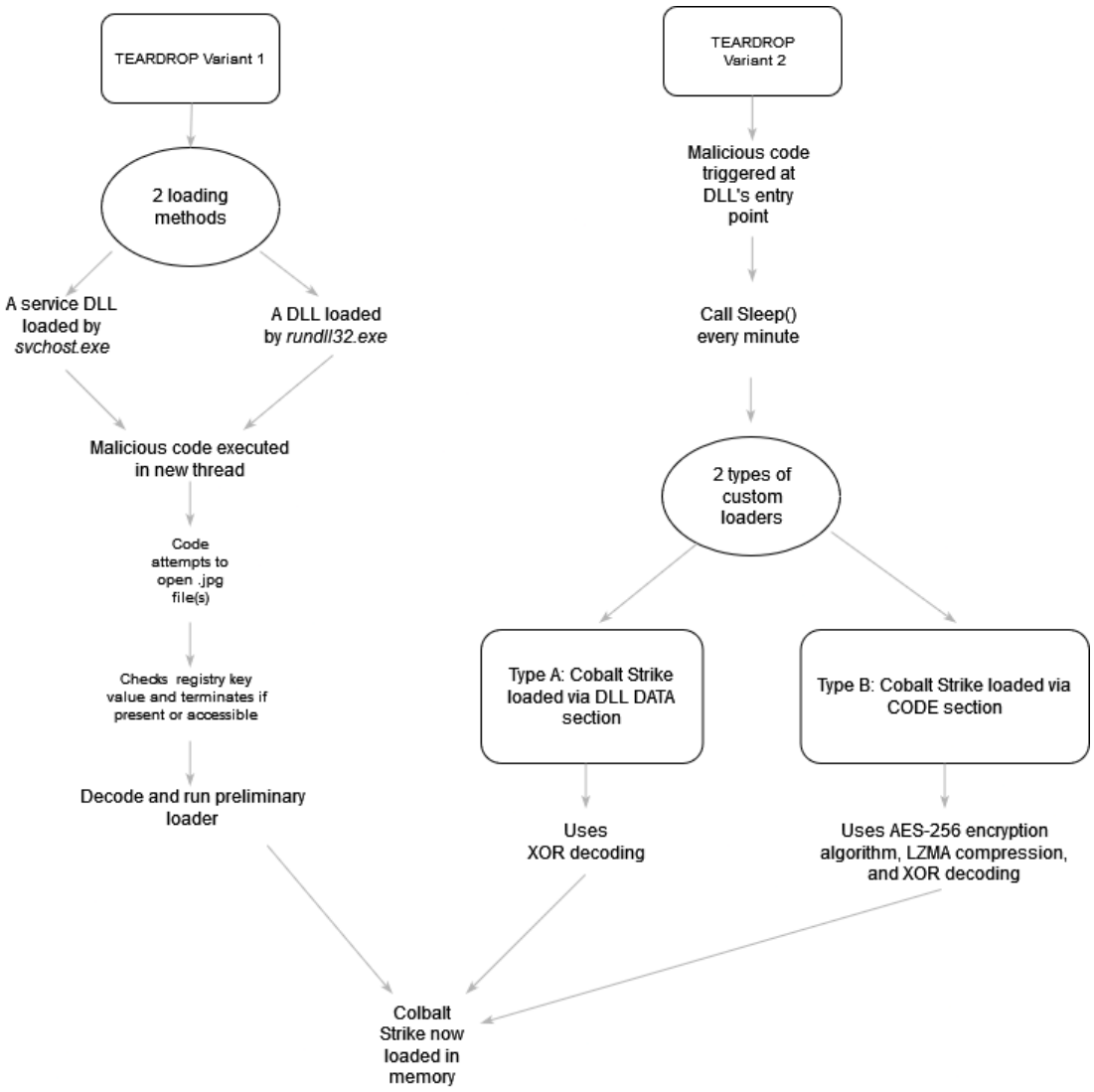


Figure 6: Cobalt Strike loading into memory

Attack graph analysis showcases an interesting theme through the use of decryption, decompression, and obfuscation methods to use the malicious code when needed. While omitted in the graph for presentation purposes, monitoring the use of these decoding or decryption algorithms, and more importantly, the results of said methodologies, may be fruitful for detection purposes.

The last stages of the attack outline several tools discovered during the late stages of the SolarWinds breach, after the adversaries gained access “through compromised credentials or the

SolarWinds binary, and after moving laterally with TEARDROP" (Ramin, et al., 2021). These ultimately represent post-conditions of the prerequisite subgraph of the Cobalt Strike beacon installation. Ramin, et al. (2021) describes how GoldMax (malware for additional C2 communication) undergoes several stages to establish and receive instructions from a server. During the setup process, GoldMax checks for certain values and terminates if unable to do so. If these stages pass, it then sets up a configuration file used to set up its runtime environment, using embedded values or values that an operator can adjust. Within the file is an activation date, and if the times match, it then establishes a session with the C2 server to get a session key, and all commands and responses are encrypted with said session key.

Prerequisite analysis during the end stages of an attack may be challenging, as adversaries typically have privileged access to a wide variety of systems or networks, and resulting nodes are post-conditions as a result. However, a common theme, as seen in GoldMax and RAINDROP, is the use of delayed execution, whether from the use of the sleep() function, dummy computations, or other means of delay. MITRE ATT&CK lists these tactics as T1497.003, which are known techniques commonly used to circumvent automated antivirus detection.

Ramin, et al. (2021) also describes Sibot, a VBScript file that runs as a task. It reaches out to a C2 server to download a DLL to a folder under System32, which is then run by rundll32.exe. Three obfuscated variants are revealed in Figure 7: the first installs the script under the registry key value, the second runs a Powershell command daily, and the third is a standalone version that runs as a file. From there, the script reads certain values on the machine before establishing a connection to the C2 server, where data about the given system is sent.

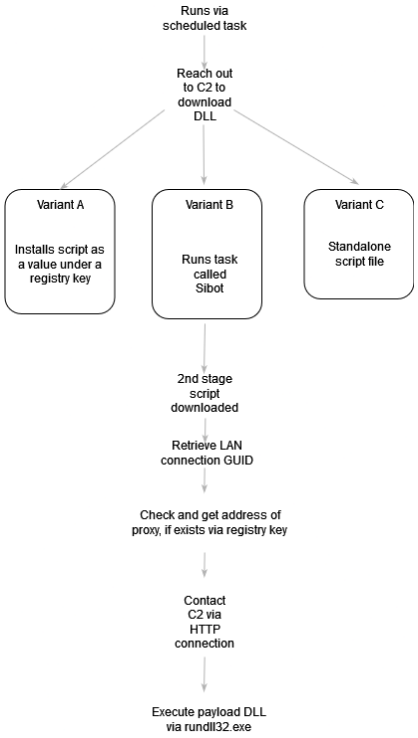


Figure 7: Attack subgraph of Sibot

GoldFinder is aimed to identify HTTP proxy servers and redirectors that would hinder C2 communication. It issues an HTTP request to a hardcoded IP address and logs responses into a log file (Ramin, et al., 2021). Figure 8 displays the relatively straightforward execution path of the malware.

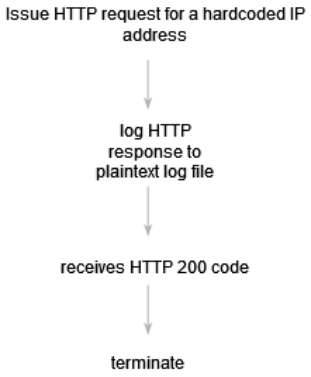


Figure 8: Attack subgraph of GoldFinder

Existing literature has focused on the prevention or detection of malware samples contacting C2 servers (Davanian et al., 2021; Villeneuve et al., 2012; Zhao et al., 2015), and the attack graph points to the same conclusion. As an example, the adversaries use a sophisticated command and control domain generation algorithm (DGA) to send and receive commands for network footprinting. The groups of nodes are highlighted in Figure 4. Without the successful retrieval of these node's events, the contact stage would fail, and thus the attack progress would be delayed. The adversaries have also demonstrated considerable efforts in separating the backdoor from the trojanized update to maintain their foothold in the network, emphasizing the initial subgraph as a core precondition that kick-starts the rest of the attack. These nodes are showcased under the Cobalt Strike section, where the use of both TEARDROP and RAINDROP were used on hosts as a loader for the Cobalt Strike Beacon and further hands-on keyboard attacks (Threat Intelligence Team, 2021).

Additionally, there are several individual nodes that must occur for attack progression. One such node is the root node: the installation of the trojanized SolarWinds update. Its relationship with the remaining nodes represents the relationship of the graph. However, it is also important to note that some of these – and other – potential nodes are omitted due to three main reasons: (1) atomic attributes: some nodes and corresponding details are too granular and may not be relevant for broader visualization purposes; (2) space complexity, as such precise details may not help security analysts or other individuals understand the attack; attack graphs are best used to present a high-level overview of a complex cyberattack to individuals, such as security analysts or high-level decision makers (Pirca & Lallie, 2023); (3) unclear edge relationships, as there are subgraphs or fragmented graphs that have unclear temporal relationships with the main graph, thus creating unclear edge relationships.

Utility of the Attack Graph for Security Analysts

Attack graphs ultimately help security analysts by providing a visual representation of the series of ordered nodes taken during a cyberattack. For instance, Li et al. (2016) highlights that attack graphs assist in identifying the most vulnerable resources and overall risks within a given network. Jha et al (2002) also emphasizes how attack graphs can form the "basis for detection, defense, and forensic analysis," such that analysts may opt to redirect focus and resources on critical nodes that may lead to significant changes and disruptions.

Subsequently, analysis for the attack graph showcases four key resources: network access, Powershell and rundll32 use, access to decoding, decryption and deobfuscation functions, and use of extensive blacklist checks. Network access is critical for communication and control, as many malware samples require the use of hands-on keyboard attacks to progress further into the network. Powershell and rundll32.exe usage are also common methods to abuse legitimate applications for illegitimate purposes, as well as deriving clean process trees for malware usage. Monitoring deobfuscation methods, such as hashes and encryption methods, especially the eventual plaintext results loaded, can reveal critical information about the functionality of malicious code. Finally, the extensive query and comparison function calls for blacklists may reveal anti-forensic methods being used.

Attack graphs, as demonstrated within this paper, also aim to integrate the wide variety of threat intelligence reports into one cohesive picture. Ren et al. (2022) reiterates this idea by emphasizing on how knowledge graphs change the expression of threat knowledge for accurate decision making, and that the timely sharing of CTI information means shorter response times. Specifically, by combining intelligence reports into a cohesive picture, attack graphs can break down complex ideas into presentable or understandable formats (Ren et al., 2022). Furthermore,

attack graphs may play a vital part in judgment and informed decision making on the allocation of resources and for viewing the organization's overall security posture.

Building on this, there is research showcasing the efficacy of attack graphs in communicating cyberattacks. Attack graphs enhance attack perceptions, such that individuals tend to perform empirically better when analyzing what happened in a cyberattack when compared to the MITRE ATT&CK matrix (Pirca & Lallie, 2023). Arguably, by providing a high-level overview of how the attack executes and traverses across a given system, the decreased time for analysis and ease of understanding for what the cyberattack entailed, security analysts are better equipped to identify the overall risk of a given network (Li et al., 2016). Lallie, et al. (2020) also focuses on the importance of improving cyber security perception and usability in systems, such as attack graphs, arguing that such efforts reduce the difficulty of understanding complex attack patterns.

Chapter 5

Discussions and Limitations

This research aimed to use attack graphs as an attack modeling tool to model the SolarWinds supply chain attack via threat intelligence reports. Analysis was performed on the graph, with a focus on temporal organization and prerequisites required. However, there are several limitations to both the construction of the attack graph and its analysis.

Graph Representation

The main limitation regarding the attack graph format is its size, such that the full representation of the graph cannot be presented. As the breach consists of a multitude of events and attack patterns, and more than 100 indicators of compromise were derived and organized, formatting such events into a model discernable at a glance is challenging. While the paper addresses this via snippets of the graph and presents analysis through text, this may not be feasible for using attack graphs for other analysis situations.

Large attack graphs are a known problem within attack graph research (Al-Araji et al.,2021; Mell & Harang, 2015). Zenitani (2023a) highlights the scalability problem by demonstrating a small network with n computers with m vulnerabilities, such that each vulnerability can be exploited or not, meaning that there are 2^{mn} possible network states. Li et al. (2016) also highlights the challenge that large-scale networks increase overall complexity due to their large number of hosts and attack paths, which makes analysis challenging as there is too much information. Even so, there are several surveys outlining research works (Lallie et al., 2020; Li et al., 2021; Zeng et al., 2019; Zenitani, 2023a) attempting to reduce the amount of

information on attack graphs. Future research into presenting and analyzing the efficacy of large attack graphs in such formats may offer valuable insights into their communication benefits.

Imperfect Information

An accordingly surplus of detailed first-hand account cyber threat intelligence reports exists, making it challenging to gain a comprehensive picture while also balancing the formatting of the attack graph. Combined with the manual extraction of events and node organization, there is bound to be missed information, either through the omission of granular nodes, the absence of key gray literature or other misinterpretations of the adversary's intricate maneuvers throughout the campaign.

More interestingly, the research work is limited to the number of whitepapers that publicly exist. As the ratio between total SolarWinds customers and actual organizations breached by the adversaries is small, different attack patterns occurring in one organization may not be applicable to another organization. Microsoft Threat Intelligence (2020b) sheds light on this idea in how the attack, particularly in its late-stage activity, is customized, thus reaching a point where the attack trajectory diverges. As a result, subgraphs occurring in the later stages of the attack graph exhibit unclear temporal relationships with its surrounding context, as the precise order for where or how these tools were integrated are ambiguous. Thus, the attack graph's applicability for specific organizations diminishes further down each attack path.

One issue discussed earlier in the paper is the idea of how detailed a node should be, as the more granular a node is, the more nodes are required to represent an event. Omitting certain events from the attack graph may improve clarity, but also contribute to gaps in an attack path, or

weak causality relationships between nodes. Furthermore, even with highly granular events included, the efficacy of the attack graph is best when presented to security analysts as a high-level overview (Pirca & Lallie, 2023).

Poor Representations for Alternative Flows

The generated attack graph generally provides a single, static view of an attack, however, there may be similar attack graphs that model the same breach but differ in specific subgraphs. This is best demonstrated with GoldMax during the file setup process (Ramin, et al., 2021). During the initial execution, GoldMax follows a set of instructions that are then abandoned upon subsequent executions. Factors such as loops, pauses, and if-else statements are inaccurately represented or not adequately captured in the attack graph, however, such conditions are common within malware analysis. Thus, attack graphs, similar to the Kill Chain, can assume that a linear set of nodes are taken to eventually reach a desired state, and that such a set is universal for any system or network. However, this assumption overlooks the nuanced reality where multiple attack paths may be achieved.

Future Research

As emphasized in existing literature and demonstrated in this paper, researching attack graph standards and their effect on cyber perception can be a fruitful endeavor for risk communication and cyber posture analysis. There also exists little integration with industry frameworks, such as the MITRE attack matrix or the Kill Chain compared to attack graphs, however, there may exist benefits when merging these AMTs. Lastly, this paper does not attempt

to survey the communication efficacy of the created attack graph to other individuals. While existing research agrees that attack graphs have scalability concerns, there is sparse research on modeling complex cyberattacks with attack graphs, or testing said graphs against various stakeholders. Such concerns may be outweighed by potential benefits.

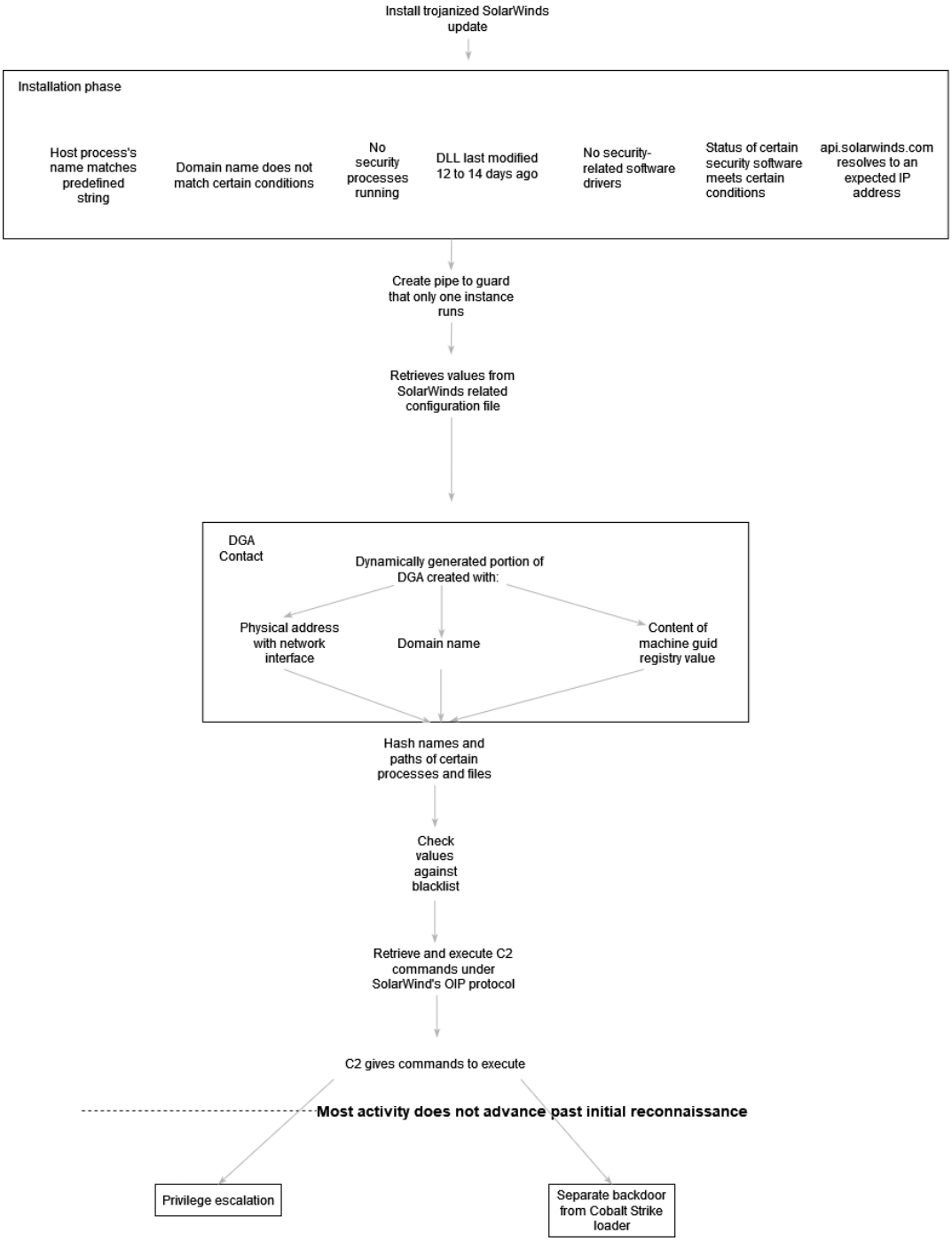
Chapter 6

Conclusion

The SolarWinds supply chain breach demonstrated the significant risks third-party software has on organizations. While the security industry witnessed the rapid and open sharing of information for reactive cyberdefense, they underscore the need for preventative measures to mitigate – or block – such incidents in the future. As demonstrated in this thesis, attack graphs showcase a promising approach for identifying root causes of security incidents. Its strong temporal and causal relationships mapping via critical nodes, combined with its enhanced visual representations in assessing cyber threats and network defense postures may bring insight into what resources and time security analysts should dedicate to. The major contributions to the surrounding literature include modeling the breach as an attack graph through a synthesis of indicators of compromise. Subsequent analysis was shown via critical nodes to shed light into novel conditions on progression. Future research work may include standardizing novel formatting and visualization techniques as well as surveying the impact graph scalability has on security analysts.

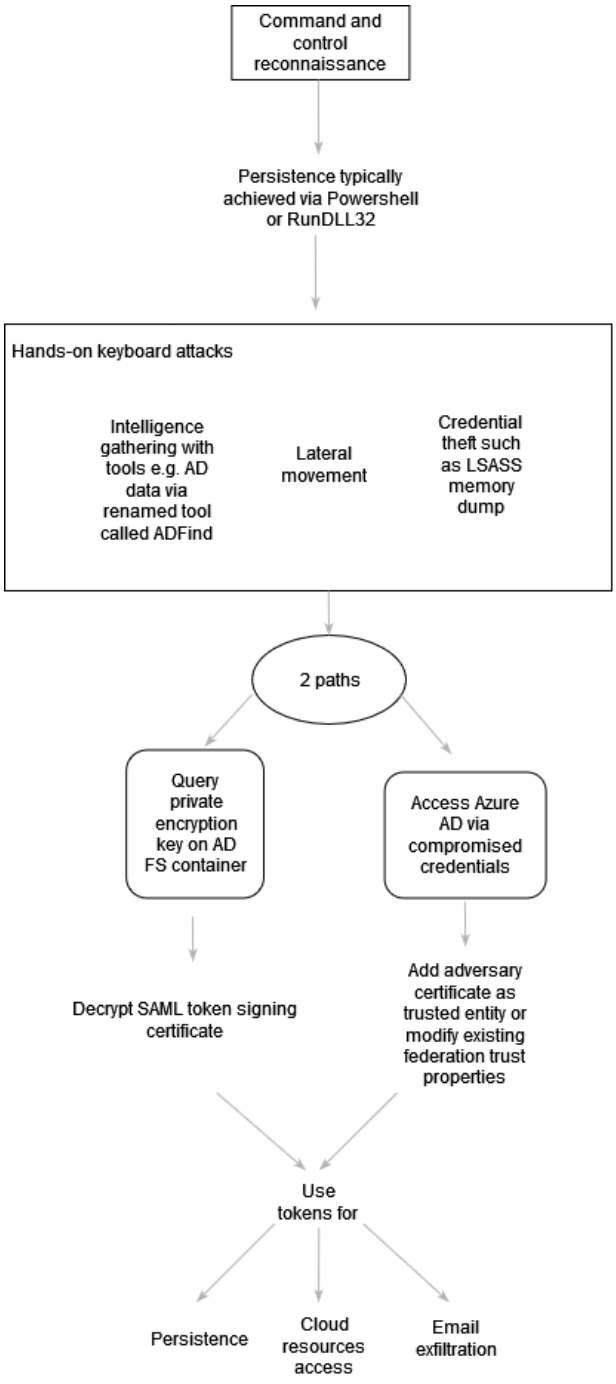
Appendix A

Initial Graph: Trojanized Update Setup



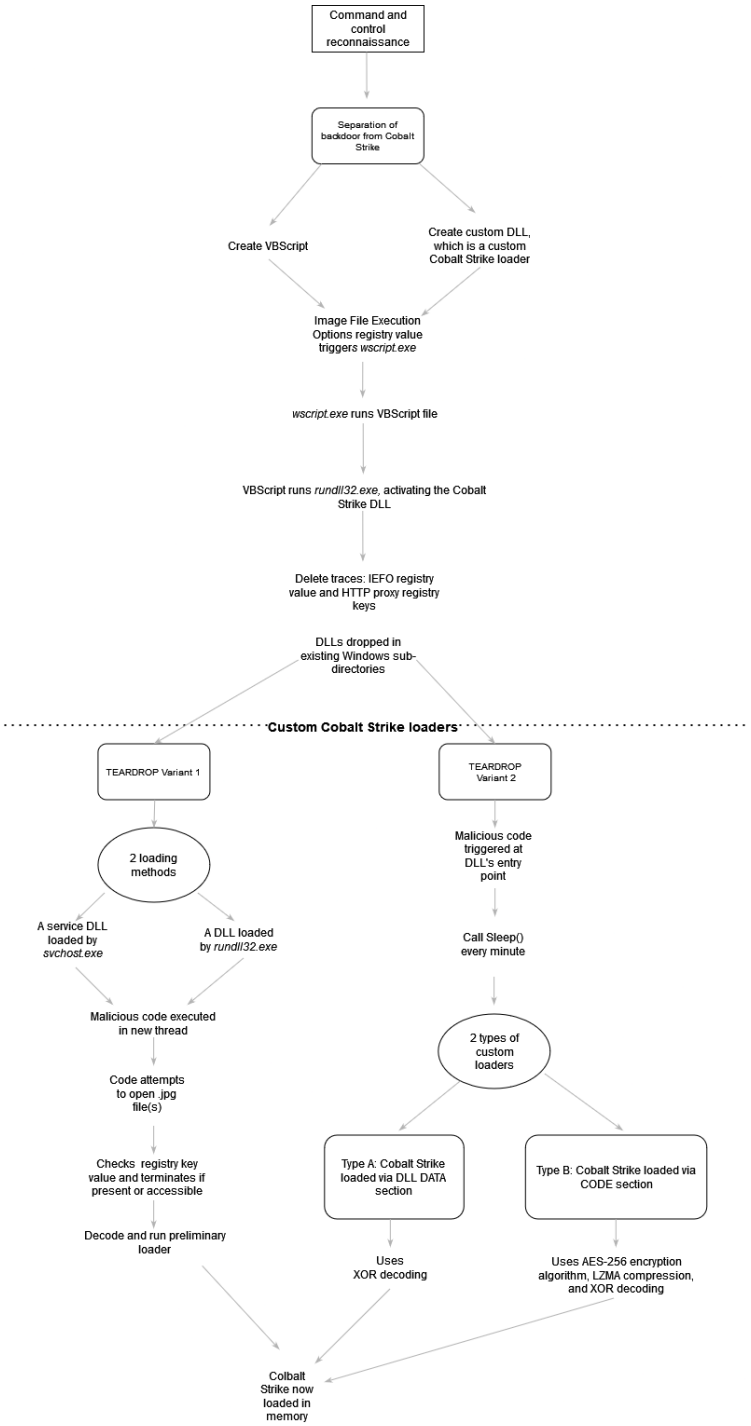
Appendix B

Subgraph 1: SAML Attack



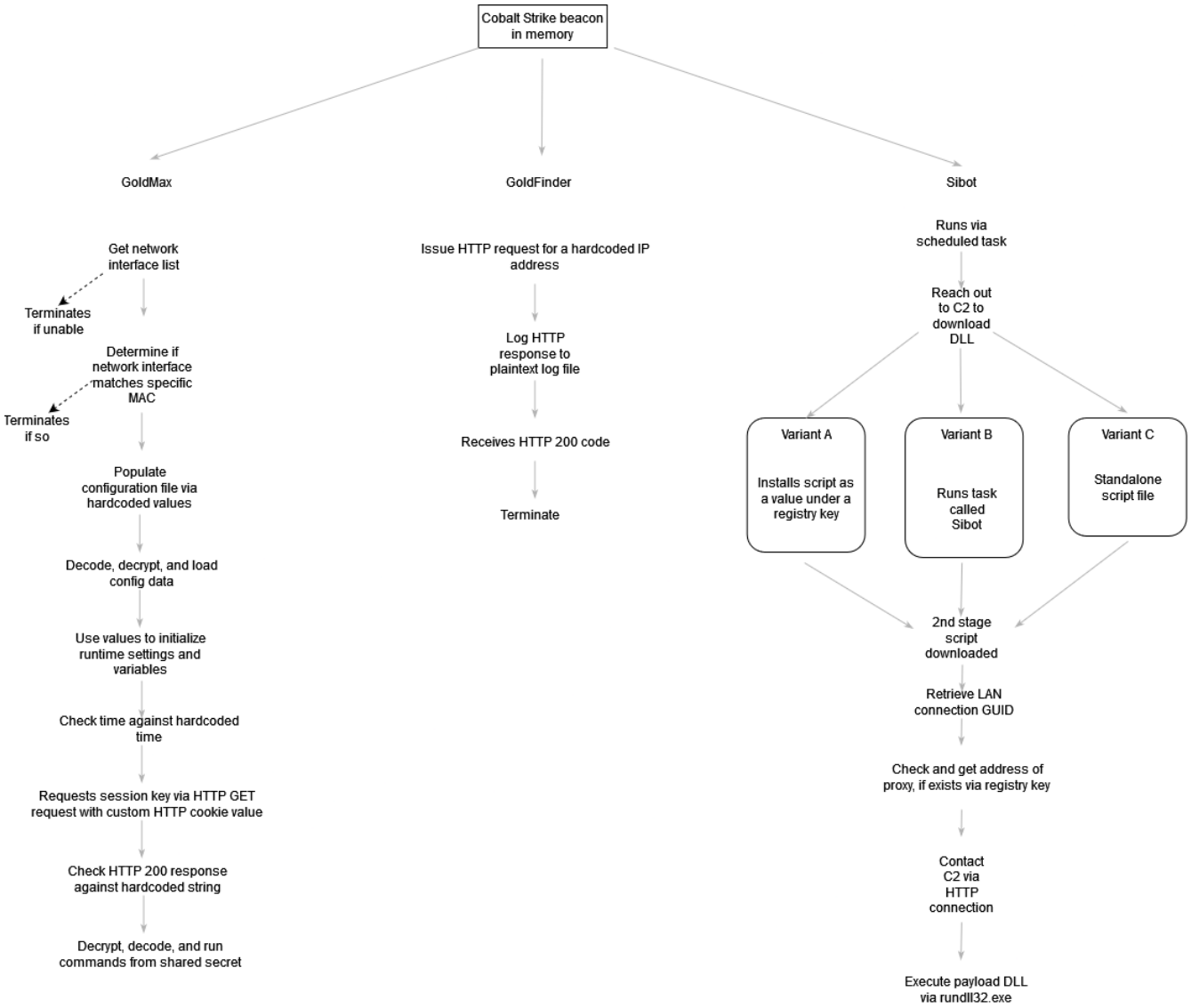
Appendix C

Subgraph 2a: Backdoor Separation



Appendix D

Subgraph 2b: Late-Stage Custom Tools



BIBLIOGRAPHY

Al-Araji, Z. J., Ahmed, S. S. S., Abdullah, R. S., Mutlag, A. A., Raheem, H. A. A., & Basri, S. R. H. (2021). Attack graph reachability: concept, analysis, challenges and issues. *Network Security*, *2021*(6), 13-19.

Al-Saraireh, J. (2022). A novel approach for detecting advanced persistent threats. *Egyptian Informatics Journal*, *23*(4), 45-55.

Anjum, M. M., Iqbal, S., & Hamelin, B. (2022, April). ANUBIS: a provenance graph-based framework for advanced persistent threat detection. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (pp. 1684-1693).

Baker, P. (2021, June 4). *The SolarWinds hack timeline: Who knew what, and when?* CSO Online; IDG Communications. https://www.csoonline.com/article/570537/the-solarwinds-hack-timeline-who-knew-what-and-when.html

Caltagirone, S., Pendergast, A., & Betz, C. (2013). The diamond model of intrusion analysis. *Threat Connect*, *298*(0704), 1-61.

Cash, D., Meltzer, M., Koessel, S., Adair, S., Lancaster, T., & Volexity Threat Research. (2020, December 14). *Dark Halo Leverages SolarWinds Compromise to Breach Organizations*. Volexity. https://www.volexity.com/blog/2020/12/14/dark-halo-leverages-solarwinds-compromise-to-breach-organizations/

Check Point Research. (2020, December 22). *SUNBURST, TEARDROP and the NetSec New Normal*. Check Point Research; Check Point Software Technologies . https://research.checkpoint.com/2020/sunburst-teardrop-and-the-netsec-new-normal/

Cimpanu, C. (2020, December 14). *SEC filings: SolarWinds says 18,000 customers were impacted by recent hack*. ZDNet. https://www.zdnet.com/article/sec-filings-solarwinds-says-18000-customers-are-impacted-by-recent-hack/

Cimpanu, C. (2021, April 15). *SolarWinds hack affected six EU agencies*. The Record; Recorded Future News. https://therecord.media/solarwinds-hack-affected-six-eu-agencies

Coe, G. B., Doty, R. C., Allen, M. D., & Chapman, A. (2014). Provenance capture disparities highlighted through datasets. In *6th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2014)*.

Cohen, I. (2023, December 23). *ITAYC0HEN/SUNBURST-Cracked*. GitHub. https://github.com/ITAYC0HEN/SUNBURST-Cracked

CrowdStrike Intelligence Team. (2021, January 11). *SUNSPOT: An Implant in the Build Process*. CrowdStrike. https://www.crowdstrike.com/blog/sunspot-malware-technical-analysis/

Dacier, M., Deswarte, Y., & Kaâniche, M. (1996). *Models and tools for quantitative assessment of operational security* (pp. 177-186). Springer US.

DARPA. (n.d.). *Transparent Computing (Archived)* . DARPA. Retrieved April 2, 2024, from https://www.darpa.mil/program/transparent-computing

Davanian, A., Darki, A., & Faloutsos, M. (2021). CnCHunter: An MITM-approach to identify live CnC servers. *Black Hat USA*.

Eckels, S., Smith, J., & Ballenthin, W. (2020, December 24). *SUNBURST Additional Technical Details*. Mandiant. https://www.mandiant.com/resources/blog/sunburst-additional-technical-details

FireEye. (2020, December 13). *Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor*. Mandiant. https://www.mandiant.com/resources/blog/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor

Goyal, A., Han, X., Wang, G., & Bates, A. (2023, February). Sometimes, you aren't what you do: Mimicry attacks against provenance graph host intrusion detection systems. In *30th Network and Distributed System Security Symposium*.

Han, X., Pasquier, T., Bates, A., Mickens, J., & Seltzer, M. (2020, February 23). *Unicorn: Runtime Provenance-Based Detector for Advanced Persistent Threats*. NDSS Symposium. https://www.ndss-symposium.org/ndss-paper/unicorn-runtime-provenance-based-detector-for-advanced-persistent-threats/

Jankowicz, M., & Davis, C. (2020, December 15). *These big firms and US agencies all use software from the company breached in a massive hack being blamed on Russia*. Business Insider. https://www.businessinsider.com/list-of-companies-agencies-at-risk-after-solarwinds-hack-2020-12

Jha, S., Sheyner, O., & Wing, J. (2002, June). Two formal analyses of attack graphs. In *Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15* (pp. 49-63). IEEE.

Kiuwan. (2021, January 19). *Solarwinds hack timeline*. Kiuwan. https://www.kiuwan.com/blog/solarwinds-hack-timeline

Kovacs, E. (2022, November 7). *SolarWinds Agrees to Pay $26 Million to Settle Shareholder Lawsuit Over Data Breach*. SecurityWeek; Wired.

https://www.securityweek.com/solarwinds-agrees-pay-26-million-settle-shareholder-lawsuit-over-data-breach/

Lallie, H. S., Debattista, K., & Bal, J. (2020). A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review*, *35*, 100219.

Lambert, J. (2021, November 10). *The hunt for NOBELIUM, the most sophisticated nation-state attack in history*. Microsoft Security Blog. https://www.microsoft.com/en-us/security/blog/2021/11/10/the-hunt-for-nobelium-the-most-sophisticated-nation-state-attack-in-history/

Lenaerts-Bergmans, B. (2023a, February 22). *What AreLiving off the Land (LOTL) Attacks?* Crowdstrike. https://www.crowdstrike.com/cybersecurity-101/living-off-the-land-attacks-lotl/#FilelessRansomware

Lenaerts-Bergmans, B. (2023b, September 27). *What is a Supply Chain Attack? | CrowdStrike*. Crowdstrike. https://www.crowdstrike.com/cybersecurity-101/cyberattacks/supply-chain-attacks/

Li, M., Huang, W., Wang, Y., & Fan, W. (2016, October). The optimized attribute attack graph based on APT attack stage model. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)* (pp. 2781-2785). IEEE.

Liu, C., Singhal, A., & Wijesekera, D. (2012, August). Using attack graphs in forensic examinations. In *2012 Seventh International Conference on Availability, Reliability and Security* (pp. 596-603). IEEE.

Liu, H., & Jiang, R. (2023). A Causal Graph-Based Approach for APT Predictive Analytics. *Electronics*, *12*(8), 1849.

Li, Z., Chen, Q. A., Yang, R., Chen, Y., & Ruan, W. (2021). Threat detection and investigation with system-level provenance graphs: a survey. *Computers & Security*, *106*, 102282.

Li, Z., Zeng, J., Chen, Y., & Liang, Z. (2022, September). AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports. In *European Symposium on Research in Computer Security* (pp. 589-609). Cham: Springer International Publishing.

Mell, P., & Harang, R. (2015, November). Minimizing attack graph data structures. In *The Tenth International Conference on Software Engineering Advances*.

Microsoft Cyber Defense Operations Center (CDOC), & Microsoft Threat Intelligence. (2021, January 20). *Deep dive into the Solorigate second-stage activation: From SUNBURST to TEARDROP and Raindrop*. Microsoft Security Blog; Microsoft. https://www.microsoft.com/en-us/security/blog/2021/01/20/deep-dive-into-the-solorigate-second-stage-activation-from-sunburst-to-teardrop-and-raindrop/

Microsoft Threat Intelligence. (2020a, December 18). *Analyzing Solorigate, the compromised DLL file that started a sophisticated cyberattack, and how Microsoft Defender helps protect customers*. Microsoft Security Blog. https://www.microsoft.com/en-us/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/

Microsoft Threat Intelligence. (2020b, December 28). *Using Microsoft 365 Defender to protect against Solorigate*. Microsoft Security Blog. https://www.microsoft.com/en-us/security/blog/2020/12/28/using-microsoft-365-defender-to-coordinate-protection-against-solorigate/

Milajerdi, S. M., Gjomemo, R., Eshete, B., Sekar, R., & Venkatakrishnan, V. N. (2019, May). Holmes: real-time apt detection through correlation of suspicious information flows. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 1137-1152). IEEE.

MSRC. (2024, March 8). *Update on Microsoft Actions Following Attack by Nation State Actor Midnight Blizzard*. Microsoft. https://msrc.microsoft.com/blog/2024/03/update-on-microsoft-actions-following-attack-by-nation-state-actor-midnight-blizzard/

Nguyen, H. H., Palani, K., & Nicol, D. M. (2017, April). An approach to incorporating uncertainty in network security analysis. In *Proceedings of the hot topics in science of security: symposium and bootcamp* (pp. 74-84).

Phillips, C., & Swiler, L. P. (1998, January). A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 workshop on New security paradigms* (pp. 71-79).

Pirca, A. M., & Lallie, H. S. (2023). An empirical evaluation of the effectiveness of attack graphs and MITRE ATT&CK matrices in aiding cyber attack perception amongst decision-makers. *Computers & Security, 130*, 103254.

Ramakrishna, S. (2021a, January 7). *Our Plan for a Safer SolarWinds and Customer Community*. Orange Matter; SolarWinds. https://orangematter.solarwinds.com/2021/01/07/our-plan-for-a-safer-solarwinds-and-customer-community/

Ramakrishna, S. (2021b, January 11). *New Findings From Our Investigation of SUNBURST*. Orange Matter; SolarWinds. https://orangematter.solarwinds.com/2021/01/11/new-findings-from-our-investigation-of-sunburst/

Ramakrishna, S. (2021c, May 7). *FORM 8-K*. SEC. https://www.sec.gov/ix?doc=/Archives/edgar/data/1739942/000173994221000076/swi-20210507.htm

Ramin Nafisi, Andrea Lelli, & Microsoft Threat Intelligence. (2021, March 4). *GoldMax, GoldFinder, and Sibot: Analyzing NOBELIUM's layered persistence*. Microsoft Security Blog; Microsoft. https://www.microsoft.com/en-us/security/blog/2021/03/04/goldmax-goldfinder-sibot-analyzing-nobelium-malware/

Ren, Y., Xiao, Y., Zhou, Y., Zhang, Z., & Tian, Z. (2022). CSKG4APT: A cybersecurity knowledge graph for advanced persistent threat organization attribution. *IEEE Transactions on Knowledge and Data Engineering*.

Satter, R. (2021, April 13). *SolarWinds says dealing with hack fallout cost at least $18 million*. Reuters. https://www.reuters.com/technology/solarwinds-says-dealing-with-hack-fallout-cost-least-18-million-2021-04-13/

SolarWinds. (2021, April 6). *SolarWinds Security Advisory*. SolarWinds. https://www.solarwinds.com/sa-overview/securityadvisory

Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., & Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: a survey and new perspectives. *IEEE Communications Surveys & Tutorials*.

Threat Hunter Team. (2021, January 18). *Raindrop: New Malware Discovered in SolarWinds Investigation*. Symantec Enterprise Blogs; Broadcom. https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/solarwinds-raindrop-malware

Unit 42. (2020, December 23). *SolarStorm Timeline: Details of the Software Supply-Chain Attack*. Unit42; Palo Alto Networks. https://unit42.paloaltonetworks.com/solarstorm-supply-chain-attack-timeline/

Venkataraman, S., & Drummonds, S. B. (2000, October). POIROT: A logic fault diagnosis tool and its applications. In *Proceedings International Test Conference 2000 (IEEE Cat. No. 00CH37159)* (pp. 253-262). IEEE.

Villeneuve, N., & Bennett, J. (2012). Detecting apt activity with network traffic analysis. *Trend Micro Incorporated Research Paper*, 1-13.

Xie, Y., Muniswamy-Reddy, K. K., Feng, D., Li, Y., & Long, D. D. (2013). Evaluation of a hybrid approach for efficient provenance storage. *ACM Transactions on Storage (TOS)*, *9*(4), 1-29.

Zeng, J., Wu, S., Chen, Y., Zeng, R., & Wu, C. (2019). Survey of attack graph analysis methods from the perspective of data and knowledge processing. *Security and Communication Networks*, *2019*, 1-16.

Zenitani, K. (2023a). Attack graph analysis: an explanatory guide. *Computers & Security*, *126*, 103081.

Zenitani, K. (2023b). From attack graph analysis to attack function analysis. *Information Sciences*, *650*, 119703.

Zhao, G., Xu, K., Xu, L., & Wu, B. (2015). Detecting APT malware infections based on malicious DNS and traffic analysis. *IEEE access*, *3*, 1132-1142.

Zhu, Z., & Dumitras, T. (2018, April). Chainsmith: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In *2018 IEEE European symposium on security and privacy (EuroS&P)* (pp. 458-472). IEEE.