SECTOR

**SecTor 2024 – Submission Examples**

**EXAMPLE # 1**

**Title**
AI Apocalypse Prevention 101: Meet AIBOM, Your New Best Friend!

**Speakers**
Helen Oakley, (Director of Software Supply Chain Security and Secure Development, SAP)
Larry Pesce, (Product Security Research and Analysis Director, Finte State)

**Tracks**
Emerging Tech and AI, ML, & Data Science

**Abstract**
In the rapidly evolving field of artificial intelligence (AI), understanding and managing AI components and their interdependencies is crucial. This talk introduces the concept of AI Bill of Materials (AIBOM), a comprehensive framework inspired by the Software Bill of Materials (SBOM), which aims to catalog the various elements within an AI system. By integrating AIBOMs with SBOMs, we can enhance transparency, traceability, and security in AI development and deployment.

We will explore the current state of AIBOMs, examining their development, adoption, and existing challenges. Additionally, we will delve into the future potential of AIBOMs, highlighting areas where the community can contribute, such as identifying and mitigating vulnerabilities (CVEs, CWEs), and expanding the scope to include detailed training data definitions and configuration items.

Through real-world case studies, including the misuse of an AI chatbot in an auto dealership, errors in vehicle quote generation and more, we will illustrate the importance of robust AIBOM practices. Finally, we will discuss how both blue and red teams can leverage AIBOMs to enhance security, demonstrating their dual-use potential for both offensive and defensive strategies.

Join us to gain insights into the significance of AIBOMs, learn from practical examples, and discover how you can contribute to advancing this essential framework.

**Presentation Outline – NOTE THE DETAILED OUTLINE**
1. Introduction
a. Welcome to the new Era, where AI is in the forefront
b. Overview of SBOMs
c. Introduction to AIBOMs
d. How SBOMs and AIBOMs work together

2. Current State of AIBOMs
a. Development and adoption
b. Existing challenges
c. Examples of current AIBOM implementations

3. Future of AIBOMs
a. Potential improvements
b. Community contributions needed: CVEs, CWEs, CPEs
c. Expanding coverage: enhanced training data definition, configuration items, training data bias, …

d. CISA.gov SBOM Tiger Team for AIBOM

4. Case Studies
a. Auto dealership chatbot misuse
i. Scenario description
ii. Impact and lessons learned
b. Vehicle quote generation errors
i. Scenario description
ii. Impact and lessons learned
c. Data poisoning
i. Scenario description
ii. Impact and lessons learned

5. Dual-use of AIBOMs
a. Positive applications for improved security (blue team)
b. Potential for exploitation (red team)
c. Strategies for balancing dual-use

6. Conclusion
a. Three takeaways
i. Importance of AIBOMs in supporting the software supply chain security of AI
ii. How AIBOMs could enable security teams to implement resilient AI systems
iii. Call for action to the community and opportunities that AIBOM can provide

**Is This Content New or Has it Been Previously Presented / Published?**
This is new content that has never been presented before.

**What New Research, Concept, Technique, or Approach is Included in Your Submission?**
We're talking about how one can meld supply chain security with the use of AI, which is a new concept, and is arguably important with the meteoric rise in the use of AI across many verticals and applications.

**Provide 3 Takeaways**
1. Importance of AIBOMs in supporting the software supply chain security of AI
2. How AIBOMs could enable security teams to implement resilient AI systems
3. Call for action to the community and opportunities that AIBOM can provide

**Will You Be Releasing a New Tool? If Yes, Describe the Tool.**
No

**Is This a New Vulnerability? If Yes, Describe the Vulnerability.**
N/A

**If This is A New Vulnerability, Has It Been Disclosed to the Affected Vendor(s)?**
N/A

**Will Your Presentation Include a Demo? If Yes, Describe the Demo.**
N/A

**Provide the Names of the Speakers Presenting and Their Previous Speaking Experience.**
Helen Oakley:

- Speaking frequently at cybersecurity and tech conferences, including SecTor (last presentation was for SecTor Cloud Summit in 2023, also presented in previous years), RSAC, SecureWorld, CISO Forum Canada, and Elevate.
- Helen does not have a video of past presentations (eg. SecTor) but here is an interview by ITSP Magazine at RSAC 2024 on the topic of AIBOM: LINK REDACTED

Larry Pesce:

- Over the last 20 years, Larry has spoken at Shmoocon, DEF CON (DEFCON 20 Panelist "Security FAIL 5.0" 2012, DEFCON 21 Panelist "Security FAIL 6.0" 2013, DEF CON 26 (Phishing without Failure and Frustration 2017), AutoCyber '24, Derbycon, Wild West Hackin' Fest, Cactuscon, RSAC, and is a Principal Instructor with SANS and a prolific podcaster (Paul's Security Weekly for 19+ years).
- The following is one example of Larry's presentation skills: LINK REDACTED

**Does Your Company/ Employer Provide a Solution to the Issue Addressed? If Yes, Please Provide Details.**

- Helen Oakley:
  SAP does not offer any product for an SBOM / AIBOM, but we do research and implement software supply chain security, hence the intent is to share best practices and encourage the community to follow us on this journey of securing AI systems. This talk will be vendor-agnostic.
- Larry Pesce:
  As an employee of a company that generates SBOMs and eventually AIBOMs, they can be integrated with many open source tools. My intent is to illustrate not that we have a solution, but that we need solutions from a wide range of options, and as a community.

**Who is the Intended Audience?**
Security practitioners; Director/VP

**Is the Content Tailored for a Canadian Audience?**
No

**White Paper**
N/A

**SecTor 2024 – Submission Examples**

**EXAMPLE # 2**

**Title**
Isolation or Hallucination? Hacking AI Infrastructure Providers for Fun and Weights

**Speakers**
Hillai Ben-Sasson (Security Researcher, Wiz)
Sagi Tzadik (Security Researcher, Wiz)

**Tracks**
AI, ML, & Data Science and Cloud Security

**Abstract**
More and more companies are adopting AI-as-a-Service solutions to collaborate, train and run their artificial intelligence applications. From emerging AI startups like Hugging Face and Replicate, to mature cloud companies like Microsoft Azure and SAP – thousands of customers trust these services with their proprietary models and datasets, making these platforms attractive targets for attackers.

Over the past year, we've been researching leading AI service providers with a key question in mind: How susceptible are these services to attacks that could compromise their security and expose sensitive customer data?

In this session, we will present our novel attack technique, successfully demonstrated on several prominent AI service providers – including Hugging Face and Replicate. On each platform, we utilized malicious models to break security boundaries and move laterally within the underlying infrastructure of the service. As a result, we were able to achieve cross-tenant access to customers' private data, including private models, weights, datasets, and even user prompts. Furthermore, by achieving global write privileges on these services, we could backdoor popular models and launch supply-chain attacks, affecting AI researchers and end-users alike.

Join us to explore the unique attack surface we discovered in AI-as-a-Service providers, and learn how to mitigate and detect the kind of vulnerabilities we were able to exploit.

**Presentation Outline – NOTE THE DETAILED OUTLINE**
1. Introduction
a. About us

2. Intro to AI-as-a-Service Providers
a. We will provide a quick overview of cloud-based AI providers, explaining their purpose within the AI ecosystem.
b. We will demonstrate the common usages for AI-as-a-Service providers.

3. The problem with AI infrastructure providers
a. AI models come in multiple formats, produced by frameworks like TensorFlow and PyTorch.
b. Most of those formats are prone to code execution (some are literally executable files!). Despite that, we found many AI platforms failing to treat these formats with appropriate guardrails.
c. Following this idea, we went on an extensive research project with a mission:
i. Step 1: Deploy malicious AI artifacts to AI service providers.
ii. Step 2: Execute code on the service's infrastructure.
iii. Step 3: Try to move laterally within the environment.
iv. Step 4: Prove access to private data.
v. Step 5: Profit! (as in, report our vulnerabilities and make these platforms safer)

4. Case Study #1: Hacking Hugging Face with a malicious model file

a. We uploaded a backdoored PyTorch model using the pickle format, injecting hidden functionality into an otherwise legitimate language model. The backdoor will act as a regular LLM chatbot, but will execute arbitrary shell commands upon receiving a specially crafted prompt.
b. Using this backdoor, we started exploring Hugging Face's underlying infrastructure. Exploiting a chain of EKS misconfigurations, we successfully escaped from our isolated customer container.
c. After escaping, we were able to access other customers' data – including all private repositories! The data we uncovered included models, datasets, and proprietary source code from thousands of organizations.
d. Furthermore, we were surprised to find that a similar attack on Hugging Face Spaces gave us write permissions – allowing us to overwrite popular AI applications hosted on Hugging Face, and opening a path to wide-scale supply-chain attacks.

5. Case Study #2: Hacking Replicate with a malicious model container
a. Replicate uses a self-developed format called "Cog" to deploy models. The format is a thin wrapper around Dockerfiles, and essentially allows users to deploy Docker containers with their own arbitrary code. We chose to deploy a malicious Cog file, with a Docker container that spawns a reverse shell.
b. We encountered a hardened K8s environment, that limited our permissions and prevented common lateral movement paths. Instead of focusing on container escape vulnerabilities, we utilized our capabilities within the container to exploit several design bugs we found on Replicate's infrastructure. This, in turn, granted us read-write access to the centralized inference queue!
c. Our newfound access allowed us to intercept every prompt and prediction made by any AI model on Replicate, allowing us to both read sensitive customer data, as well as write our own malicious predictions and serve them to customers.

6. Case Study #3: Hacking SAP with a malicious training procedure
a. SAP AI Core offers a K8s-based infrastructure solution for model training. Each customer pod spawns in an isolated network state, blocking access to any internal network resources.
b. However, using a chain of vulnerabilities, we were able to bypass this isolation and gain free access to the virtual network.
c. Once inside that network, we exploited several misconfigurations to gain admin-level access on the K8s cluster, as well as several critical SAP resources (container registry, build system, various cloud accounts, and more).
d. Using our newfound privileges, we had the ability to access (as well as overwrite) customer models, datasets, source code and Docker images. We also had access to external customer secrets; for example, many of SAP's customers used S3 to host training files, allowing us to access their AWS account independently of SAP. Same goes for several other CSPs supported by the platform.

7. Multi-tenancy in AI: the risks
a. Our research demonstrates the tangible risk of cross-tenant vulnerabilities in training and inference infrastructure.
b. The impact of the attacks we described can range across multiple fields:
i. Intellectual property theft: Many companies trust AI service providers with their private models and datasets, which are often the unique IP that differentiates a company from its competitors. By compromising an inference service like we've demonstrated above, an attacker could steal these corporate secrets and use them for nefarious purposes.
ii. Customer privacy: People use services like ChatGPT and DALL-E for a wide variety of purposes, many of them are private, proprietary, or confidential. By compromising an AI service and intercepting prompts, one could leak mass amounts of PII and private information from unsuspecting end-users.
iii. Supply-chain RCE attack: Since most model formats are prone to arbitrary code injection, by attacking the AI pipeline and injecting malicious code into popular models, one could plant backdoors in production services as well as developer workstations and testing environments.
iv. Supply-chain poisoning attack: By attacking the AI pipeline and intercepting customer prompts, one could inject arbitrary answers and predictions whenever certain words are present. This attack could abuse the trust people have in the answers provided by AI models.

8. Detection and Mitigation

a. Finally, we will detail our proposals for detecting and mitigating these kinds of attacks. Following are a few key points.
b. Treat AI as code: AI models are often capable of running arbitrary code as part of their spec. However, many companies fail to treat AI models like they would treat untrusted user code running in their environment – even though these are equivalent risks. Stronger guardrails should be applied when dealing with external models.
c. Use multiple security boundaries: Multi-tenant AI environments should apply more than one security boundary between different tenants. In several engagements, we found that a single vulnerability is all it takes for a customer to reach other customers' data.
d. Improve scanning capabilities for AI files: Some AI formats are capable of running arbitrary code; a prominent example is the pickle format used by PyTorch. Scanning these files for malicious contents can be challenging, but it's very important to ensure model integrity when collaborating with others.
e. Shift to safer AI formats: There are several safer alternatives to the popular executable file formats, i.e. the open source "safetensors" format developed by the Hugging Face team.

9. Summary and takeaways
a. Finally, we will summarize with a short overview of our conclusions from these comprehensive research engagements:
i. The anatomy of a real-life AI-as-a-Service attack
ii. The different risks throughout the AI pipeline
iii. AI models are (often) code, and should be treated as code

**What New Research, Concept, Technique, or Approach is Included in Your Submission?**
We are presenting a new kind of AI vulnerability; while others have talked about the subject of dangerous AI formats, none have escalated this concept into a full-blown service takeover. We are going to demonstrate 3 different variations of this attack path in the real world, on 3 major AI companies, to illustrate the potential risks of such attacks and the ways to avoid them.

**Provide 3 Takeaways**
1. Attendees will get a detailed understanding of what real-life attacks on AI service providers look like, as well as their potential implications, and the ways to defend against them.
2. Attendees will gain an understanding of the similarities and differences between attacks at different stages of the AI pipeline – attacks during development, during training, during deployment, and during inference – and the unique risks associated with each one.
3. AI models should be treated as code: Many popular AI formats allow arbitrary code execution by design. Appropriate guardrails should be applied when dealing with these kind of files – especially when collaborating with external resources and teams. Attendees will learn about different methods for safer sharing of AI artifacts.

**What Problem Does Your Research Solve?**
Our research sheds a light on a novel attack vector in the AI space – hostile takeover of AI service providers. By spreading awareness of our methodologies and the risks we uncovered, we can assist AI practitioners in securing their own models and applications.

**Will You Be Releasing a New Tool? If Yes, Describe the Tool.**
No.

**Is This a New Vulnerability? If Yes, Describe the Vulnerability.**
Yes. Our research shows a new attack vector: taking over AI-as-a-Service providers, using a combination of malicious AI models and cross-tenant vulnerabilities. As part of this research, we will showcase 3 unique vulnerability chains for each AI service.

**If This is A New Vulnerability, Has It Been Disclosed to the Affected Vendor(s)?**
All our findings were responsibly disclosed to the relevant vendors.

**Will Your Presentation Include a Demo? If Yes, Describe the Demo.**
Yes. We will present a demo of each vulnerability – Hugging Face, Replicate and SAP.

**Provide the Names of the Speakers Presenting and Their Previous Speaking Experience.**
Hillai Ben-Sasson – spoke at Black Hat USA, SecTor, and KubeCon USA in 2023.
Sagi Tzadik – spoke at Black Hat Europe 2021 and RSA 2023.

**Who is the Intended Audience?**
Security researchers at the practitioner level will learn about practical attack techniques on the AI ecosystem. Security executives will learn about the novel concepts that constituted our attack scenarios, and the new threat on AI-based applications we are showcasing.

**White Paper**
N/A

**SecTor 2024 – Submission Examples**

**EXAMPLE # 3**

**Title -** Hello From the Dumpster Fire: Real Examples of Artificially Generated Malware, Disinformation and Scam Campaigns

**Speaker -** Ashley Jess, (Senior Intelligence Analyst, Intel 471)

**Tracks -** AI, ML, & Data Science and Malware & Human Factors

**Abstract**
All technological developments, even when they're made with the best of intentions, have two sides — the side that can be used for good, and the side that can be maliciously exploited.

The capabilities of artificial intelligence (AI) tools and large language models (LLMs) when it comes to features such as real-time spoken conversations, text and "vision," real-time translation, and memory capabilities are astounding, and have a wealth of legitimate uses. But also: these tools are available to anyone, and this includes malicious actors. The growth of these tools means that the barrier to detection is going to rise significantly, leaving many people (both the luddites and the tech-savvy) susceptible to becoming victims. In all likelihood, while technology companies clamber to come out on top of the AI renaissance, romance and pig butchering scams will continue to soar past already unprecedented rates, political disinformation will become harder to spot, and more and more threat actors will be able write code with little to no technical expertise or background.

This presentation will provide a brief overview of the current state of AI before we dive into showcasing real examples of malicious use of generative AI in various types of cyber campaigns, to include: malware development, disinformation and scam campaigns. Our real examples will showcase a combination of legitimate and illicit AI tools, with the aim of providing insight into the various tactics, techniques and procedures leveraged by malicious actors. The talk will conclude with coverage on the importance of content provenance and methods for detecting artificially generated content.

**Presentation Outline – NOTE THE DETAILED OUTLINE**

A. The Current State of AI (3 minutes)
a. Basic Introduction to the Topic
    i. What is an LLM?
    ii. What is a deepfake?
b. Prevalence

B. Threat Actor Toolkit (15 minutes)
a. Legitimate Tools (Broad description)
b. Illegitimate Tools i. Past Tools
    i. [Redacted Name of Tool]
    ii. WormGPT (Got a lot of notoriety, shut down)
    iii. Generator 3.0 (Longtime service that joined the AI bandwagon, claiming to be AI (but wasn't), also gained notoriety and shut down)
c. Present Tools
    i. DarkGemini
    ii. [Redacted Name of Tool]

d. Lifespan of illicit AI tools

C. Examples of malicious generative AI (15 minutes)
a. Malware development
    i. [Redacted Name of threat actor]
    ii. Black Mamba
    iii. RatChatPT
b. Disinformation and propaganda
    i. Doppelganger campaign
    ii. Election examples worldwide
    iii. Assessment of risk to Canada
c. Scam campaigns
d. Miscellaneous other uses of interest
    i. KYC bypass
    ii. Prompt injection of LLM memory
    iii. Data parsing

D. Demo (7 minutes)
a. Showcasing an LLM creating a malicious script
b. Uploading it to an AV service to see if it gets a pass score

E. Prevention and detection (5 minutes)
a. Content provenance (broad overview of the concept)
b. Detection & Prevention
    i. Increasing difficulty of manual detection
    ii. Recent efforts and research in the field
    iii. Products
        1. Vendor landscape (Large amount of offerings, questionable results)
    iv. Basic cyber hygiene

**Is This Content New or Has it Been Previously Presented / Published?**
This is a new presentation.

**What New Research, Concept, Technique, or Approach is Included in Your Submission?**
The approach for this talk is to take a deep-dive into what is really happening with generative AI amongst cybercriminals, showing real capabilities, with a pragmatic approach to demonstrate how they have evolved over the last year and a half and ending with an assessment of what is coming next and how to properly protect your organization.

**Provide 3 Takeaways**
1. Gain practical knowledge of new ways threat actors are leveraging AI for various cybercriminal schemes
2. Identify various generative AI tools that are being leveraged by threat actors
3. Identify new ways to protect against malicious use of generative AI, both in the short- and long-term

**What Problem Does Your Research Solve?**
While this presentation covers a broad issue, providing real, tangible examples helps increase situational awareness and helps security teams create a proactive, effective response to the threat.

**Will You Be Releasing a New Tool? If Yes, Describe the Tool.**

No.

**Is This a New Vulnerability? If Yes, Describe the Vulnerability.**
No.

**If This is A New Vulnerability, Has It Been Disclosed to the Affected Vendor(s)?**
Not Applicable.

**Will Your Presentation Include a Demo? If Yes, Describe the Demo.**
In the talk, I would showcase prompting an LLM to create a keylogger or loader, followed by uploading it to a service like VirusTotal to showcase the malicious script getting a pass score. This is to demonstrate how low the barrier now is for cybercriminals to attempt to generate code, while also showcasing the actual quality of code that these LLMs can create, good or bad. Time permitting, I would also like to showcase using the LLM to autonomously generate until it found code that was not detected. I would run that in the background while I resumed presenting, and return to those results by the end of my presentation.

**Provide the Names of the Speakers Presenting and Their Previous Speaking Experience.**

- Speaker: Ashley Jess
- Previous presenter at the NCFTA's Slam Spam (DISRUPTION) conference in 2022. The talk was highly attended, and well-reviewed. (Not Recorded)
- A video of me virtually presenting can be seen on BrightTalk, though registration may be required: LINK REDACTED
- Also, to hear more of my speaking style, here is an episode of a podcast I did back in 2020: LINK REDACTED

**Does Your Company/ Employer Provide a Solution to the Issue Addressed? If Yes, Please Provide Details.**
No, we do not offer LLMs, content provenance software, nor AI detection software. The section on the vendor landscape does not include us, nor is it a sales pitch for any company listed. It is simply to showcase how many companies are attempting to enter the space, and what research is being done in the field of content authentication and generative AI detection from a technical standpoint.

**Who is the Intended Audience?**
Primary Audience: Security Practitioners
Secondary Audience: Executives, Students, anyone with interest in AI (general)

**Is the Content Tailored for a Canadian Audience?**
No

**White Paper**
We are looking into drafting a white paper ahead of October. The whitepaper is pending source risk evaluations.

**SecTor 2024 – Submission Examples**

**EXAMPLE # 4**

**Title**
Breaching AWS Accounts Through Shadow Resources

**Speakers**
Yakir Kadkoda, (Lead Security Researcher, Aqua Security)
Ofek Itach, (Senior Security Researcher, Aqua Security)
Michael Katchinskiy, (Senior Security Researcher, Former Aqua Security)

**Tracks**
Cloud Security & Defense & Enterprise Security

**Abstract**
The cloud seems complex, but it's what happens behind the scenes that really complicates things. Some services utilize others as resources as part of their logic/operation. Interestingly enough, it turns out that this could lead to catastrophic results if done unsafely.

This talk will present six critical vulnerabilities that we found on AWS, along with the stories and methodologies behind them. We will present in detail the vulnerabilities found in CloudFormation, Service Catalog, Glue, EMR, SageMaker, and CodeStar. The vulnerabilities range from remote resource injection that facilitate remote code execution, potentially causing full account takeover, to information disclosure, potentially exposing sensitive data, or causing Denial of Service.

The session will share our story of discovery, how we were able to identify commonalities among them, and how we developed a method to uncover more vulnerabilities and enhanced the impact by using common techniques leading to privilege escalation.
The assumption that AWS account IDs are not sensitive data will be challenged, as we will show that this is all an attacker really needs to know in order to take over an account.
We will then detail our approach for mapping service external resources and release our Open-Source tool to research service internal API calls. We will also present a method to check if accounts have been vulnerable to this vector in the past.

We will conclude our talk with the lessons learned during this research and our future line of research. We will highlight new areas that cloud researchers need to explore when hunting for cloud vulnerabilities and highlight best practices for developers to use in complex environments.

**Presentation Outline – NOTE THE DETAILED OUTLINE**
Introduction (5 minutes):
- The session will begin with an outline of the origins of the research and the motivation behind it.
- A brief overview of vulnerabilities in the cloud that have been discovered will be presented, and they will be categorized by the common characteristics they share.
- The range of cloud providers and the array of services they offer will be explored, albeit very generally. Furthermore, by our tool, which will be delved into more deeply later, the complexity and numerous API requests that occur behind the scenes for simple operations will be illustrated, using a simple service as an example.

Revealing an Attack Vector: Shared Resources Between Services (5 minutes):
- The process of discovering the first vulnerability in AWS CloudFormation will be described.
- A novel concept of shared resources between services as a new attack vector will be introduced, highlighted by the ability for anyone to create an s3 bucket that is then used as part of the logic for other AWS services. The approach to attacking a service via Shared Resources will be elucidated through a variety of examples and scenarios.

Escalate the Vulnerability from DoS to RCE (7 minutes):
- The limitations encountered with the initial CloudFormation vulnerability, which was exclusively a DoS issue at this stage, along with ideas for new attack vectors, will be outlined.
- The root cause of the vulnerability will be explained, along with how it was escalated to information disclosure through the creation of a shared resource (in this case a s3 bucket) with a lax policy.
- The methods used to chain multiple techniques for achieving remote code execution (RCE) and full account takeover will be demonstrated. The concept of Resource Injection in CloudFormation Templates will be explained (known technique), along with a presentation of a Proof of Concept (PoC) developed to automate this technique.
- A PoC video/demo of the CloudFormation vulnerability will be shown, demonstrating how the entire account was taken over.
- It will be shown how the attack was expanded across all regions by claiming all the unclaimed bucket names (of the targeted service) of the victims.

Exploring Additional Vulnerable Services (7 minutes):
- In this phase, the focus will be on expanding the research to other AWS services that utilize shared resources as part of their functionality.
- Attention will be directed towards two specific services: Glue and SageMaker, with a focus on the unique aspects of their vulnerability vectors. In Glue, for instance, an issue was identified that enables attackers to implant backdoors in the service's scripts, concealing these backdoors from victims due to a graphical user interface (GUI) bug. SageMaker's vulnerability relates to information disclosure. However, given its nature as a machine learning service, it also presents risks such as data poisoning/manipulation (AI risks). A PoC video demonstrating these vulnerabilities will be presented.
- The vulnerability vectors in other services, including EMR, Service Catalog, and CodeStar, will be explained. Their vectors will be discussed briefly, as they follow the same underlying concept.
- Details will be provided on services initially thought to be vulnerable but were found not to be, due to fixes implemented over time (for example – Athena). This will underscore the diverse development processes and decisions unique to each service in the cloud environment, along with the challenges associated with maintaining the same principles in different components.

Vulnerable Assets in Open-Source Projects (2 minutes):
- This section will reveal that the issue extends beyond AWS's internal design, identifying additional vectors of this vulnerability in open-source project.
- The demonstration will cover template files and bash scripts in open-source projects that lead to similar vulnerabilities.
- Suggestions for mitigation and identification of poor practices in creating scripts/templates susceptible to these vulnerabilities will be provided.
Harvesting AWS Account IDs and Hashes to Weaponize Our Attack Vector (3 minutes):
- This section will challenge the assumption that AWS account IDs are not sensitive data and outline the research method for harvesting AWS account IDs and explain its significance in exploiting this vector.

- We will illustrate how it is possible to achieve this by using GitHub search, various collections, and even fuzzing of existing bucket names, along with statistics, demonstrating how easy and popular it is to find AWS account IDs or hashes of many users.
Automating Discovery – AWS Shared Resource Hunter Open-Source Tool (5 minutes):
- We will introduce our tool to analyze AWS service API calls (and release it as free open-source) - this tool utilizes CloudTrail to observe communication between services. It analyzes the interaction and hierarchy of the AWS REST API, allowing us to track the call stack for every Amazon API request.
- We will highlight the differences between the API of the AWS CLI and the AWS console (GUI), and how some of the vulnerabilities that we found were related to this.
- We will show a demo of the tool on a specific service and explain how the flow of the API and communication between services can be observed.

Disclosure and Timeline (1 minute) – We will provide a brief description of the timeline of the disclosure to AWS, including when the fix was implemented.

Mitigation and Recommendations (3 minutes):
- Insights will be shared from this research, offering mitigation strategies for developers of complex environments (for example- creating thread modeling for resources used by internal services and checking the owner of resources before using them, etc.)
- We will present some methods to determine if an organization was at risk of this attack vector before. (After our report, this vector in the reported service is no longer possible, but there are checks that users can perform to ensure that attackers haven't claimed their resources/buckets in the past).

Summary (2 minutes) - We will conclude our session with a brief summary of our session and highlight new areas that cloud researchers need to explore when hunting for cloud vulnerabilities. We will also highlight best practices for developers to use in complex environments.

Q&A (5 minutes)

**Is This Content New or Has it Been Previously Presented / Published?**
This content has never been presented or published before.

**What New Research, Concept, Technique, or Approach is Included in Your Submission?**
Our submission includes six new critical vulnerabilities that we discovered across different AWS services. They all share a common new technique that we have termed "Shared Resources" - This technique involves the internal logic of a service utilizing other AWS resources that can be claimed by an attacker in new regions. In this manner, the attacker can control the service logic from outside. For instance, this led to remote code execution and full account takeover of the victim. All the attacker needed to know was the Account ID of the victim, which is considered non-sensitive data, and find a region where the victim didn't use the vulnerable service yet (which expands the attacker's surface and increases the odds of a successful attack).

In more detail, some AWS services use S3 buckets as part of their operations, and these services require S3 buckets to follow a specific naming pattern predefined for each service. For example, Glue uses the pattern "aws-glue-assets-{Account-ID}-{Region}". Since any S3 user can create a bucket with a yet-to-be-claimed name, an attacker could create these buckets in advance, set them to public access, and grant read/write permissions to anyone. Furthermore, the attacker could monopolize all the S3 buckets associated with a specific service across unclaimed regions, given that the region of the service is part of the bucket's name. This strategy allows the attacker to wait for a victim to initiate the service in a new or previously unclaimed region for the first time.

Behind the scenes, the victim's service would inadvertently attempt to "create" or trust the attacker controlled s3 bucket, which is directly linked in a 1:1 relationship with the victim's service. Consequently, this enables the attacker to access and potentially alter the data the victim's service stores in the attacker's bucket, including injecting malicious content or extracting sensitive data. Users will remain unaware that the S3 buckets for their services are not under their control.

We have included in this submission the original report and some PoC videos (to gain a better understanding of the research).

**Provide 3 Takeaways**
1. Learn about new types of risks and vulnerabilities in different cloud provider, and offer a way to navigate the complexity of the API and the interactions between different services.
2. Give mitigation and recommendations for these vulnerabilities, and similar issues that may be discovered in the future. For example, from the developer side, verify that all the required components needed for an application/service are created in the correct account. This mainly applies to open-source projects that check if a service exists but don't check where it exists. And in addition, create threat modelling for each component when developing complex architectures, especially if there is interaction between two different services/resources.
3. Learn that services maintaining a state are sensitive. If attackers can access them, they could potentially execute code with elevated privileges or similar.

**What Problem Does Your Research Solve?**
The research revealed a new concept of vulnerabilities that could exist in the cloud and provided best practices for developers and users to minimize these risks.

**Will You Be Releasing a New Tool? If Yes, Describe the Tool.**
Yes, we plan to release an open-source tool licensed under the Apache License, coinciding with our talk's date. Our research highlighted the necessity for a tool capable of monitoring and comparing internal AWS API calls to the original API calls we executed. To achieve this, we utilized AWS's auditing service, CloudTrail. The tool tracks all API calls, emphasizing those initiated by AWS internals involving interactions between two or more services. This methodology enabled us to uncover six critical vulnerabilities across various AWS services. We have attached a screenshot of the tool to this submission, where you can see the analysis of the CloudFormation Service and Glue.

**Is This a New Vulnerability? If Yes, Describe the Vulnerability.**
Yes, we have discovered 6 new critical/severe vulnerabilities within AWS that could enable external attackers to target other AWS users and carry out a variety of harmful actions.
These include remote code execution and resource injection, potentially leading to a complete compromise of the victim's cloud environment. Other vulnerabilities could result in information disclosure of sensitive data, while some may lead to Denial of Service (DoS), preventing users from accessing specific services.

High-Level description of the vulnerable services:
1. CloudFormation: Identified a remote resource injection vulnerability that facilitates remote code execution (RCE), leading to potential full account takeovers and other critical impacts. (PoC video attached)
2. Service Catalog: Identified a remote resource injection vulnerability that facilitates remote code execution (RCE), leading to potential full account takeovers and other critical impacts.
3. Glue: Discovered a vulnerability that enables remote code execution. (PoC video attached)
4. EMR (Elastic MapReduce): Found an RCE vulnerability within EMR Studio.

5. SageMaker: Uncovered a vulnerability leading to information disclosure, potentially exposing sensitive data.
6. CodeStar: Identified a Denial of Service (DoS) vulnerability, which could prevent users from accessing the service.
Vulnerability Overview and Root Cause Analysis: Some AWS services use S3 buckets as part of their operations, wherein these services require S3 buckets to follow a specific naming pattern that is predefined for each service. For example, Glue uses the pattern "aws glue-assets-{Account-ID}-{Region}". Since any S3 user can create a bucket with a yet-to-be-claimed name, an attacker could create these buckets in advance, set them to public access, and grant read/write permissions to anyone. Furthermore, the attacker could monopolize all the S3 buckets associated with a specific service across unclaimed regions, given that the region of the service is part of the bucket's name. This strategy allows the attacker to wait for a victim to initiate the service in a new or previously unclaimed region for the first time. Behind the scenes, the victim's service would inadvertently attempt to "create" or trust the attacker-controlled S3 bucket, which is directly linked in a 1:1 relationship with the victim's service. Consequently, this enables the attacker to access and potentially alter the data the victim's service stores in the attacker's bucket, including injecting malicious content or extracting sensitive data. Users will remain unaware that the S3 buckets for their services are not under their control. Some of the vulnerabilities require us to create automation or involve other techniques to escalate them. More information can be found in the white paper.

**If This is A New Vulnerability, Has It Been Disclosed to the Affected Vendor(s)?**
Yes, the vulnerabilities were reported. We are in contact with the AWS security team, who received and approved this submission. The vulnerabilities were reported on February 16, and patches for 5 out of 6 services were made at different times in March. Currently, there is a need to fix the ServiceCatalog service, but it will be done before the conference.

**Will Your Presentation Include a Demo? If Yes, Describe the Demo.**
Yes, we want to show a demo or video for at least 2 of the vulnerable services in order to demonstrate the full flow and explain the vulnerability.

**Provide the Names of the Speakers Presenting and Their Previous Speaking Experience.**

- Yakir Kadkoda – Black Hat Asia Briefings 23, RSA USA 23
- Ofek Itach – Video of Ofek speaking at RSA USA '24 .
- Michale Katchinskiy - Kube Day Israel 23 – LINK REDACTED

**Does Your Company/ Employer Provide a Solution to the Issue Addressed? If Yes, Please Provide Details.**
No, since these vulnerabilities lie within the internal logic of some AWS services, AWS implemented updates for all users after our report and will send messages to their customers.

**Who is the Intended Audience?**
The audience for this session is very vast because we will discuss many security aspects of cloud environments and services. However, it may be more suited to security practitioners and developers.

**Is the Content Tailored for a Canadian Audience?**
No

**White Paper**
We are attaching to this submission the original report and the PoC videos of CloudFormation and Glue, and screenshot of the open-source tool.
We are attaching a link to our Google Drive. LINKS REDACTED