PrimeHarbor Technologies

Securosis

The Universal Cloud Threat Model

Version 1.0
Released: April 19, 2024

A joint research project by PrimeHarbor Technologies and Securosis
www.primeharbor.com www.securosis.com

Authors' Note

The content in this report was developed independently by Chris Farris of PrimeHarbor and Rich Mogull of Securosis.

Special thanks to Chris Pepper for editing and content support.

Copyright

Table of Contents

Introduction

The Universal Cloud Threat Model applies to all organizations which operate in the public cloud, regardless of industry and which cloud provider(s) in which they operate on. The UCTM was designed as a cloud-centric update to traditional threat modeling. Standard threat models such as STRIDE are excellent, but do not account for the different operating models of cloud computing. The UCTM was developed to address three primary gaps in existing models:

- In the cloud, infrastructure and applications are often deeply entangled and even indistinguishable thanks to options like serverless and infrastructure as code.
- In the public cloud, the Internet-facing attack surface now includes the administrative management plane. This is unlike traditional infrastructure, where most administrative functions are protected on internal networks behind firewalls and DMZs.
- In the public cloud nearly all organizations run on the shared infrastructure of three primary cloud service providers, followed by a slightly larger set of secondary providers (for IaaS, our focus for this threat model).

These three differences combine to expand the range of *undifferentiated* (target of opportunity) attacks, along with the potential for an attacker to pivot into a *differentiated*/targeted attack. Attackers search first for common initial vectors for attacks on a cloud provider, such as exposed credentials. Then they may use them for a more targeted attack if they identify a target of potentially higher value, such as financial services. The vectors and sequences of these attacks can be mapped, and pivot points identified.

In our research and experience, the vast majority of cloud attacks fall first into the untargeted/ undifferentiated category, even for highly desirable targets, and defenders who focus first on these vectors are more resilient. Similarly, even small and uninteresting targets offer greater financial rewards to attackers who then use the smaller target as a foothold into the Cloud Service Provider (CSP) for 'free' resources such as cryptomining — even a small cloud customer can run extensive and expensive resources before hitting service limits — or as a platform for launching other attacks. Successful exploitation of even such a small and uninteresting target enables free networking and IP addresses — at least for the attacker.

That is why we call this the Universal Cloud Threat Model. It identifies the commonalities all organizations face equally based on cloud usage — regardless of size, vertical, or nationality. We call these the "90% of attacks experienced by 90% of organizations using the cloud"[1].

[1] A rule of thumb, not a research-backed quantitative metric.

The Model

The UCTM is designed to work with other threat modeling, but to more clearly identify cloud-specific factors. Organizations should extend this universal model to their specific threats to help determine when they transition from undifferentiated and standard attack sequences to differentiated (targeted) attacks that are more manually directed and variable.

The Universal Cloud Threat Model can be broken down into the following statement:

Threat Actors have **Objectives** against **Targets** using **Attack Vectors** which are observed by defenders as **Attack Sequences**.

In this formulation,

- **Threat Actors** initiate attacks.
- **Objectives** are desired outcomes for threat actors, such as financial gain through cryptomining or theft of PII for use in financial crimes.
- **Targets** are assets of organizations of any size hosting resources in the cloud.
- **Attack Vectors** are misconfigurations and vulnerabilities which enable an attacker to gain an initial foothold on a target.
- **Attack Sequences** are the specific steps an attacker uses to achieve their ultimate objectives. They may change based on what the attacker learns during their attack.

Note that this model does not follow the standardized attack patterns of models like the Lockheed Cyber Kill Chain and MITRE ATT&CK. We see no need to recreate that excellent work, and the goal of the UCTM is to detail, simply and usefully to non-security practitioners, the specific steps of the most common attack patterns.

Threat Actors

The first element in our UCTM equation is threat actors. They may have malicious intent (cyber criminals) or merely threaten your business due to self-inflicted mistakes (auditors). We distinguish threat actors here because each has different motivations and capabilities.

Threat actors are opportunistic or directed. A threat actor may also start opportunistically and then transition to directed based on any results.

- Opportunistic threat actors don't care who you are. They scan everyone for vulnerabilities to exploit. For example many threat actors continuously scan public code repositories for cloud credentials and use them in automated attacks.
- Directed threat actors are more discerning. They have specific objectives, usually geopolitical or large financial capers. Targeted threat actors happily use any tactics here, but are more likely to supplement their attacks with directed phishing against key personnel, or perform extensive pre-attack reconnaissance to identify target-specific vectors in CSPs. For example they might explore public websites and applications using DNS to identify the CSP in use and deployment identifiers (*e.g.,* AWS account ID), or map out the target's software supply chain.

Threat Actors include:

- State-nexus threat actors
- Cybercriminals and financially motivated threat actors
- Hacktivists and cause motivated threat actors
- Insider threats
- Experimenters (script kiddies, reputation builders, 80s-style hackers)
- Non-hostile Threat Actors
  - Auditors
  - Mergers and Acquisitions (the acquired/merged entity)
  - Security companies and researchers scanning indiscriminately (e.g. Shodan)
- Rich's cat Goose. He's a legitimate jerk.

Auditors are not generally hostile, but failure to pass an audit can threaten an organization. Mergers and Acquisitions introduce new threats, increase attack surface, and distract teams from program work.

Why This Matters

Threat actors tend to engage in slightly different patterns, which is one of the biggest factors for determining whether an attack is more likely to be differentiated or undifferentiated.

- **Differentiated:** State-level nexus, hacktivists, and insiders tend to move more quickly into differentiated (targeted) attacks and leverage more advanced techniques. Some cybercriminals also focus more on differentiated attacks, but they are less consistent.
- **Undifferentiated:** Cybercriminals, experimenters, and non-hostile threat actors tend to limit themselves to targets of opportunity, and may never pivot to a more directed attack. For example crypto and ransomware attackers don't invest time into extensive reconnaissance or burn zero-day vulnerabilities.

Objectives

The most common Threat Actor Objectives in the cloud are:

- Financial gain from resource hijacking — which includes but is not limited to:
  - Financial gain from cryptomining
  - Financial gain from spam
- Financial gain from ransomware (encryption or deletion)
- Financial gain from sensitive information disclosure (blackmail)
- Financial gain from selling sensitive data on the black market
- Leveraging victim cloud infrastructure for financial attacks against others
- Leveraging victim cloud infrastructure for geopolitical attacks against others
- Industrial espionage (to gain business advantage through IP theft)
- Nation-state espionage
- Fame & Fortune (bug bounty claims, blogging, research)

As a reminder, this list is specific to public cloud computing, and does not include all attacker objectives.

For example North Korea uses cyptomining to fund its nuclear program and evade Western sanctions. They are a well-resourced state-nexus threat actor with a financial objective. They don't need differentiated attacks, and will target anyone. Ransomware collectives are also financially motivated, but don't yet have the resources of a nation-state.

> Alongside the surge in ransomware incidents throughout 2023, the escalation in cases of data theft extortion when compared to previous quarters aligns with publicly reported trends indicating a growing number of ransomware groups are pilfering data and coercing victims without encrypting their files or resorting to the deployment of traditional ransomware. Although data theft extortion is not a new phenomenon, the number of incidents this quarter suggests that financially motivated threat actors are increasingly seeing this as a viable means of receiving a good payout.
> - ENISA THREAT LANDSCAPE 2023

Why This Matters

There is an association between threat actors and their *most common* objectives — there are no firm rules in adversarial cybersecurity.

Targets

Targets are the assets (such as data or infrastructure) which attackers target to achieve objectives. An organization itself may be a selected target, which means the attacker will be willing to engage in a greater effort (*e.g.,* use more advanced techniques) and more likely to target any accessible assets of the organization.

Some organizations are specifically targeted by threat actors who may use organizational knowledge in their attacks and devise specific objectives and desired outcomes. Opportunistic attackers don't know what organization they've compromised until they get in and explore, and because they use automated tools, they may never bother to identify a target organization.

Most organizations have some or all these targets in their cloud environment, which attackers go after:

- Data Targets
  - **PII/PHI:** Any data which could be resold, or the release of which could be used to threaten the organization
  - **Financial Data:** Data with direct monetary value (credit card numbers, bank accounts) or cryptowallets. Some PII is also considered financial data.
  - **Operational Data:** Data required for the business to function. It is of no value to an outsider and limited value to a competitor but useful in ransomware or other attacks where the loss/modification of the data could be detrimental.
  - **Reputational Data:** not PII/PHI or otherwise regulated, but if released this could cause reputational damage.
  - **Competitive Intellectual Property (IP):** Data competitors or other nation-states might seek for competitive advantage.
  - Stored Credentials to allow the attacker to pivot to other resources.
- Compute Targets
  - Containers and virtual machines used for:
    - Cryptomining
    - Phishing or C&C (Command and Control) Infrastructure
    - Attack platforms
    - Pivoting via stored credentials or IAM permissions
- Network Targets
  - Botnets used for DDOS
  - Spam Relays
  - Networks for hosting attack tools and routing/masking attacks
- CI/CD Pipelines
  - Most cloud deployments are managed using CI/CD pipelines with privileged cloud credentials. An attacker can escalate or move laterally in an environment by targeting a vulnerable pipeline.
  - Pivoting via poisoning configuration files or code: *e.g.,* injection of commands into a Terraform state file
- Cloud Software Supply Chain
  - Many cloud deployments use base resources (*e.g.,* VM images) hosted by their cloud provider or shared natively across cloud platforms.

- Threat actors perform name-squatting or publish malware to fool development teams into incorporating malicious code.
- Threat actors may also target pipeline tools themselves to capture stored credentials, operational intelligence, or use the tooling to pivot into a direct attack (e.g. insert malware).

Why It Matters

Targets are the resources and assets you need to defend. Knowing these assets allows you to focus and prioritize defensive efforts. Not all targets are created equal.

Knowing whether your entire organization is the target, and for which threat actors, also helps to align and prioritize defenses. Heavily targeted organizations need to defend a wider range of internal targets (assets) because threat actors are more likely to spend more effort getting a foothold and pivoting from lower-value targets to higher-value ones. Banks, for example, should expect non-stop series of differentiated attacks from sophisticated threat actors.

Organizations less likely to be targeted using directed attacks can prioritize defense against undifferentiated attacks of opportunity. Organizations which know they *will* be targeted still prioritize mitigating undifferentiated attacks first, then move into defending against the differentiated attacks which go beyond the scope of this paper.

Attack Vectors

All this leads to the various Attack Vectors which a Threat Actor might use against a Target to accomplish an Objective. The most common cloud Attack Vectors include:

- Lost, stolen, or exposed credentials
- Publicly exposed resources
- Credentials exposed via application security flaws
- Unpatched vulnerabilities and zero-days in overly exposed systems
- Denial of Service attacks
- Subdomain takeover
- Supply chain compromise

This list is deliberately not exhaustive, and focuses on the *most likely attack vectors used in undifferentiated attacks*, which is the focus of the Universal Cloud Threat Model. Once an attacker gains successful entry via a vector, they initiate an attack sequence — a provider-specific series of steps to move from the initial vector to the objective.

Lost, stolen, or exposed credentials

Credential exposure and theft is the top attack vector in the cloud. Most other attack vectors are only stepping stones after credential theft.

> Based on Q1 2023 observations by our Google Cloud IR teams, more than 60% of compromises involved credential issues, 19% involved misconfigurations, and only 2.4% involved vulnerable software. (Mandiant August 2023 Threat Horizons Report)

Examples:

- AWS Access Keys committed to public repositories
- Compromised GitHub Personal Access Tokens (and non-GitHub equivalents), leading to theft of source code and additional secrets (from private repositories)
- Malware and infostealer pulling credentials from well-known places on systems
- Phishing attacks

> According to the Unit 42 Cloud Threat Report Volume 7, 83% of organizations expose hard-coded credentials within the production code repositories. The report offers recommendations that organizations can use to improve security around IAM credentials.
> – Unit 42: CloudKeys in the Air: Tracking Malicious Operations of Exposed IAM Keys

What to do about it

- Reduce or eliminate static credentials and tokens for the cloud management plane. Where possible use alternatives like IAM Roles and Azure Managed Identities.
  - Automation is your friend. There are plenty of free and commercial tools to hunt these down.
- Scan code for stored credentials in CI/CD pipelines and repositories, even if they are private.
- Require MFA for all human access, even when using API keys.
- Limit the duration of temporary access credentials so they expire after only a few hours.
- Migrate administrators and highly-privileged access to Just in Time access and/or use hardware tokens (*e.g.,* Yubikeys).
- Implement a "data perimeter" (this can be difficult in established enterprises, but can be very effective, even if only some basics are implemented).

Publicly Exposed Resources

Another common attack vector is publicly exposed resources. If you search for "cloud data breach," this is the cause of most or at least many of the hits. Exposed resources come in two forms: cloud resources which do not require any authentication to access, and network resources anyone on the Internet can connect to. These are nearly always due to misconfiguration, often due to a developer or admin not understanding the service or how to write resource policies.

The primary example of the first type is Public S3 buckets. Amazon S3 buckets suffered from several critical flaws in their initial design and rollout. S3 buckets share a global namespace, so if Netflix creates the public S3 bucket "viewer-habits", any other company can easily guess or stumble across that bucket name. The second key flaw with S3 was how they first rolled out ACLs and later added IAM, creating so many layers of complexity that Amazon itself needed to use machine learning (automated reasoning) to determine whether a bucket was public.

Another form of public resource is compute resources with overly permissive firewall (security group) permissions. These can be operating systems, containers, or APIs — all feature prominently in the

next two attack vectors. They are exposed directly to the background radiation of automated Internet scanning and attacks which have existed since before the cloud existed.

Cloud providers have expanded many services to support direct Internet access to resource types, from serverless functions to message queues to all flavors of databases. All these can be protected with resource-based policies and, in some cases, network security controls. But nothing is simple at scale, and mistakes will be made.

What to do about it

- Use your cloud service provider's built-in assessment tools or a free or third-party scanner to identify public resources. There is no shortage of these tools — the trick is to get sufficient coverage and consistently act on the results. If you have a commercial tool this will be a feature — and if not get another tool.
- Use your cloud provider's policy tools (guardrails) to prevent the creation of public resources where possible. For example Azure Policy, AWS Service Control Policies, and AWS Block Public Access for S3.
    - This may require a nuanced process, especially in large organizations, but the payoffs are massive.
    - Your process **must** support rapid exemptions when public resources are required, such as when you are hosting a public website in an S3 bucket (yes, they do that also).
- Some organizations have implemented automated remediation for things they can't block. Depending on your profile this might be home-grown automation, a feature from your provider, or a third-party tool.
    - We see this most successfully used for reversing really terrible mistakes, like opening port 22 to the Internet.

Credentials exposed via application security flaws

Everything exposed on the Internet is liable to being scanned and indexed for vulnerabilities. Misconfigured applications can leak credentials to threat actors, who can use them for various purposes. Common application security issues include Server-Side Request Forgery (SSRF), improperly configured reverse proxies, and debugging information exposed to the outside.

SSRF and reverse proxies can enable credential exposure by giving threat actors a path to the metadata service, the channel via which cloud providers communicate information to compute resources.

Mandiant highlighted one example of the SSRF/Reverse Proxy vulnerability in their report on UNC2903:

Given that the infrastructure is hosted within Amazon Web Services cloud, IMDS is an attractive target for threat actors like UNC2903. In UNC2903's case, the threat actor was observed targeting exploitable web applications which were also running IMDSv1. Amazon's IMDSv1 permits web requests to a specialized URL against the link local IP address (169.254.169.254) which was designed to enhance internal service communication and troubleshooting within the overall hosting platform. The retrievable metadata includes information to understand configuration, topology, and even obtain user role and credentialing

What to do about it

- Reduce the exposure of applications to the public Internet. Place compute resources behind load balancers which only allow specific API routes.
- Do not give applications highly privileged access policies. IAM policies for applications should explicitly enumerate the resources each application needs to access. Avoid using AWS policies such as FullAccess and Basic Roles in GCP. Your tools (free or commercial) can help identify these.
- Use the same techniques discussed in *exposed credentials*.
- Scan code in CI/CD pipelines for static credentials.
- Ensure that access to the Instance Metadata Service requires authentication headers and limits the number of network hops permitted. For example require IMDSv2 on AWS.
- Use a WAF to mitigate undifferentiated application layer attacks.
- If your cloud provider offers credential misuse detection (*e.g.,* session credentials used from someplace where the session didn't originate), use it. These aren't perfect and only catch these attacks sometimes, but they can really help. AWS GuardDuty is one option.
- While challenging and complex, implementing a data perimeter is one of the best defenses for this. *Prioritize this option if you are the target of differentiated attacks*.

Unpatched vulnerabilities and zero-days in overly exposed systems (virtual machines or containers)

Another method threat actors use when attacking is to leverage zero-day or unpatched vulnerabilities against Internet-exposed systems. Exposed systems are usually virtual machines or containers, which run standard operating systems.

While not every vulnerability can enable cloud credential exfiltration, the worst of these vulnerabilities lead to remote code execution and offer threat actors a gateway into the private network, or a way to exfiltrate credentials to enumerate the cloud environment. Cybercriminals have leveraged zero-days to gain footholds as part of ransomware or blackmail attacks, but plenty of known vulnerabilities don't require attackers to use zero-day exploits. Recent examples include MoveIt, Ivanti, etc.

> Attackers have therefore been forced to ‘change up’... by targeting the perimeter again. Knowing that they are less likely to be able to rely on poor passwords or misconfigurations, they are increasingly looking at products on the network perimeter (such as file transfer applications, firewalls and VPNs), finding new zero-day vulnerabilities in these products, and waltzing right in. Once a vulnerability is known, other attackers join resulting in mass exploitation.
>
> Finding zero-day / new vulnerabilities might sound highly advanced, but many of these are well-understood classes of web vulnerability and are trivial to find and exploit. At his OffensiveCon 23 keynote, Dave Aitel remarked "It's only hard to find vulnerabilities if you look for hard vulnerabilities. You should look for easy ones.”
>
> Sadly, the days where a fully patched perimeter meant you were safe from all but the most advanced attackers are long gone. Anything on your perimeter, even fully patched, is increasingly in the firing line, and unless you have evidence that it can withstand attacks, you should consider removing it. We are entering the days where organisations need to start aiming for a perimeter scan with no ports found accessible.
>
> — Products on your perimeter considered harmful
>
> UK National Cyber Security Centre

What to do about it

- Minimize the number of systems which are exposed to the entire Internet. But we understand some of these systems are exposed to the Internet for legitimate reason, such as VPN servers and jump boxes, which you cannot always eliminate like the inadvertently exposed public resources mentioned earlier. You should still scan them and know where and how they are configured.
    - Cloud Workload Protection Platforms (CWPP), which may also be part of a Cloud Native Application Protection Platform (CNAPP — blame Gartner, not us), can help identify exposed systems and perform vulnerability assessment without taking systems offline or installing an agent. Your CSP may also offer this as an option.
- For containers consider using a container security platform or your cloud provider's container scanning features. Make sure your tool can identify vulnerabilities in running containers, not just images in storage — or can identify which containers are running vulnerable images.
- Enable your cloud provider’s threat detection service (*e.g.,* GuardDuty), which can help detect cryptomining; it is a common result of this attack vector.
- Exposed systems should never hold static credentials and should have least-privilege IAM permissions. We know this is hard, but if you are going to stick a target on the Internet, don’t be surprised if someone takes you up on your offer. At least try to minimize the potential damage.

- Develop SBOM capabilities to discover and respond to zero-day threats when they arise quickly.

Denial of Service Attacks

Denial of Service attacks (DoS and DDoS) differ from our other vectors because there is no separate outcome, vector, or sequence; the objective is also the vector and sequence. Denial of Service attacks can be differentiated or undifferentiated, depending on whether the cloud provider or customer is the target. In an undifferentiated attack the provider is the target, and customers may be affected. For example Microsoft and Amazon have suffered DoS attacks which degraded customer service. These attacks are differentiated to the provider, but undifferentiated as far as customers are concerned, because they are not specifically targeted.

Cause-motivated actors and even disgruntled customers may target specific organizations.

What to do about it

- For critical workloads/applications plan for DoS attacks and use standard precautions, such as a service from your CSP or a third-party vendor (e.g. AWS Shield Advanced, or Google Cloud Armor).
- Never expose workloads directly to the Internet. Even when they need to be exposed, stick them behind a load balancer, Content Delivery Network (CDN), or similar insulating service. This is usually required to get DoS protection from your cloud provider. In some cases the CSP will offer basic DoS protection free, just by using their load balancing service.

Subdomain Takeover

Subdomain takeover occurs when a trusted reference (e.g. a DNS entry) to a resource is de-referenced by deletion of the referenced resource. This is more common in the cloud, because of its ephemeral nature and greater use of CNAME DNS entries to point to resources owned by the cloud provider. If the customer releases that resource that name goes back into the resource pool, and someone else can detect it is no longer referenced, a threat actor can use the same name to take over that traffic. The multitenant nature of the cloud enables threat actors to create their own target resource, to which your trusted reference now points.

An example of this occurred with the npm package *bignum*. Bignum originally hosted some data in an S3 bucket, but over time it moved that data and deleted the bucket. However its installer still referenced the bucket. An attacker was able to recreate the bucket and serve a malicious version of a bignum dependency.

Subdomain takeover attacks also happen when DNS entries are left in place after the underlying resource is deleted. If I have a DNS record, say payments.fooli.media, and I terminate the EC2 instance, the pointer to the IP address still exists. Attackers have figured out they can keep requesting IP addresses until they get the one to which the DNS record points. From there they can leverage phishing attacks which work because they're directing victims to a legitimate host in your domain.

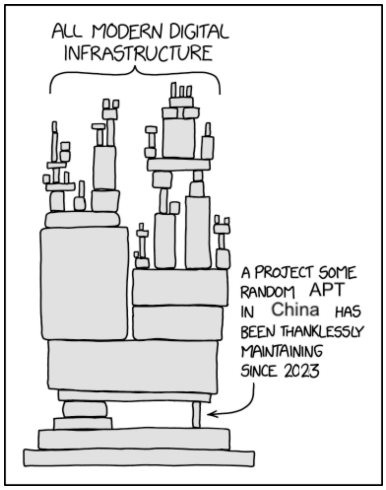
Threat actors scan DNS automatically to detect when it points to a deleted resource. They then perform takeovers to use the organization's brand in phishing and other attacks.

What to do about it

- Automation is your friend. You can identify dangling DNS entries through commercial or home-grown domain monitoring tools designed to identify these conditions.

Supply Chain Compromise

A wide range of attacks are executed against the software supply chain, and this is a growing global concern, especially as state actors continue to target software providers. However for the UCTM we will focus on the downstream implications of the supply chain as a vector for an attack which would impact many organizations, possibly including your own.



With apologies to Randall Monroe (https://xkcd.com/2347/)

In a supply chain attack the threat actor deliberately targets an open source or commercial software library, application, or component; and injects a back door or other hostile code into it. This becomes a vector to compromise customers/users who run or integrate that code/library/application into their environment. The SolarWinds attack is one of the best-known examples.

Supply chain attacks can be an effective vector for compromising cloud deployments because in the cloud we make heavy use of shared base images, libraries, and software components — due to greater use of virtual machines, containers, and Functions as a Service. We also run these in the cloud, where there is potentially greater access to cloud credentials, API tokens, and Internet access.

Attacks don't need to rely on compromising major, well-known sources. There is a proliferation of certified, pre-pwned images and libraries which disguise themselves as trusted artifacts from trusted sources — or in some way leverage real trust relationships.

What to do about it

- Only use base images and software components from known (and ideally, trusted) sources.
- Pull packages from package manager repositories rather than directly from GitHub to avoid repo-jacking.
- Cache software dependencies inside your organization so future malicious modifications are not automatically introduced into your environment.
- Software Composition Analysis (SCA) can help identify some malicious libraries and libraries of dubious provenance.

Up and Coming Attack Vectors

These are uncommon but on the radar of the cloud security community, and worth keeping in mind:

- Novel generative AI attacks could include:
  - Prompt injection and jailbreaking to disclose data
  - Alignment gaps or bypasses to produce content which is harmful or misleading
  - Proprietary data leaks: competitive IP leaked into the training data and vector databases used by AI models.
- Denial of Wallet Attack
  A denial-of-wallet attack is a cloud-specific flavor of denial of service in which the service isn't taken down due to the resilience of the cloud, but the victim receives a massive bill due for all the resources spun up to handle the increase in load. While previously theoretical, there is new research into how this is done. As of today, most of these attacks we see are an additional effect of resource hijacking for cryptomining or spam/phishing.
- CSP Compromise
  - Cloud Providers are a high-value target, as demonstrated recently by Midnight Blizzard and Storm-0558's compromise of Microsoft. These supply chain attacks carry international security implications.

Prioritizing Attack Vectors

You can prioritize attack vectors with the following qualitative equation:

Discoverability * exploitability * impact = priority

This helps focus on which vectors to find and minimize first, especially when faced with an extensive list of findings.

For example public S3 buckets are easily discoverable, easy to exploit, and — depending on content — can be high-impact. So they should be high priority. Public SQS queues are harder to discover (they have random URLs) and harder to exploit (you need to know the message format), with variable impact.

Public-facing servers/containers with exposed administrative ports are essentially effortless for attackers to discover. Exploitability depends on vulnerabilities or use of weak credentials. The impact varies depending on the chosen attack sequence, which we will cover next.

Attack Sequences

Attack sequences are the specific steps an attacker uses to achieve an objective. The attack vector is the initial method of entry, and the sequence is every action after that. We would love to eliminate all vectors but that isn't realistic.

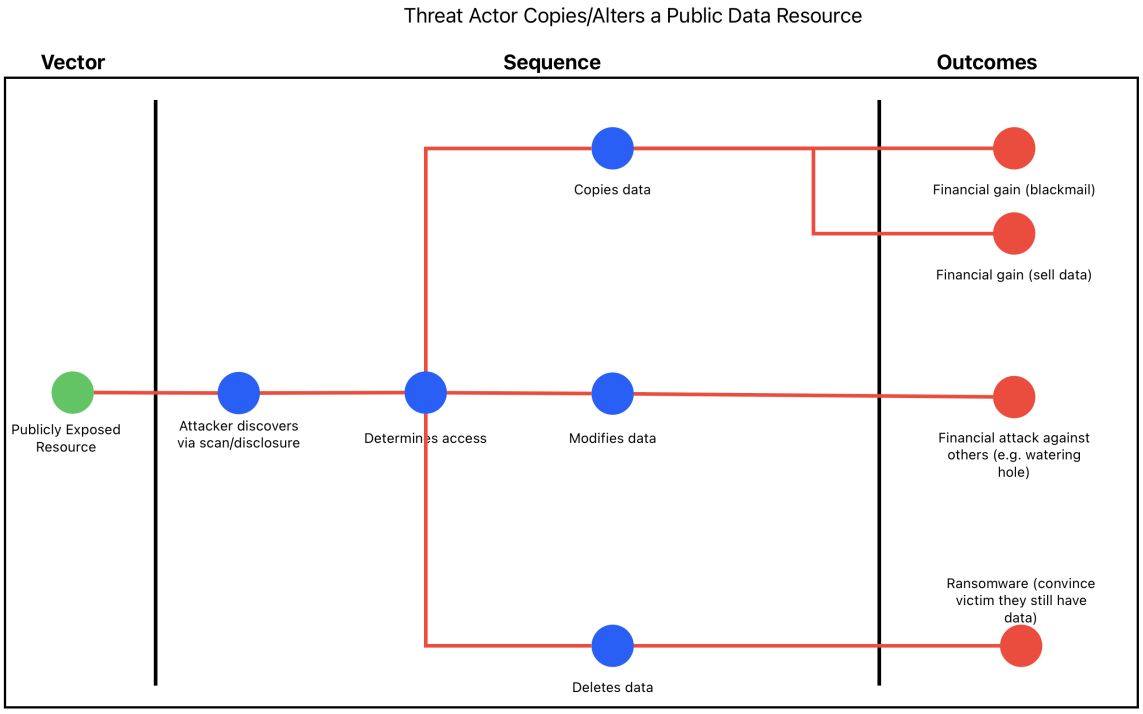
An attack sequence always starts with an initial vector. There is a many-to-many relationship between vectors and sequences because a vector could trigger multiple sequences, and a sequence may be triggered by multiple vectors.

Our goal for the UCTM is to highlight the top undifferentiated attack sequences — not every possible undifferentiated or differentiated sequence. Our philosophy is that organizations should reduce their risk of experiencing an undifferentiated attack first, since they are endemic and continuous due to attacker automation. Undifferentiated attacks also constitute the vast majority of security incidents for most organizations, as shown in nearly every threat and incident report, and confirmed by our own experiences. There's no use in worrying about expert burglars until you've stopped porch pirates.

We believe the following list covers the majority of attacks the majority of organizations will experience. *This is the "90/90" rule of cloud incident response: 90% of attacks in 90% of organizations are undifferentiated*. While these aren't quantitative numbers, they form a basis for guidance any organization can adopt. Here are the top 5 sequences most organizations encounter at some point:

1. A threat actor finds a public resource, and proceeds to copy or alter data.
2. A threat actor hijacks resources for cryptomining, spam, or phishing.
3. A threat actor uses credentials to perform a ransomware attack.
4. A threat actor finds an exposed system with a known vulnerability, uses that for lateral movement in the network or to steal credentials, and proceeds to find non-public data and exfiltrate, perform resource hijacking, or transitions to a directed/differentiated attack.
5. A threat actor discovers a dereferenced pointer (DNS, resource name, etc.) and proceeds to create the resource pointed to, then engages in achieving an objective.

Threat Actor Copies/Alters a Public Data Resource

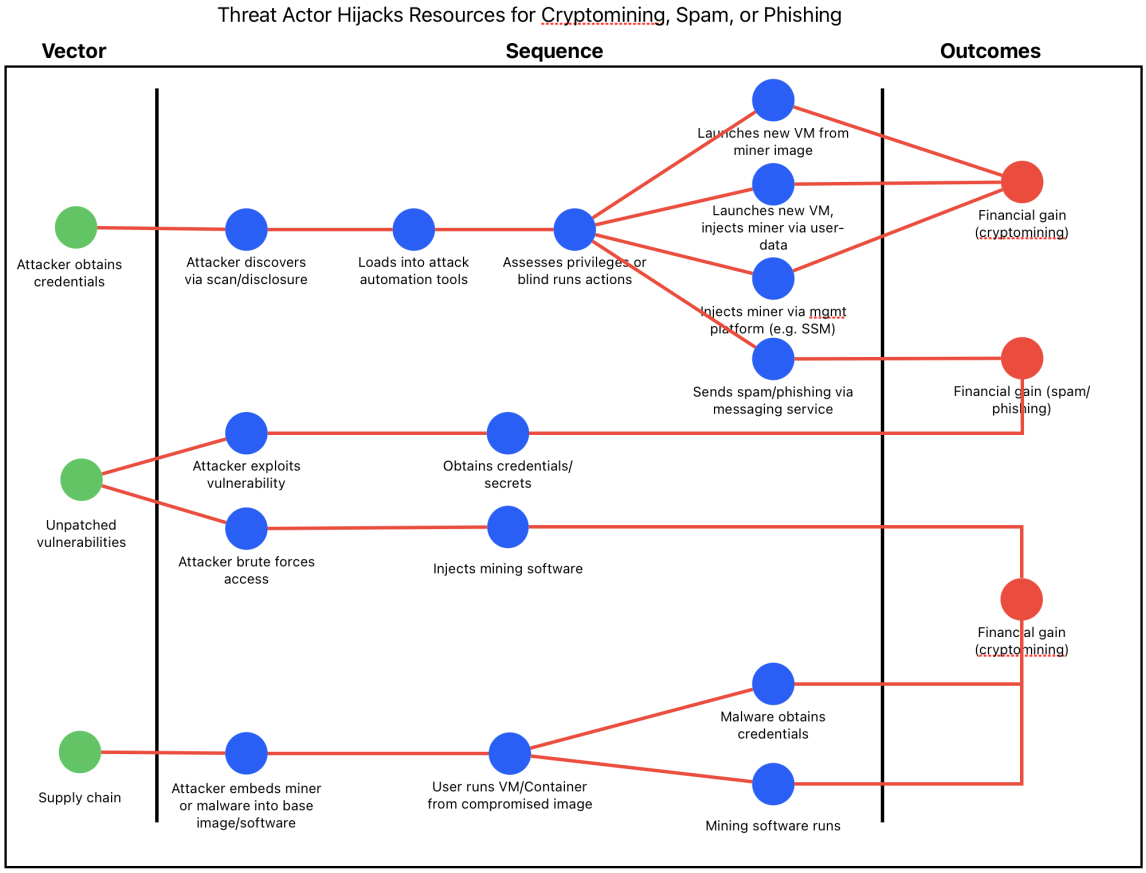


- **Vectors:** Publicly Exposed Resources
- **Targets**: Data
- **Objectives**:
    1. Financial Gain from Ransomware (encryption or deletion)
    2. Financial Gain from sensitive information disclosure (blackmail)
    3. Financial Gain from selling Sensitive Data on the black market
    4. Hacktivism
- **Sequence**:
    1. Attacker discovers a resource via scan or information sharing
    2. Attacker determines read and/or write access
        1. Attacker copies data
            - Objectives:
                - Financial Gain from sensitive information disclosure (blackmail)
                - Financial Gain from selling Sensitive Data on the black market
        2. Attacker modifies data
            - Objectives:
                - Leveraging Cloud infrastructure for financial attacks against others

- *E.g.,* injection of cryptomining malware via watering hole attack as seen in the LA Times incident.

3. Attacker deletes data
    - Objectives:
        - Convince the victim that attacker has a copy, and will return or decrypt it for ransom.

Threat Actor Hijacks Resources for Cryptomining, Spam, or Phishing



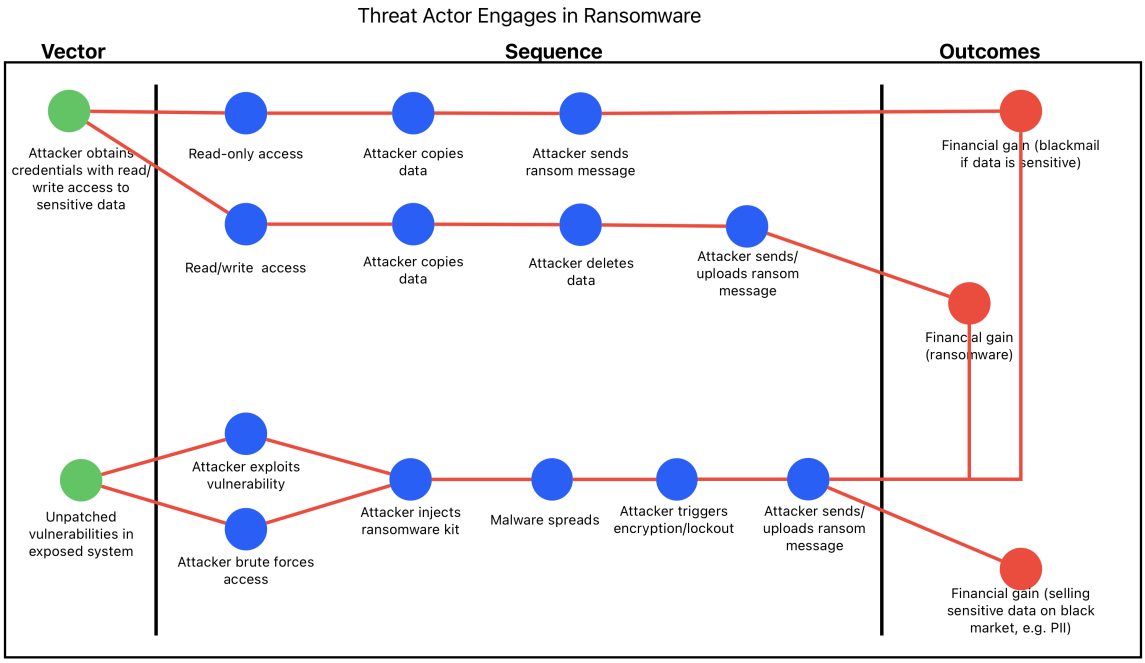
- **Vectors**:
    1. Lost, stolen, or exposed credentials
    2. Publicly exposed resources
    3. Credentials exposed via application security flaws
    4. Unpatched vulnerabilities and zero-days in exposed systems
    5. Supply chain
- **Target:** Compute, Cloud Services (E.g., SES, Pinpoint)
- **Objectives**:
    1. Financial Gain from resource hijacking
        1. Cryptomining

2. Spam/Phishing

2. Leveraging Cloud infrastructure for financial attacks against others

- **Sequence**:
    1. Vector: Attacker obtains credentials
        1. Attacker determines permissions
            - Attacker launches new VMs/containers based on image with mining software
                - Objectives:
                    - Financial gain via cryptomining
            - Attacker launches new VMs/containers and injects mining software via user-data
                - Objectives:
                    - Financial gain via cryptomining
            - Attacker injects mining software into running VM via a runtime management platform (*e.g.,* AWS SSM)
                - Objectives:
                    - Financial gain via cryptomining
            - Attacker sends spam/phishing email or other message
                - Objectives:
                    - Financial gain via spam/phishing
    2. Vector: Unpatched vulnerabilities and zero-days in exposed systems
        1. Attacker exploits vulnerability or weak/known username/password
        2. Attacker injects cryptomining software
            - Objectives:
                - Financial gain via cryptomining
        3. Attacker obtains credentials
            - Objectives:
                - Financial gain via spam/phishing
    3. Vector: Supply Chain
        1. Attacker embeds cryptomining software into a base image (VM or container)
        2. Attacker steals credentials via malicious package
        3. User runs workload on pre-owned image
            - Objectives:
                - Financial gain via cryptomining

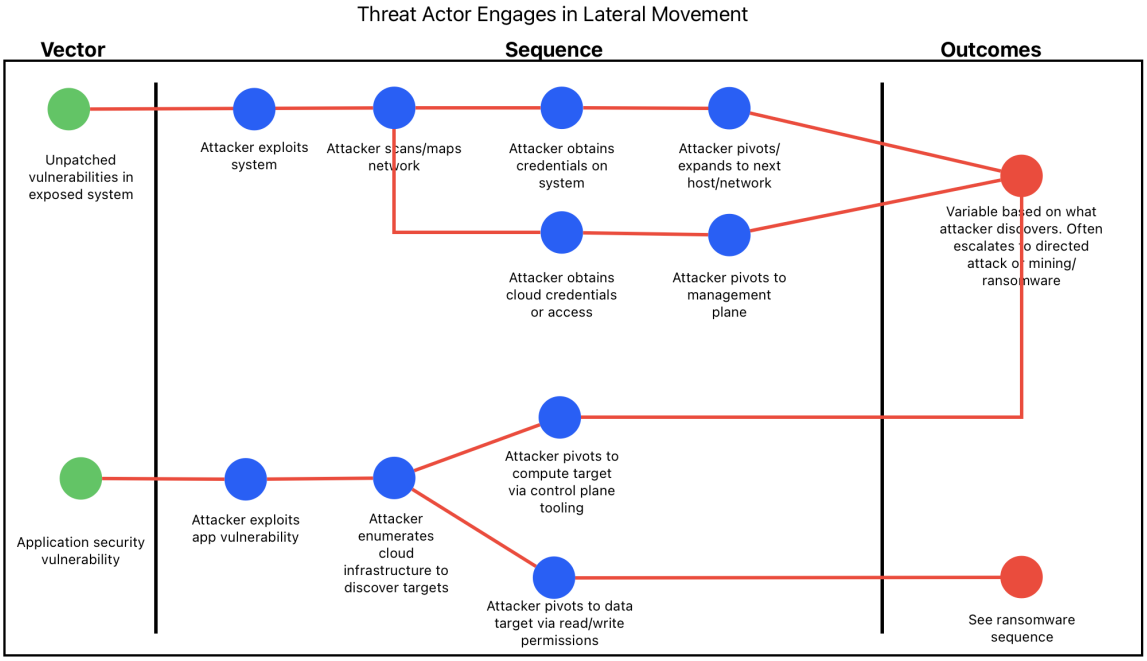
Threat Actor Engages in a Ransomware Attack



- **Vectors**:
  1. Lost, stolen, or exposed credentials
  2. Credentials exposed via application security flaws
  3. Unpatched vulnerabilities and zero-days in exposed systems
- **Targets**: Data
- **Objectives**:
  1. Financial Gain from ransomware (encryption or deletion)
  2. Financial Gain from sensitive information disclosure (blackmail)
  3. Financial Gain from selling sensitive data on the black market
- **Sequences**:
  1. Vector: Attacker obtains credentials with read/write access to storage with sensitive data
     1. Read-only access:
        - Attacker copies data
        - Objective:
          - Blackmail if the data is sensitive
     2. Read/Write access
        - Attacker copies data
        - Attacker deletes data or encrypts (rare — deletion is more common)
        - Attacker uploads ransom image/note

        - Objective:
            - Financial gain from ransomware
    2. Vector: Unpatched vulnerabilities and zero-days in exposed systems
        1. Attacker exploits vulnerability or weak/known username/password
        2. Attacker injects a ransomware kit
        3. Software scans and expands across resources within the network blast radius
        4. Attacker triggers host or data encryption
            - Objectives:
                - Financial Gain from ransomware (encryption or deletion)
                - Financial Gain from sensitive information disclosure (blackmail)
                - Financial Gain from selling sensitive data on the black market

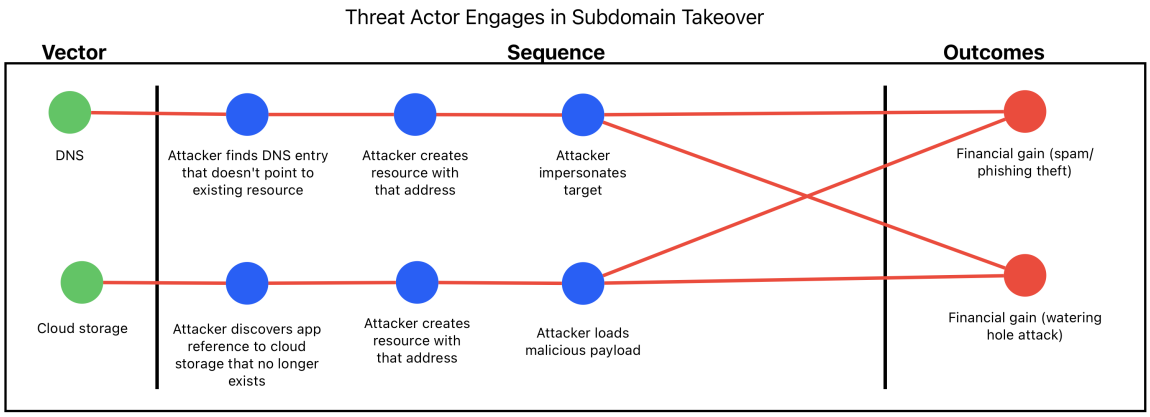
Threat Actor Engages in Lateral Movement for further attacks



- **Vectors**:
    1. Unpatched vulnerabilities and zero-days in exposed systems
    2. Credentials exposed via application security flaws
- **Targets:** Data, Compute, Application
- **Objectives**:
    1. Financial Gain
    2. Espionage

- **Sequences**:
    1. Vector: Unpatched vulnerabilities and zero-days in exposed systems
        1. Attacker enumerates network paths via network scanning
        2. Attacker seeks credentials to other systems
            - If cloud credentials are found, attacker pivots to management plane
                - Objectives:
                    - Variable, based on what the threat actor discovers. Often transitions to a directed attack or mining/ ransomware.
        3. Attacker pivots to next host/network
            - Objectives:
                - Variable, based on what the threat actor discovers. Often transitions to a directed attack or mining/ransomware.
    2. Vector**:** Cloud credentials exposed via application security flaws
        1. Attacker explores cloud infrastructure to discover network paths and targets
        2. Attacker uses cloud management tools to pivot from the control plane into target compute or storage resources
            - Objectives:
                - Variable, based on what the threat actor discovers. Often transitions to a directed attack.
                - If data access is obtained, attack transitions to ransomware

Threat Actor Engages in Subdomain Takeover



- **Vector:** Subdomain takeover
- **Targets:** Compute or Network Targets
- **Objectives**:
    - Leveraging Cloud Infrastructure for financial attacks against others
    - Leveraging Cloud Infrastructure for geopolitical attacks against others
    - Supply Chain Compromise

- **Sequences**:
  - Vector: DNS
    - Attacker finds a DNS entry that doesn't point to an active resource
    - Attacker creates a resource, which is pointed to by the DNS entry
    - Attacker impersonates the target for phishing
      - Objective:
        - Financial gain through spam/phishing
        - Financial gain via watering hole attack (typically cryptomining or ransomware)
  - Vector: Cloud Storage
    - The attacker finds a cloud storage bucket referenced within an application stack
    - Attacker creates a cloud storage bucket under their own control, referenced by the application stack
    - Attacker hosts malicious payload
      - Objectives:
        - Financial gain via watering hole attack (typically cryptomining or ransomware)
        - Financial gain through phishing: credential theft, selling PII, or selling financial information

Extending the Model

Some organizations face directed threat actors. These adversaries have specific objectives and targets. Once you move past undifferentiated threats into directed attacks, you need to extend this Universal Cloud Threat Model to your organization. At this point you need to hand it off to a dedicated team of experts, not just a DevOps engineer. Questions you want to address are:

1. Are there adversaries who would target us specifically?
2. What are their objectives and motivations?
3. What do we have in our cloud environment that could be a target?

From there identify which other attack vectors you need to consider. Attacks against your identities are more likely to occur when your organization is specifically targeted. These can come via credential stuffing or spraying attacks (as seen in the Midnight Blizzard attack against Microsoft), or phishing and smishing attacks against key personnel discovered via OSINT.

This is the point to transition to standard threat models, such as MITRE ATT&CK.

But remember that targets of directed attacks are still under attack by the many more opportunistic threat actors. Don't allow your extended cloud threat model to take focus away from the basics above.

Conclusion

As cloud security professionals continuously engaged in active research for over a decade, we understand that the vast majority of cloud attacks experienced by ourselves, our friends, and the companies we work with continue to fall into a very narrow slice of all possible cyber attacks. Most of these attacks can be prevented with a handful of basic practices. That doesn't mean there is anything easy or simple about cloud security — nothing simple scales — but it's easier to scale when you know which basics to focus on.

The information security industry has used the term "background radiation" for decades to describe automated attacks on Internet-accessible resources. Cloud computing hasn't changed this, but it *HAS* expanded the spectrum of these attacks. Attackers can now focus on the management plane and credentials. Even the nature of 'traditional' attacks, like exploiting OS vulnerabilities and ransomware, use techniques which have been updated to account for cloud deployments. Defending the management plane, a single Internet-accessible interface for essentially controlling entire fleets of data centers, requires new approaches now that we can't simply adjust firewall rules.

We found a need for a way of thinking which clarifies the new background conditions, and is designed to cut through the noise of all potential cloud attacks. We hope this can help defenders focus efforts on the fundamentals first, before they have to worry about the more sophisticated and obscure attacks that dominate industry news — but which organizations are much less likely to experience. Attackers always follow the path of least resistance before resorting to advanced capabilities.

This Universal Cloud Threat Model is intended to cover that 90%. Whether you are JP Morgan, Netflix, or a 6-day-old start-up the same actors, objectives, targets, and vectors apply. Protect yourself from the most active attacks first, then move onto more detailed threat models which work well for higher-complexity lower-frequency risks.

Appendix of Reports Enumerating Attack Vectors and Sequences

- KMSEC - Passive Takeover - uncovering (and emulating) an expensive subdomain takeover campaign
  - Vector: Subdomain takeover, leveraging passive DNS
- DarkLab - Trouble in Paradise
  - Vector: Unpatched vulnerabilities and zero-days in overly exposed systems
  - Vector: Credentials exposed via application security flaws
- CrowdStrike - Compromised Cloud Credentials Facilitate Widespread Lateral Movement (pg 36)
  - Vector: Credentials exposed via application security flaws
- Talos - Incident Response trends Q2 2023
  - Target: Data
  - Objective: Financial Gain
- Unit 42 - CloudKeys in the Air: Tracking Malicious Operations of Exposed IAM Keys
  - Vector: Lost, stolen, or exposed credentials
- AWS - The anatomy of ransomware event targeting data residing in Amazon S3
  - Vector: Lost, stolen, or exposed credentials
  - Target: Data Targets
  - Objective: Financial Gain from sensitive information disclosure (blackmail)
  - Objective: Financial Gain from selling Sensitive Data on the black market
- DataDog - Using malicious AWS activity to spot phishing campaigns
  - Vector: Lost, stolen, or exposed credentials
  - Target: Compute & Network Targets
  - Objective: Financial Gain from Phishing
  - Objective: Leveraging our Cloud Infrastructure for Financial attacks against others
- Stephan Berger - AWS Ransomware
  - Vector: Lost, stolen, or exposed credentials
  - Target: Data Targets
  - Objective: Financial Gain from Ransomware (deletion)
- Checkmarx - Hijacking S3 Buckets: New Attack Technique Exploited in the Wild by Supply Chain Attackers
  - Vector: Subdomain Takeover (unregistered S3 Bucket)
  - Target: Supply chain
  - Objective: unknown - the intermediate objective was credential collection
- Mandiant - Cloud Metadata Abuse by UNC2903
  - Vector: Unpatched vulnerabilities and zero-days in overly exposed systems
  - Target: Data Targets
  - Objective: unknown - threat actor was never attributed.
- Breaches.cloud - CommuteAir
  - Vector: Credentials exposed via application security flaws
  - Target: PII / Embarrassing Data
  - Threat Actor: Experimenter (reputation builders, 80s-style hackers)
  - Objective: Fame & Fortune
- Breaches.cloud - Codespaces (2014)
  - Vector: Credentials exposed via application security flaws
  - Target: Operational Data (via deletion)
  - Objective: Financial Gain from Ransomware (deletion)
- Breaches.cloud - LA Times Cryptomining
  - Vector: Publicly Exposed Resources
  - Target: Operational Data
  - Objective: Financial Gain from CryptoMining

About the Authors

About Chris

Chris is an experienced IT professional with a career spanning over 25 years. During this time he has focused on various areas including Linux, networking, and security. For the past ten years he has been deeply involved in public cloud and public cloud security in media and entertainment, leveraging his expertise to build and evolve multiple cloud security programs.

Chris is passionate about enabling the broader security team's objectives of secure design, incident response, and vulnerability management. He has developed cloud security standards and baselines to provide risk-based guidance to development and operations teams. As a practitioner he has architected and implemented numerous serverless and traditional cloud applications — focusing on deployment, security, operations, and financial modeling.

He is one of the organizers of the fwd:cloudsec conference, and presented at various AWS conferences and BSides events. He was named one of the inaugural AWS Security Heroes. Chris shares his insights on security and technology on social media platforms including Twitter, Mastodon, and https://www.chrisfarris.com.

About Rich

Rich is a Researcher and CEO of Securosis, and the SVP of Cloud Security at FireMon, where he focuses on leading-edge cloud security research and implementation. Rich joined FireMon through its acquisition of DisruptOps, a cloud security automation platform built on his research as CEO of Securosis. He has over 25 years of security experience and currently specializes in cloud security and DevSecOps, having started working hands-on in cloud over 10 years ago. He is an AWS Community Builder and the principal course designer of the Cloud Security Alliance CCSK training. He is primary author of the latest version of the CSA Security Guidance, has taught cloud security and incident response at Black Hat for over 10 years, and actively works on developing hands-on cloud security techniques. Prior to founding Securosis and DisruptOps, Rich was a Research Vice President at Gartner on the security team. Prior to his seven years at Gartner Rich worked as an independent consultant, web application developer, software development manager at the University of Colorado, and systems and network administrator.

Rich is the Security Editor of TidBITS and a frequent contributor to industry publications. He is a regular industry speaker at events including the RSA Security Conference, Black Hat, and DefCon; and has spoken on every continent except Antarctica (where he's happy to speak for free -- assuming travel is covered).

He writes at https://securosis.com and publishes a free, weekly cloud security training program at https://slaw.securosis.com. His email is rmogull@securosis.com, and yes, he gets a lot of spam, so use a creative subject line.