Vojnotehnicki glasnik/Military Technical Courier
ISSN: 0042-8469
ISSN: 2217-4753
vojnotehnicki.glasnik@mod.gov.rs
University of Defence
Serbia

Malicious code in the cloud

Damjanovi#, Dragan Z.
Malicious code in the cloud
Vojnotehnicki glasnik/Military Technical Courier, vol. 70, no. 3, 2022
University of Defence, Serbia
Available in: https://www.redalyc.org/articulo.oa?id=661772310013
DOI: https://doi.org/10.5937/vojtehg70-37168
http://www.vtg.mod.gov.rs/copyright-notice-and-self-archiving-policy.html

Review papers

Malicious code in the cloud

ВРЕДОНОСНЫЙ КОД В ОБЛАЧНОМ ХРАНИЛИЩЕ
ЗЛОНАМЕРНИ КОД У ОБЛАКУ

Dragan Z. Damjanović
Serbia
damjanovic1971@gmail.com
https://orcid.org/0000-0001-8169-4507

DOI: https://doi.org/10.5937/vojtehg70-37168
Redalyc: https://www.redalyc.org/articulo.oa?id=661772310013

Received: 25 March 2022
Accepted: 24 June 2022

Abstract:

Introduction/purpose: The paper analyzes the impact of malicious codes in the cloud. Malicious code is an unauthorized piece of code that violates the integrity of an application and infrastructure to cause certain effects, such as security breaches, spread of infections, and data infiltration from the computer with the help of malicious software - this is a simple form of data theft which can lead to disastrous consequences in all segments of society, especially when it comes to national security. To overcome this challenge, it is necessary to detect holes in the safety of cloud environments and repair them before the attackers use these vulnerabilities to bypass the integrated cloud infrastructure.
Methods: Structural analysis, functional analysis, comparative analysis, synthesis.
Results: There are many factors for collecting, comparing, and delivering intelligence data on cloud threats. Cloud applications are increasingly being targeted because their use to store and share data with mobile application hosting has been increased exponentially, enabling industrial automation and business information monitoring and procurement. In addition, billions of devices on the Internet use the cloud infrastructure as a background for processing and transmitting large data sets. Malicious code is easily distributed due to the ease of sharing documents and files via the cloud.
Conclusion: As cloud technologies are taking a central place in the world of digital transformation, the threat to the cloud environment is expected to grow exponentially. This means that organizations need to ensure that the cyber security position of the cloud infrastructure they possess is robust and mature enough to combat all relevant security threats in order to minimize business risks. Understanding the nature of practical security controls and how they are assessed enables organizations to build a practical approach to security and privacy in the cloud.

Keywords: malicious code, cloud, malware, intelligence.

Р е з ю м е :

Введение/цель: В данной статье анализируется влияние вредоносных кодов в облачном хранилище. Вредоносный код – это несанкционированный фрагмент кода, который нарушает целостность приложения и сетевой инфраструктуры, вызывая определенные последствия, такие как: нарушение безопасности, распространение инфекций и проникновение в компьютерные данные с помощью вредоносного программного обеспечения. Иными словами, речь идет о простой краже данных, которая может привести к неизгладимым последствиям во всех сферах общества, угрожая тем самым национальной безопасности. Для того чтобы преодолеть эту проблему, необходимо обнаружить дыры в безопасности облачных сред и устранить их до того, как злоумышленники воспользуются ими для обхода интеграции облачной инфраструктуры.
Методы: В статье использовались: структурно-функциональный анализ, сравнительный анализ и метод синтеза.
Результаты: Существует множество факторов для сбора, сравнения и предоставления разведывательной информации об облачных угрозах. Облачные приложения часто становятся мишенью, поскольку их использование для хранения и обмена данными с хостингом мобильных приложений растет в геометрической прогрессии, что позволяет осуществлять промышленную автоматизацию, мониторинг и сбор деловой информации. Кроме того, миллиарды устройств в интернете используют облачную инфраструктуру в качестве фона для обработки и передачи больших массивов данных. Вредоносный код легко распространяется из-за простоты обмена документами и файлами через облако.
Выводы: Поскольку облачные технологии занимают центральное место в мире цифровой трансформации, предполагается, что угроза облачной среде будет экспоциально расти. Это означает, что для того чтобы минимизировать бизнес-риски организациям необходимо обеспечить сверхнадежную систему кибербезопасности облачной инфраструктуры, которая сможет противостоять всем угрозам безопасности. Понимание характера практических средств контроля

безопасности и способов их оценки поможет организациям выработать практический подход к обеспечению безопасности и конфиденциальности в облаке.

Ключевые слова: вредоносный код, облако, вредоносное ПО, интеллект.

Abstract:

Увод/циљ: У раду је извршена анализа утицаја злонамерних кодова у облаку. Злонамерни код (малвер) неовлашћени је део кода који нарушава интегритет апликације и инфраструктуре како би изазвао одређене ефекте, као што су нарушавање безбедности, ширење инфекције и инфилтрација података са рачунара уз помоћ злонамерног софтвера (ради се о једноставнијој крађи података, која може довести до погубних последица у свим сегментима друштва, посебно када је у питању национална безбедност). Како би се овај изазов превазишао, неопходно је открити безбедносне рупе у окружењима у облаку и поправити их пре него што нападачи искористе ове недостатке и заобиђу интегритет инфраструктуре облака.
Методе: Структурна анализа, функционална анализа, компаративна анализа, синтеза.
Резултати: Постоји много метода за прикупљање, поређење и достављање обавештајних података о претњама у облаку. Клоуд апликације су све чешће на мети актера претњи, јер се употреба апликација у облаку за складиштење и дељење датотека са хостингом мобилних апликација, које омогућавају индустријску аутоматизацију, праћење и прикупљање пословних информација, експоненцијално повећала. Поред тога, милијарде уређаја на интернету користе инфраструктуру облака као позадину за обраду и пренос великих скупова података. Злонамерни код се лако дистрибуира због лакоће дељења докумената и датотека преко облака.
Закључак: Како технологије облака заузимају централно место у свету дигиталне трансформације, очекује се да ће претње окружењима у облаку експоненцијално расти. То значи да организације треба да обезбеде да сајбер безбедносна позиција инфраструктуре облака коју поседују буде довољно робустна и зрела за борбу против свих релевантних безбедносних претњи како би се минимизирали пословни ризици. Разумевање природе практичних безбедносних контрола и начина на који се оне процењују омогућавају организацијама да изграде практичан приступ безбедности и приватности у облаку.

Keywords: злонамерни код, облак, малвер, интелигенција.

Introduction

Malicious code (malver) is an unauthorized piece of code that violates the integrity of an application and infrastructure to cause certain effects, such as security breaches, the spread of infection, and data infiltration (it can lead to disastrous consequences in all segments of society, especially when it comes to national security). Attackers can use the cloud infrastructure to set up malicious code that performs malicious operations and unauthorized activities, such as spreading malware, filtering out sensitive information, and launching additional attacks. This malicious code can take many forms including scripts, add-ons, executables, binaries, and applets (any small application that performs a specific task within a larger program). (Sood, 2021)

Malicious code can (Hutchins et al, 2011):

- spread infections to a large number of Internet users,
- filter out sensitive and critical data from compromised systems,
- use additional online systems to spread infections within the network,
- install advanced malware, such as remote administration toolkits (RAT) and ransomware (Some types of rensomers can block a computer by creating a blackmail message that the user cannot remove without paying a ransom. Other types can encrypt files. In that case, the user whose computer is infected is asked to ransom in exchange for decrypting the data recorded on the disk),
- reconnoiter and collect information about the target environment,
- use endangered infrastructure for additional abuses, such as launching miners,
- arm compromised systems to act as launching platforms for targeted and widely based attacks, and
- violate the integrity of cloud-based cloud web sessions.

A complex code can combine many of these functions in collaboration for infection distribution and data filtering (Sood & Zeadally, 2016).

Malicious code distribution: Download attack model

It is important to understand the most prominent model for distributing malware and attacks on the Internet. Attackers typically opt for a download attack (Sood & Zeadally, 2016) that involves creating a socially-crafted "fishing" email to encourage a user to visit an attacker-controlled URL that distributes malware.

An attacker embeds a link in an email with a tempting (or even ominous) message to trick users into clicking on the link. Let us understand this model by dissecting its steps:

Step 1: An attacker sends an official email containing a built-in link to a malware download site. This is called a social engineering technique because it relies on the interests or emotions of users and tricks them into clicking on an embedded link.

Step 2: The user opens the link in the email and is redirected while the browser downloads the contents of the file hosted on the cloud infrastructure (applications, storage services or instances).

Step 3: The browser automatically downloads the malicious file located on the cloud infrastructure to the end user's system. Depending on the type of attack, an attacker can either force the browser directly to download a malicious executable file or download a maliciously created file that exploits a vulnerability in the browser to install the payload into the system. After a successful operation of the system, a dropper is installed in the system. (Droper is an intermediate file that installs the final malicious load on the system.)

Step 4: A droper loads malicious code into the user's system, which can (in some cases) bypass system security checks to perform unauthorized operations. Upon completion of these steps, a successful download attack was successfully achieved. The cloud infrastructure here acts as a launching platform for the distribution of infections (Sood & Zeadally, 2016).

Similarly, an attacker may host phishing sites (phishing or network identity theft is an attempt to steal Internet user data through a forged website) on the cloud infrastructure to steal credentials from the end user system (Sood, 2021).

Hosting a malicious code in cloud storage services

This allows us to understand the true picture of cloud infrastructure abuse, especially that of storage services.

Misuse of inherent storage service functionality

Attackers abuse cloud storage service functionality to host malicious codes and spread infections on the Internet. Attackers take advantage of cloud storage functionality either through free accounts or by using compromised accounts to host malicious code. Cloud storage services allow users to host files and share links with a specific set of users or more, that is, anyone who has a link to a file can download the file.

In addition, certain cloud services allow us to download files directly when the link is opened in a browser without any notification from the browser. Both of these features allow attackers to host a malicious code, make it public, and share a connection with large sections of Internet users. When the user downloads the connection, the file is automatically downloaded to the system. The first case study is a witty DNS malware. This shows how attackers can abuse the functionality of cloud storage service providers to host and distribute a malicious code.

The install.sh file is a bash shell file that, when executed, runs the specified commands. Looking at the contents of the install.sh file, we notice the URL for the cloud storage provider.

URL points to /brut.zip?dl=1. Due to its inherent functionality, the cloud service provider supports binary verification using dl as a parameter. If the dl value is set to 0, the browser downloads the file only after displaying a notification. If the dl value is set to 1, the browser automatically downloads it. This indicates that the attacker may force users to download the gross. zip without any user intervention (Sood, 2021).

Hosting scareware for social engineering

Scareware is another social engineering technique that allows an attacker to trick (or manipulate) users into believing they have to perform certain actions such as downloading files, providing specific information, opening additional links, or buying malicious software. This can either spread infections or extract sensitive information from the target user.

Attackers use intimidating software in conjunction with social engineering tricks to force users to perform actions by playing on their fears, such as sending computer virus notifications, indicating they will be subject to a tax audit or even pretend to be a bank infringer. Users must now re-authenticate or verify their account information. Modern attackers who carry out online scams largely use this intimidating code.

This example of scareware illustrates user fraud by causing fear of virus infection in the end user's system. This is potentially a phone scam, because the intimidating software asks the end user to call the specified number in order to get support and solve the virus problem in the end user's system. In reality, the real goal is to deceive the user. An important point is the distribution of this scareware through the cloud storage service (Sood, 2021).

Man-in-the-Browser (MitB)

Another client-side attack is Man-in-the-Browser (MitB), which attackers use to steal the credentials of cloud management accounts by installing malicious code on the end user's system. There are two variants of MitB malware. One involves installing malicious code on the system as an executable file, the other installs it in a browser as a browser add-on or extension.

Both variants of MitB are capable of bypassing browser functionality to perform unauthorized operations. These two models of attacks undermine the integrity of the browser by implementing hooks into the components of the browser and initiating a process to control the execution of the task, which ultimately leads to the theft of sensitive information.

Hooking (Sood & Enbody, 2011) is an inherent technique for controlling the execution behavior of running processes by intercepting the flow of communication, which changes the known behavior of the operating system. In this MitB model, the attacker has already installed malicious code into a system that has the ability to monitor communication that takes place from a browser.

Let us say a user opens a cloud management account from a browser. As malicious code is found in the system, it filters that traffic and implements hooks to redirect requests that the browser sends to the domain controlled by the attacker.

If you give credentials, the malicious code steals the credentials through a hook and leaves the original request to a legitimate server.

The response was received from the server and the communication was successful. This attack occurs on the end user's system before the request actually passes through the network. The manipulation is going smoothly and no one knows that the credentials for the cloud management accounts have already been stolen. This model reflects the MitB attack, as malicious code is capable of modifying or stealing browser communication.

There are, of course, other variations of the MitB attack.

Grabbing form (Sood et al, 2011): Malicious code searches for an HTML form that an application displays in a browser to request credentials. For example, it may display a web page to sign up for a cloud management account. When a user provides their credentials, malicious code makes a copy of the complete HTML form data, which is mostly an HTTP POST request, and transmits it to the domain managed by the attacker. As a result, letters of credit were stolen.

Inserting Content: Malicious code can easily insert unauthorized HTML content on the client's side and lead the user to believe that the content is legitimate. Let us say a user logs into a cloud management account through a browser. Malicious code can insert HTML content to trick the user into believing that the content is coming from a cloud server, but in fact, malicious code on the system injects unauthorized content into the HTTP response before it is displayed.

In addition to the above techniques, MitB malware can perform potentially catastrophic operations in active web sessions with the cloud management console.

We will show a few of them:

Stopping Elastic Cloud Compute (EC2) cloud instances,
Change inbound and outbound filtering rules to change communication settings,
Inserting malicious code into S3 bins and making it publicly available to spread malware,
Initiation of workloads for illegal crypto mining operations of bitcoin,
Data filtering via data backups and recordings,
Gaining access to private S3 baskets,
Deleting other user accounts,
Hosting phishing websites on cloud instances,
Hosting illegal services and advertising accordingly using newly created unauthorized instances, and
Synchronizing malicious files via cloud agents with storage services from compromised systems.

It is clear how significant MitB attacks are and have the inherent ability to abuse the integrity of the operating system and installed packages (Sood, 2021).

Cloud CLI exfiltration of stored credentials

Cloud administrators and engineers use Command Line Interface (CLI) tools to execute commands directly in the cloud infrastructure. This design gives them an easy way to perform operations. However, for CLI tools to work, they store credentials in a client-side figurative file. The local configuration is unencrypted and the credentials are stored in plain text on the end user's machine. If an attacker successfully installs malware, then it is easy to filter out all saved credentials for cloud management accounts. Installed malware can easily transfer credentials from a hidden .avs directory. Even in this attack mode, the malicious actor does not directly attack the cloud infrastructure. Instead, they first compromise the end-user system and then use stolen credentials to misuse the cloud infrastructure. In addition, they can also use the AVS CLI package to execute commands on behalf of users on the AVS account. As mentioned earlier, a malicious actor can perform countless operations to affect the environment in the cloud (Sood, 2021).

Synchronization token exfiltration via human cloud attack (MitC)

Man-in-the-Cloud (MitC) (Dulce & Shulman, 2015) synchronization token filtering (MitC) is another variant of the MitB attack, but in this scenario, malicious code installed on the end-user system has a built-in synchronization token targeting module used by various agents installed on end-user systems to synchronize

cloud files. As mentioned earlier, malicious code running in a compromised system can be very powerful in interacting with system software and running processes.

Numerous users install cloud vendor software agents to synchronize files present in dedicated directories with cloud storage. This allows the user to store the files in the appropriate directory, and the agent will automatically sync the files. For this, agents need a synchronization token to verify authentication and authorization for the cloud storage service before the data synchronization operation begins. To facilitate the synchronization process, the token is stored on the local machine so that the user does not have to enter a password every time the synchronization operation starts. This improves the ability of users to work seamlessly with the cloud and allows files to be synchronized in an automated way. If malicious code steals this token, then any device can sync and access files available in the cloud storage for cloud user accounts. Attackers use the MitC technique to filter out tokens and use tokens from various devices to gain access to files or synchronize malicious files to trigger chain infections. In some cases, malicious code can modify tokens to avoid detection as a result of missing tokens and trigger alerts. All in all, the MitC technique is an advanced approach that abuses the file synchronization mechanism using cloud agents that run the system.

Infecting virtual machines and containers

Attackers can choose different ways to infect VMs and containers to inject malicious code or misuse it to perform unauthorized cloud operations. Numerous attack models that we talked about earlier can contribute to the infection process, but there are some additional ways that attackers can go after targeting VMs and containers.

Exploiting vulnerabilities in network services

Launching misconfigured and unsecured containers and orchestration frameworks attracts threat actors, who then attack and use them for evil purposes. Docker containers (Cimpanu, 2020) and Kubernetes orchestration are often targeted by attackers via automated malicious code to steal information or launch other malicious useful content, depending on the design of the vulnerable component.

Inserting code into container images

Endangering the integrity of container images (Remillano, 2020) is another technique that attackers use to distribute malicious code. A number of developers use images and it is possible to insert malicious code into an image and distribute it. When developers retrieve and deploy a container image in a cloud environment, malicious code is activated and unauthorized operations are performed, such as scanning vulnerable dockers on the Internet or installing crypto miners.

Unsecured API endpoints

Unauthorized and insecure API endpoints in containers are the most prominent vectors for compromising containers and installing malicious code. Threat actors scan exposed API endpoints for container-based services and execute code to perform unwarranted operations. One such example is the malicious code Doki (Fishbein & Kajiloti, 2020) which scans unsecured Docker images and compromises them for evil activities on the Internet.

Secretly executing malicious code in VMs

VMs run as background processes without any visible element to end users. This means that there is no graphical user interface (GUI) for the VM and the user has no way to interact with the VM using the GUI. As most VMs share resources, such as disks and resources with the host OS, it is possible to misuse this design with specially crafted malicious code. One such example is Ragnar (Arghire, 2020) locker ransomware, which attackers distribute using a VM without performing ransomware operations by encrypting files on the host over a guest VM over shared resources.

Applying software without patches

One of the biggest security concerns is placing unpatched and outdated software in containers and VMs. Running code infiltrated with security vulnerabilities makes the cloud infrastructure vulnerable to exploitation - for example, running an insecure OS in VMs, placing vulnerable database software in containers, and so on.

This makes it much easier for attackers to exploit the inherent software and plant malicious code to perform illegal operations from the cloud infrastructure. In one case, unpatched Linux server software (Poston, 2020) was used by attackers to install persistent backdoor or malicious code to gain access to Linux servers.

Embedding malicious code through vulnerable applications

Deploying vulnerable applications to containers and VMs is one of the prominent vectors used by attackers to distribute malicious code. Applications that allow injection attacks, such as multi-site scripting (KSSS), structured query language (SKL), non-SKL (NoSKL), OS commands, Extensible Markup Language (KSML), and Simple Object Access Protocol SOAP), allow attackers to enter unverified useful data that is executed dynamically. After the successful execution of useful data, the code provided by the attacker is executed in the context of the application and unauthorized operations are performed. A recent study (Millman, 2020) highlighted an exponential increase in attacks on web applications where the CDN security provider blocked billions of web layer attacks.

Threat intelligence

Threat intelligence is defined as evidence-based knowledge that includes detailed system artifacts, events, compromise indicators (IoC), attack mechanisms, and potential risks for gaining detailed visibility of system status to proactively detect and prevent threats, including incidents. In general, evidence-based knowledge can be gathered only if there is sufficient insight into systems, networks, and overall infrastructure - including end-user behavior.

Cloud threat reporting

There are many factors for collecting, comparing, and committing cloud threat intelligence. A number of factors are:

Cloud applications are increasingly being targeted by threat actors because

- the use of cloud applications to store and share files with mobile application hosting, enabling industrial automation, monitoring and collecting business information, has increased exponentially,
- multiple cloud environments is seamlessly integrated for large-scale data transfer for sharing and productivity purposes, and
- billions of devices on the Internet use the cloud infrastructure as a backdrop to process and transmit large data sets.

Malicious code is easily distributed due to the ease of sharing documents and files via the cloud.
Malicious code is used to filter sensitive data from cloud instances.
Cloud infrastructure is often used for unauthorized operations such as cryptocurrency mining.
Detecting and preventing security breaches reduces business risks and potential damage to the brand.
Understanding the behavior of users who communicate with the cloud is used for fingerprints of suspicious and anomalous behaviors.
Violations of privacy and compliance may occur due to insecure application of controls.
The effectiveness of security controls is assessed in order to create defense against threats.

Based on these scenarios, it is vital to gain and import visibility into cloud infrastructure using organized threat alert operations (Sood, 2021).

Classification of threat intelligence

It is important to understand what we mean by the classification of "threat intelligence". In general, threat reporting includes contextual information from multiple resources needed to bring about information about the threats in particular environment, and then take appropriate action or precautionary measures accordingly. These actions are specific to detecting and preventing malware, as well as manual, targeted attack frames. Within the environment, it is possible to obtain and manage contextual data (granular event-related details) from multiple resources to generate threat intelligence.
These resources are:
In-house platforms - internal platforms for handling large-scale contextual data to build threat intelligence,
Enterprise platforms - platforms managed by third party organizations that provide contextual data, which can then be used directly in the internal platform, and
Platforms open source - community researchers use it to manage and provide contextual data in open source format, which can then be used directly in the internal platform to make informed decisions (Sood, 2021).

Basic threat intelligence classification model

Once contextual data has been received, different types of threat intelligence can be generated:

Strategic intelligence

- threat Intelligence to help make strategic and informed decisions by conducting high-level analysis and building risk profiles,

Operational intelligence

- threat intelligence related to the mode of operation of attacks (broad-based, targeted) and threat actors (attackers) associated with those attacks,

Tactical intelligence

- threat intelligence that reveals details of advanced and covert techniques, tactics and procedures adopted by threat actors to launch various attacks,

Technical intelligence

- threat intelligence covering technical aspects of threats, such as detection indicators that detect the functionality of malware in the system and the network level to build technical intelligence that can be incorporated into automated detection and prevention products (Sood, 2021).

Threat intelligence frameworks

In this section, we consider cyber threat intelligence frameworks that use a modular approach to implementing the various phases and building blocks of a mature threat intelligence platform.

Basic information for different cyber threat frameworks: DNI cyber threat framework. The US government has introduced a cyber threat framework (ODNI, 2021) to provide a consolidated approach to the classification and categorization of various cyber threats. This DNI framework is designed to provide a common language for describing the number of cyber threats and related suspicious activities. It also allows policy makers and researchers to communicate about threat events in a structured way so that appropriate action can be taken.

The framework emphasizes the rival life cycle, which consists of four phases: preparation, engagement, presence and consequences. In addition to these phases, the framework also explicitly relies on objectives, actions and indicators to detect threats and conflicting activities.

MITER ATT & CK Framework MITER Corporation provides the ATT & CK Framework (MITER, 2021) to highlight techniques, tactics and procedures adopted by opponents to launch either targeted or broad-based attacks, depending on the conditions. This box provides information that can be used to categorize various attacks and threats to be detected in particular environment as part of a threat notification platform. In essence, the latest version of ATT & CK illustrates the comprehensive paths of attack from reconnaissance to the persistence and exfiltration of various attacking entities.

This framework can be used in many ways, such as building threat intelligence logic, cyber risk analytics, enemy detection / prevention techniques, technology stack application, and automated attack assessment. The MITER framework for conducting cyber threat modeling (Bodeau, McCollum, Fox, 2018) can also be used to detect potential threats against cloud infrastructure in a proactive way. The framework enables the dissection of infrastructure and the implementation of threat modeling by supporting a variety of threat-focused approaches, systems and assets.

It supports attack characterization using a cyber defense framework in which risks can be categorized into devices, people, data, network, and applications. In addition, the risk associated with cloud infrastructure and how vulnerable the cloud environment is to threats and attacks can be calculated. Overall, this framework allows the application of threat information to detect unknown infections by adopting a single standard. Overall, both the DNI and MITER frameworks provide an efficient way to use different types of cloud threat information to design threat intelligence frameworks. These frames can be used directly or adapted to specific requirements.

Conceptual view of the threat notification platform

The threat notification platform is designed to enter raw data from multiple resources and process it to create intelligence that can be used to detect and prevent threats.

Data collection

This component is designed to enter large sets of raw data in the form of records, events, devices and CIDR lists from a wide range of hosts working in the infrastructure, including different types of networks and end-user devices. The goal is to collect data on a continuous basis and maintain it for processing. Data includes objects such as IP addresses, URLs, domains, file hashes, customer information, and complete billing for users and services. All types of records are entered, such as debugging and application execution, cloud service execution, access and communication protocols.

Data operations

When data is collected, it is passed to the next component for operations. The intention is to create a structural data format after performing normalization and duplicate removal operations to build a generic data format, remove duplicate records, and clear data entries with missing information. Once the data is normalized and cleaned, it is transformed into a structural format before validation and analytical operations are performed on it.

Certified Intelligence

This component handles validated threat intelligence from multiple sources, such as enterprise security tools deployed in the environment, enterprise classification feeds, malware family information, and open source threat sources to link to data operations and analysis machine. This is validated threat information that is used in conjunction with data from various organizational resources to build contextual threat intelligence.

Correlation and data analysis

Once the data structure is in place, it is time to perform data correlation and analysis using a variety of data science techniques, including machine learning and artificial intelligence, to link large amounts of data to detect anomalies and threats located in the organization of infrastructure. The goal is to detect threats that are in the system by analyzing raw data and using threat intelligence to reveal the time frame of the threat.

Contextual intelligence threats

Contextual Intelligence Threats (CTI) emphasizes threats that are found in systems in significant detail with the intention of showing business risks to the organization. CTI can provide very specific insights into the various assets used in infrastructure, including end users, and assess how prone these entities are to malware infections or are already infected with malicious code. This component also provides the ability to search for contextual intelligence for any particular entity (end user, system, and device). CTI can also be used for other purposes, such as performing risk mapping and proposing safety remedies. It may be particularly useful to specify areas of unacceptably high risk and exposure.

Understanding compromise and attack indicators

The Compromise Indicator (IoC) highlights data or metadata that reflects potential system compromise or the presence of threat actors in the environment, especially in this context, the cloud infrastructure. The IoC can help assemble the automated response needed to detect a threat in the environment so that appropriate prevention steps can be taken. Threat intelligence and security solutions import the IoC database to instruct tools to scan network data and endpoints from different systems in the infrastructure to detect network and endpoint threats, respectively. Another term used in the same context is Attack Indicators (IoA), which provides information regarding a potential attack that is ongoing or has previously been performed on cloud infrastructure. The primary difference between the IoC and the IoA is that the IoC indicates that a compromise has been reached, while the IoA reflects that the threat actor launched the attack, but there is no confirmation of a compromise. In order to obtain a detailed context on the security stance, an alert triggered by scans, assessments, and other security software must be linked using IoC and IoA to make specific calls about potential threats to the system and determine how they arose (Sood, 2021).

Understanding malware protection

It is crucial to apply inherent security protection in a proactive way to defend against malicious code and thus significantly reduce the impact and risk on the organization.

Proactive security mechanisms help prevent the spread of malicious code by detecting infections and stopping problems in early stages of infection. This helps to significantly reduce business risk, and thus reduce the occurrence of security breaches. The term "protection" here includes both "detection" and "prevention". This means that "malware protection" includes security mechanisms and strategies for the implementation of "malware detection" and "malware prevention" controls (Sood, 2021).

Malware detection

Controls that can be applied to detect malware in the cloud.

All cloud computing instances (hosts) should have a Host Intrusion Detection System (HIDS) installed that is able to do the following:

- Apply File Integrity Monitoring (FIM) to assess changes that occur in system files and maintain the state of the modified file. The goal is to check for file integrity violations on critical cloud servers.
- Detect anomalies using log analysis to build a risk attitude so that potential security risks can be analyzed. Detecting anomalies also helps identify potential attacks targeting cloud instances, such as brute force and cracking accounts. This technique is also called log-based intrusion detection (LID).
- Process and file level analysis to detect malicious code, such as rootkits running on the system. HIDS enables the detection of suspicious and hidden processes in critical cloud servers in order to detect possible infections.

All critical servers must have antivirus mechanisms installed to scan for malicious code (viruses, trojans, ransomware and rootkits) running on the system. Antivirus programs are updated at regular intervals with advanced signatures and heuristics to stay up to date to detect malicious code in the system. The antivirus engine has a built-in ability to scan documents, executables, mail and archive files to detect malicious code.

Scan files stored in cloud storage to detect potentially malicious code. By default, storage baskets do not have the built-in ability to check the nature of files. Either a third-party security solution or a cloud vendor-specific security service must be implemented to scan files in the storage bin for malware.

Implement an improved scanning process to dissect the contents of files uploaded to cloud services to detect the presence of malicious code. This content verification check must be enabled for each file upload feature in cloud applications.

Implement scanning of the content of specially embedded links and attachments for emails associated with cloud accounts, such as O365, to detect phishing attacks, such as

- embedded URLs that point to malicious domains for download attacks and
- attachments that contain malicious files resulting in malware installation.

Check the integrity and security of third-party applications integrated with cloud accounts for enhanced functionality to ensure that malicious files are not served through these third-party services.

Always scan network traffic for intrusion detection by dissecting network traffic and related protocols to detect command and control (C&C) communication, data exfiltration, and sensitive data leakage. In addition, scan the network traffic for malicious code that served as part of the download attack and the spread of the infection.

Mandatory implementation of a system for detecting suspicious behavior of end systems in relation to critical cloud services displayed on the Internet. For example, for attempts to retrieve accounts that target SSH and RDP services, the end customer sends multiple brute force requests and account breaches to gain access. The same system of behavior should detect a wide range of attacks and malicious code.

Perform periodic Azure authentication checks from Active Directory Federation Services (ADFS) to ensure that all authentication traffic flows properly through the ADFS instance and that no "golden SAML" cards have been created to bypass normal authentication (Vijayan, 2020).

Malware prevention

If any malicious files detected during the scanning process are implemented at the operating system level, ensure that the quarantine file takes place in an automated manner to avoid any interference. This helps filter out malicious files on the fly and restrict access to, share or transfer malicious files.

While uploading files to the cloud environment, i.e. application or storage services, if the file is found to be malicious in nature, it should be discarded, never stored in storage bins. This helps prevent malicious files from spreading after storage.

During the email scanning process, if malicious files are detected as part of an attachment or malicious URL, apply automated quarantine to filter emails that contain malicious content.

During the network scanning process, if intrusions are detected, make sure that the intrusion prevention system restricts malicious code and communication to prevent malicious code from reaching the end user's system via the cloud.

If the system detects a data leak during the on-line scan process in which the contents of the file are scanned to see if any sensitive data is present in the file, make sure the system restricts file sharing with other users and filter accordingly. A file containing sensitive data can be transferred as part of a data exfiltration process using malicious code.

Since systems detect suspicious communication using behavioral tracking, such as retrieval attempts, be sure to blacklist the end customer by limiting the IP address to prevent retrieval attacks. Make sure that all software running in the cloud has no vulnerabilities. If vulnerable packages or network services are found to be active, make sure patches are applied to remove vulnerabilities or poor configuration in the cloud environment. Make sure that there is a strong strategy for backing up and recovering the implementation in case of a ransomware attack. This helps administrators recover damaged data from backups at some point. In general, the detection and prevention of malware depend on each other to protect against malicious code in the cloud. This is because in order to prevent malicious code infections, they must first be detected. This means that gaining insight into the work of malicious code is the most important task. Once an understanding of the malicious code and how it affects the cloud infrastructure is gained, preventative solutions can be applied to completely disrupt the life cycle of the malicious code. Thus, a complete malware protection framework can be applied to prevent malicious use of cloud infrastructure (Sood, 2021).

Techniques, tactics, and procedures

Threat intelligence plays a significant role in building proactive and reactive security approaches in the fight against malicious code in the cloud. They also allow risk analysis to be performed to determine the level of risk associated with critical hosts, applications, and services deployed in the cloud. Threat intelligence also helps identify techniques, tactics, and procedures (TTPs) used by threat actors and malicious code. Using threat intelligence, mechanisms can be put in place to assess the effectiveness of security controls in the environment and to verify that the security stance is robust enough. Overall, it is an important condition to have an internal

threat alert platform to implement rigorous cloud infrastructure security procedures and processes. Applied intelligence on threats helps to prevent the abuse and exploitation of the cloud environment (Sood, 2021).

Conclusion

The attackers are targeting the cloud infrastructure to carry out cybercrime and vile Internet operations. The attackers use the cloud infrastructure for various attacks, such as distributing malicious code, launching crypto mining operations, running DDoS, filtering sensitive information and more. As cloud technologies take center stage in the world of digital transformation, threats to cloud environments are expected to grow exponentially. This means that organizations need to ensure that the cybersecurity position of the cloud infrastructure they possess is robust and mature enough to combat all relevant security threats so that business risks are minimized. To overcome this challenge, it is necessary to detect security holes in cloud environments and fix them before attackers take advantage of these shortcomings to circumvent the integrity of the cloud infrastructure. Understanding the nature of practical security controls and how to evaluate them enables organizations to build a practical approach to security and privacy in the cloud. There are no shortcuts to cloud security because it is an ongoing process which requires constant improvement how technology evolves.

References

Arghire, I. 2020. Ragnar Locker Ransomware Uses Virtual Machines for Evasion. *Security Week*, May 22 [online]. Available at: https://www.securityweek.com/ragnar-locker-ransomware-uses-virtual-machines-evasion [Accessed: 20 March 2022].

Cimpanu, C. 2020. Docker malware is now common, so devs need to take Docker security seriously. *ZDnet (Zero Day Blog),* November 30 [online]. Available at: https://www.zdnet.com/article/docker-malware-is-now-common-so-devs-need-to-take-docker-security-seriously/ [Accessed: 20 March 2022].

Dulce, S. & Shulman, A. 2015. Man in the Cloud Attacks. *Slideshare*, August 05 [online]. Available at: https://www.slideshare.net/Imperva/manintheclouddattacksfinal?from_action=save [Accessed: 20 March 2022].

Fishbein, N. & Kajiloti, M. 2020. Watch Your Containers: Doki Infecting Docker Servers in the Cloud. *Intezer*, July 28 [online]. Available at: https://www.intezer.com/blog/cloud-security/watch-your-containers-doki-infecting-docker-servers-in-the-cloud/ [Accessed: 20 March 2022].

Hutchins, E.M., Cloppert, M.J. & Amin, R.M. 2011. Amin Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. In: *ICIW 2011: 6th International Conferenceon i-Warfare and Security*, Washington, DC, pp.113-126, March 17-18 [online]. Available at: https://www.proceedings.com/16654.html [Accessed: 20 March 2022]. ISBN: 9781622766758.

Millman, R. 2020. Web app attacks are up 800% compared to 2019. *IT Pro*, November 23 [online]. Available at: https://www.itpro.com/security/357872/web-app-attacks-increase-2020 [Accessed: 20 March 2022].

Poston, H. 2020. Linux vulnerabilities: How unpatched servers lead to persistent backdoors. *Infosec*, September 23 [online]. Available at: https://resources.infosecinstitute.com/topic/linux-vulnerabilities-how-unpatched-servers-lead-to-persistent-backdoors/ [Accessed: 20 March 2022].

Remillano, A. 2020. Malicious Docker Hub Container Images Used for Cryptocurrency Mining. *Trend Micro*, August 19 [online]. Available at: https://www.trendmicro.com/vinfo/fr/security/news/virtualization-and-cloud/malicious-docker-hub-container-images-cryptocurrency-mining [Accessed: 20 March 2022].

Sood, A.K. 2021. *Empirical Cloud Security: Practical Intelligence to Evaluate Risks and Attacks*. Herndon, VA: Mercury Learning and Information. ISBN: 978-1683926856.

Sood, A.K. & Enbody, R.J. 2011. A browser malware taxonomy. *Virus Bulletin*, June 06 [online]. Available at: https://www.virusbulletin.com/virusbulletin/2011/06/browser-malware-taxonomy/ [Accessed: 20 March 2022].

Sood, A.K., Enbody, R.J. & Bansal, R. 2011. The art of stealing banking information – form grabbing on fire. *Virus Bulletin*, November 01 [online]. Available at: https://www.virusbulletin.com/virusbulletin/2011/11/art-stealing-banking-information-form-grabbing-fire [Accessed: 20 March 2022].

Sood, A.K. & Zeadally, S. 2016. Drive-By Download Attacks: A Comparative Study. *IT Professional*, 18(5), pp.18-25. Available at: https://doi.org/10.1109/MITP.2016.85.

Vijayan, J. 2020. SolarWinds Campaign Focuses Attention on 'Golden SAML' Attack Vector. *DARKReading*, December 22 [online]. Available at: https://www.darkreading.com/attacks-breaches/solarwinds-campaign-focuses-attention-on-golden-saml-attack-vector [Accessed: 20 March 2022].

Additional information

FIELD: IT

ARTICLE TYPE: Review paper

Alternative link

https://scindeks.ceon.rs/article.aspx?artid=0042-84692203734D (html)
https://aseestant.ceon.rs/index.php/vtg/article/view/37168 (pdf)
https://doaj.org/article/4c12298c93fc4fafa42ddf739aa6a0de (pdf)
https://elibrary.ru/item.asp?id=48719265 (pdf)
http://www.vtg.mod.gov.rs/archive/2022/military-technical-courier-3-2022.pdf (pdf)