CYBERARK

6 Key Use Cases for Securing Your Organization’s Cloud Workloads

CYBERARK®

6 Key Use Cases for Securing Your Organization's Cloud Workloads

Table of Contents

INTRODUCTION

The Continuing Rise of Cloud Adoption

Today, an increasing number of organizations are taking advantage of the benefits of cloud-based infrastructure by making the journey to public, private and hybrid cloud environments. Some that are further along in their cloud journeys leverage DevOps pipelines to increase their business agility, while others focus on cost savings and access to on-demand compute.

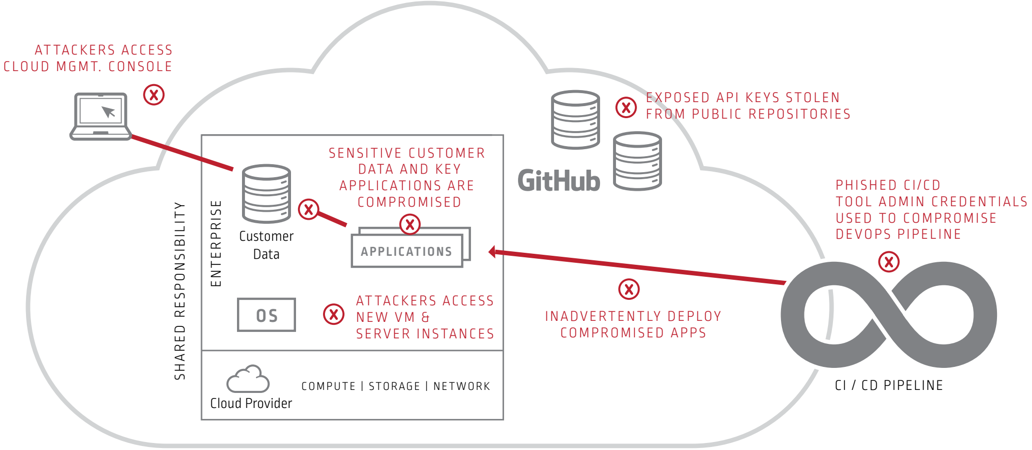
The business benefits are very real, but as more business-critical applications and services migrate to the cloud, ensuring the security of these cloud workloads becomes essential—as does the need to maintain and enforce enterprise-wide security policies in a consistent way.

As cloud vendors including AWS, Azure and Google make clear, security in the cloud is a shared responsibility. Though the public cloud vendors take great efforts to secure the cloud infrastructure—compute, storage, etc.—their customers are fully responsible for protecting basically everything above the hypervisor, including the operating system, applications, data, access to external resources and other assets and infrastructure. While the benefits of cloud computing are very real for organizations, so too are the responsibilities around ensuring the security of the organization's cloud workloads.

The vulnerabilities within the organization's cloud workloads are also very real, as some unfortunate organizations have found to their peril. The following examples represent some of the common use cases and approaches organizations should take to secure their cloud workloads.

Total worldwide spending on cloud services will reach $266 billion by 2021. Software as a Service (SaaS) dominates and will consume nearly 60% of cloud spending by 2021.*

EXAMPLES OF COMMON VULNERABILITIES IN THE CLOUD


ATTACKERS ACCESS CLOUD MGMT. CONSOLE
EXPOSED API KEYS STOLEN FROM PUBLIC REPOSITORIES
GitHub
SENSITIVE CUSTOMER DATA AND KEY APPLICATIONS ARE COMPROMISED
SHARED RESPONSIBILITY
ENTERPRISE
Customer Data
APPLICATIONS
OS
ATTACKERS ACCESS NEW VM & SERVER INSTANCES
Cloud Provider
COMPUTE | STORAGE | NETWORK
INADVERTENTLY DEPLOY COMPROMISED APPS
PHISHED CI/CD TOOL ADMIN CREDENTIALS USED TO COMPROMISE DEVOPS PIPELINE
CI / CD PIPELINE

Stat Source
* IDC, "Worldwide Semiannual Public Cloud Services Spending Guide," July 18, 2017

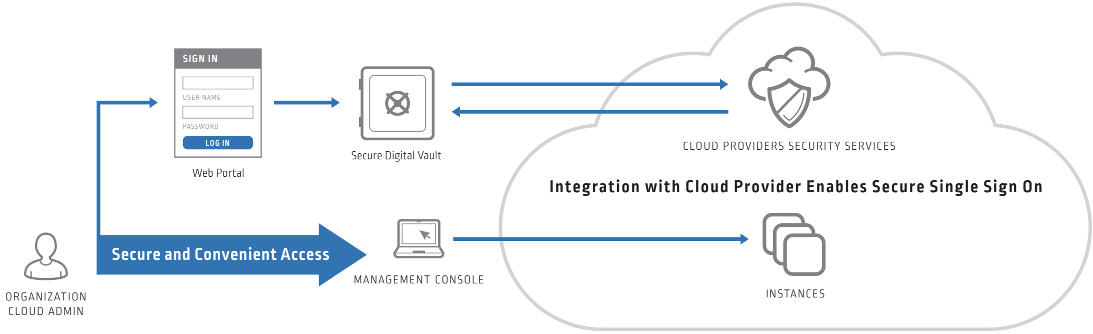
Securing the Cloud Environment: 6 Critical Use Cases to Consider

While each organization's cloud journey is different and unique to the organization, the following are common use cases that will likely need to be addressed to help ensure cloud workloads and infrastructures are secure.

Use Case 1: Secure the Management Console

Incredibly powerful cloud management consoles and portals enable complete management and control of an organization's cloud resources. They truly hold "the keys to the cloud kingdom." Accessed by both human and automated scripts (using API access keys), consoles are an attractive target for attackers.

SECURE AND CONVENIENT ACCESS TO THE CLOUD VENDOR'S MANAGEMENT CONSOLE


SIGN IN
USER NAME
PASSWORD
LOG IN
Web Portal
Secure Digital Vault
CLOUD PROVIDERS SECURITY SERVICES
Integration with Cloud Provider Enables Secure Single Sign On
ORGANIZATION CLOUD ADMIN
Secure and Convenient Access
MANAGEMENT CONSOLE
INSTANCES

The consequences of an attack can be significant. Unauthorized or uncontrolled access to the management console, directly or through APIs, can lead to data extraction or even a shutdown or takeover of the entire cloud environment. All use of the management console should be considered privileged access, and organizations should secure and monitor any and all potential access paths to the management console, including root accounts.

Root accounts

The root accounts are created when an organization initially sets up the accounts with the cloud provider. These accounts enable the most powerful access to the management console. Though many organizations use root infrequently, they still need to be protected, because once an attacker has root access the attacker essentially controls the organization's cloud infrastructure. Securing the root accounts in a digital vault has low operational impact, but it provides a significant improvement in security posture. As a best practice—as part of your organization's security policies and protocols—require multi-factor authentication (MFA) for root access, and monitor and record any sessions or activity involving root accounts.

Privileged access to cloud management consoles

Least privilege principles should be applied to all credentials associated with the management console. It is essential to isolate human privileged access to the management console via secure gateway (proxy) that runs the session, and assure that the credentials and the session information are not launched from the potentially vulnerable endpoint. Record sensitive sessions, and if access is set to require supervisor approval, grant access only for a limited period of time. Having security teams monitor active sessions in real-time will allow for sessions to be terminated when misuse or a potential attack is suspected. In addition, capabilities such as single sign-on (SSO) provide administrators and developers convenient access to the management console, without the need to know credentials.

MANAGEMENT CONSOLE SECURITY BEST PRACTICES

A	Treat all management console access as privileged, for both human and non-human users, and follow recommendations for protecting privileged accounts—apply least privilege, session isolation, rotation, session monitoring, etc.
B	Secure root account credentials in digital vaults and use MFA.
C	Ensure API and automated access, scripts, etc. are secure.
D	Consistently apply access policies to administrators that may have access to multiple cloud, on-premises and other environments (leverage AD credentials, etc.).

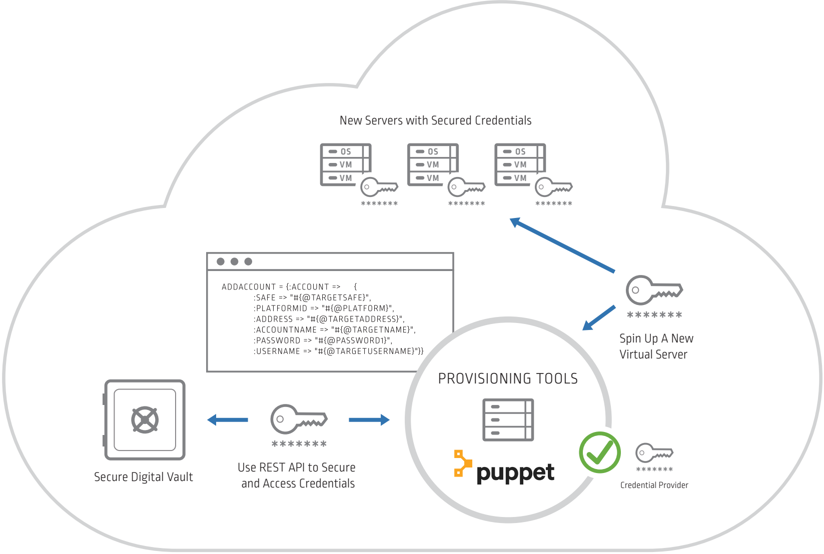
Use Case 2: Secure Your Organization's Cloud Infrastructure

Cloud-based infrastructure enables new virtual servers, data stores, containers and other resources to be provisioned as needed. When each virtual server or infrastructure resource is initiated and launched, it will be assigned privileged credentials that must be secured.

In more static environments, administrators may use the management console to spin up and assign a new server—whereas in dynamic environments, automation, scripts and provisioning tools may be used to automatically establish new compute instances. Consequently, credentials can be created at a rapid rate as organizations spin up servers to be used for just a few minutes or hours to complete a specific task, multiple times a day.

By automating the provisioning of new cloud server instances and using the REST APIs to secure and retrieve credentials in a secure digital vault, your organization can ensure that the credentials associated with the cloud infrastructure are secure when the resource is provisioned.

SECURE AUTOMATED PROVISIONING USING TOOL INTEGRATIONS AND APIs


New Servers with Secured Credentials
OS
VM
VM
****
Spin Up A New Virtual Server
PROVISIONING TOOLS
puppet
Credential Provider
Secure Digital Vault
Use REST API to Secure and Access Credentials

```
ADDACCOUNT = {:ACCOUNT =>     {
           :SAFE => "#{@TARGETSAFE}",
           :PLATFORMID => "#{@PLATFORM}",
           :ADDRESS => "#{@TARGETADDRESS}",
           :ACCOUNTNAME => "#{@TARGETNAME}",
           :PASSWORD => "#{@PASSWORD1}",
           :USERNAME => "#{@TARGETUSERNAME}"}}
```

CLOUD INFRASTRUCTURE SECURITY BEST PRACTICES

A	Immediately secure the privileged credentials associated with newly provisioned infrastructure.
B	Manage the infrastructure credentials using the principles of least privilege management.
C	Remove privileges when the infrastructure is de-provisioned.

Use Case 3: Secure API Access Keys

API access keys are widely used to enable programmatic requests to the cloud environment, such as stopping or starting a server, provisioning a container or even wiping a database. While automation enables organizations to leverage the dynamic capabilities of the cloud to the fullest, automation also relies on scripts, orchestration servers and other automation tools that contain API access keys—which can increase vulnerabilities and potentially enable unrestricted access to the cloud infrastructure.

Because API access keys are powerful credentials and used widely, securing them and applying the principle of least privilege is imperative. Once an attacker has API access keys, the attacker can gain unrestricted access to the entire cloud environment.

API KEY SECURITY BEST PRACTICES

A Remove all embedded API keys and secrets from scripts, automation tools, etc.

B Never provide human users direct access to API keys.

C Secure all API keys in a secure digital vault, allowing only authorized users and applications to access them and follow principles of least privilege—for both human and automated users.

D Leverage API access to the digital vault, using integrations with automation tools and scripts, etc. to automate and ensure the secure use of API access keys.

Attackers have used access keys, either stolen from phished endpoints or inadvertently posted to public code repositories and other sources, to steal customer data, destructively delete source code and other intellectual property and create other havoc across organizations.

Use Case 4: Secure DevOps Pipeline Admin Consoles and Tools

The DevOps pipeline enables organizations to increase business agility by reducing time-to-deployment and getting new applications and services into production faster.*

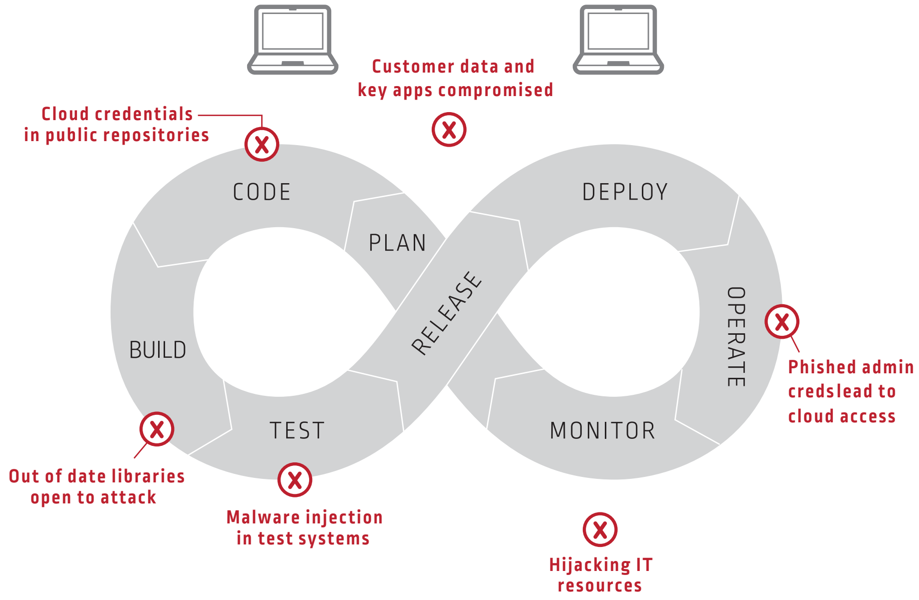
Powerful tools enable services to be automatically built and deployed. Often code is pulled from public repositories and then pushed into production—multiple times each day, with minimal human intervention. The automated pipeline enables applications to be deployed at an incredibly rapid pace.

Organizations typically use a variety of tools throughout the pipeline.
Functions range from configuration management to source control to IT automation and more. Because the tools can vary greatly, each often takes a different approach to managing credential access. Consequently, the workflows for controlling access to the secrets and privileged user accounts linked to those tools can also vary greatly. Unfortunately, this often leads to inconsistent, and even manual, strategies for controlling access to critical infrastructure—which opens organizations up to vulnerabilities.

Frequently, it's automated tools and scripts, rather than human users, accessing tools.
Regardless, organizations must secure all DevOps tools. It's important to know what tools are used—and by whom—and also where automation is employed. Organizations should maintain a single security posture by identifying and authorizing all DevOps tools under a common policy.

*DevOps is used in a range of environments, including cloud and on-premises. Use Case 5 focuses on how applications, scripts and automation tools securely interact in a dynamic cloud environment. Use Case 4 secures the powerful tools that make it possible.

DEVOPS PIPELINES: EXAMPLE ATTACK TYPES



DEVOPS PIPELINE ADMIN CONSOLE AND TOOLS SECURITY BEST PRACTICES

A Secure access to all tool admin accounts and consoles. Apply principles of least privilege, or JIT (Just in Time) privilege. Rotate, monitor and record key actions for human and automated users, etc.

B Apply consistent security policies, under a common enterprise-wide policy, to the tools and admin consoles used at each stage of the pipeline—governing who and what can use, configure, etc., access to API keys.

C For hybrid environments, work with systems of record—including existing privileged account security, privileged access management and directory services—to create one view of all of the security and privileged access across the entire infrastructure.

D Continually perform audits to learn and improve the organization's security posture.

Use Case 5: Secure DevOps Pipeline Code

Securing DevOps tools alone is not enough to protect the pipeline. The other critical area that must be secured is the code that “flows” through the pipeline. Enterprise cloud environments typically rely on a large numbers of app credentials and access keys to run applications, scripts, databases, servers, etc.

As a result, cloud applications and hybrid assets such as CRM, order-processing systems and scripts with APIs all present vulnerabilities—particularly when they use hardcoded passwords and credentials to gain access to customer data, web apps or on-premises servers. Consequences can be very serious when, for example, source code inadvertently includes hardcoded access keys to the organization’s cloud environment and is placed back into a public code repository.

Avoid Hardcoding

Hardcoding and embedding credentials in code or a script creates significant risks, because attackers can also easily access them—especially, if the credentials are in clear text. Hardcoded credentials are:

- Nearly impossible to rotate, making them an easy static target for attackers
- Difficult to track because many scripts, automation tools, applications and humans have access to them
- Difficult to monitor or assign accountability to the applications without centralized credentials management
- Frequently pursued by hackers because application credentials can be used to gain access to mission-critical systems

APPLICATIONS AND CODE SECURITY BEST PRACTICES

A Remove all secrets, keys and credentials from source code, configuration management, etc., and centralize access in a secure digital vault so that each virtual machine has unique access to keys when it is initiated.

B Apply the principles of least privilege, JIT privilege, etc.

C Monitor, control and rotate credentials according to enterprise security policies.

D Support auto-scaling and dynamic environments by automatically securing credentials when new instances are created.

E Avoid delays and frustration by including security at the start of the development process. Be proactive—don’t wait to retrofit security as new code is scheduled to go into production.

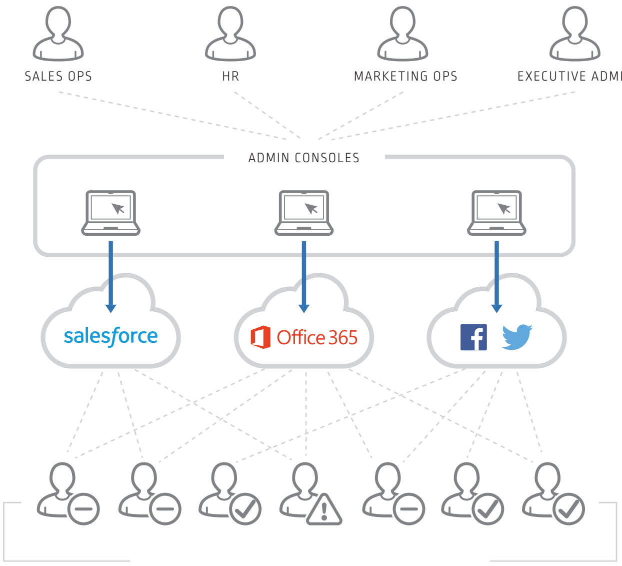
F Audit everything to continually learn and improve.

Use Case 6: Secure Admin Accounts for SaaS Applications

While many enterprises use SaaS business applications such as SalesForce, Microsoft Office 365 or SaaS-based social media such as Twitter and Facebook, the critical need to secure the administrative consoles for these cloud-based applications is not always fully recognized until there's a problem—such as publically posted corporate data or a hacker's Tweet on the corporate account.

SaaS admin consoles are often used by enterprise administrators to grant access to individual users, such as a sales exec for Salesforce, or to establish a common shared account for social media or another shared business application. This means that SaaS applications are routinely used by people who operate outside of the traditional IT organization and don't necessarily follow typical security protocols. All too frequently, SaaS passwords are simple, shared by multiple users, written down and unchanged for periods of time. This often results in a proliferation of users with credentials—which makes it difficult to track and account for legitimate usage. Without a systematic approach, undetected access to data can continue for long periods of time, putting corporate data and corporate reputations at risk.

A VARIETY OF USERS ACCESS ADMIN ACCOUNTS FOR SAAS APPLICATIONS



SAAS ACCOUNT SECURITY BEST PRACTICES

A Treat the administrative accounts for SaaS applications as privileged accounts and apply all the principles of privileged access to them.

B Where possible, leverage AD groups for application provisioning /de-provisioning so that SaaS console access is automatically removed when employees leave the organization.

C Use session isolation and monitoring and recording for sensitive business applications such as payroll systems.

Approaching Security with a Deeper Understanding of the Unique Cloud Challenges

Clearly, moving workloads to the cloud can bring significant business benefits, but it also expands the attack surface and allows unprotected privileged accounts, credentials and secrets to become damaging security vulnerabilities.

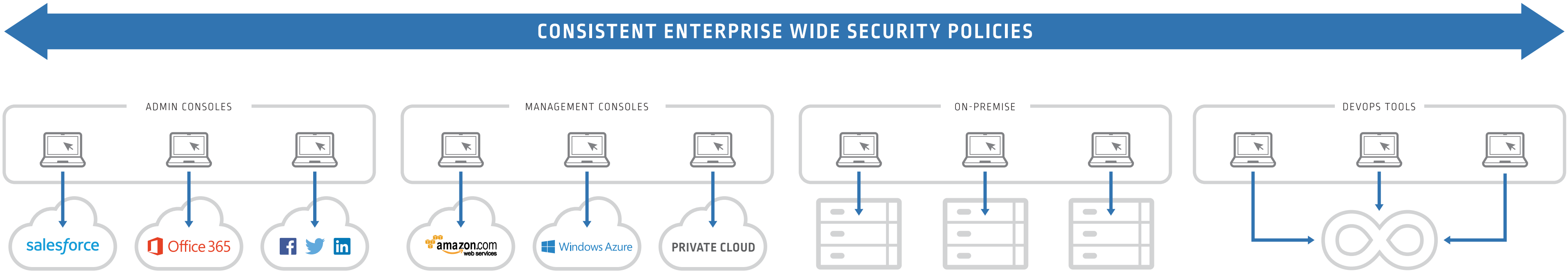
To protect cloud workloads, organizations must understand the unique security challenges that cloud environments and automation present—and take steps to address them head-on.

Hybrid environments must be secured.
While the use case examples in this ebook focus on protecting cloud environments and workloads, the best practices also apply to hybrid environments. One of the unique challenges of hybrid environments is the need to secure both the on-premises and cloud environments, and this can be even more challenging when organizations use multiple cloud vendors and operate multiple on-premises environments. Security is critical—attackers have, in some cases, successfully moved laterally from cloud to on-premises environments.

Security policies should be enforced consistently.
IT and security leaders must consistently enforce security and access policies across their entire organization—regardless of the compute environment, development philosophy or complexity. This must be done at scale, today and as the organization and its cloud infrastructure evolve. This can be achieved with a single point of control that enables consistent management of privileged accounts, credentials and secrets across each of the compute and development environments.

CyberArk can help organizations throughout their cloud journeys.
No other vendor has the deep cloud experience and offers a comparable breadth of privileged account security for on-premises, cloud and DevOps environments. The CyberArk Privileged Account Security Solution supports organizations across their entire cloud journey—from securing an initial cloud project in a hybrid environment to fully embracing the cloud and DevOps.

ORGANIZATIONS WANT CONSISTENT "ENTERPRISE-WIDE" SECURITY POLICIES



To learn more about how CyberArk can help your organization secure its cloud workloads and environments, visit cyberark.com/cloud or request more information online.

CyberArk believes the information in this document is accurate as of its publication date. The information is provided without any express, statutory, or implied warranties and is subject to change without notice.

THIS PUBLICATION IS FOR INFORMATIONAL PURPOSES ONLY AND IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER WHETHER EXPRESSED OR IMPLIED, INCLUDING WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, NON-INFRINGEMENT OR OTHERWISE. IN NO EVENT SHALL CYBERARK BE LIABLE FOR ANY DAMAGES WHATSOEVER, AND IN PARTICULAR CYBERARK SHALL NOT BE LIABLE FOR DIRECT, SPECIAL, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, OR DAMAGES FOR LOST PROFITS, LOSS OF REVENUE OR LOSS OF USE, COST OF REPLACEMENT GOODS, LOSS OR DAMAGE TO DATA ARISING FROM USE OF OR IN RELIANCE ON THIS PUBLICATION, EVEN IF CYBERARK HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

U.S., 08.17. Doc # 169

CYBERARK®