- Filename: eccouncil-ceh31250-v10-19-2-1-cloud-computing-attacks
- Show Name: CEHv10 (312-50)
- Topic Name: Cloud Computing
- Episode Name: Cloud Computing Attacks
- Description: In this episode, Adam and Wes discuss the vulnerabilities and attacks that can lead to Cloud systems being compromised.

==

CEH v10 - Module 19 - Cloud Computing

What is Cloud Computing? - “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

What are the key Cloud Computing characteristics?

1. On-demand self-service - A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

2. Broad network access - Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

3. Resource pooling - The provider’s computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

4. Rapid elasticity - Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

5. Measured service - Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

What are the accepted Cloud Service Models?

Infrastructure as a Service (IaaS) - The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Platform as a Service (PaaS) - The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Software as a Service (SaaS) - The capability provided to the consumer is to use the provider’s applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings.

What are the accepted Cloud Deployment Models?

Private cloud - The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud - The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

Public cloud - The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid cloud - The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

The NIST cloud computing reference architecture (NIST SP 500-292) defines five major actors: cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker. Each actor is an entity (a person or an organization) that participates in a transaction or process and/or performs tasks in cloud computing.

Actor Definition

Cloud Consumer A person or organization that maintains a business relationship with, and uses service from, Cloud Providers.

Cloud Provider A person, organization, or entity responsible for making a service available to interested parties.

Cloud Auditor A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.

Cloud Broker An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.

Cloud Carrier An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

What are the threats to Cloud Computing that I should be aware of?

ALL OF THEM !!! :)

Mostly the same threats that we need to worry about in a traditional non-cloud infrastructure environment PLUS cloud specific threats:

1. Data breach/loss
2. Abuse of cloud services
3. Insecure interfaces/API's
4. Insufficient due diligence
5. Insufficient infrastructure design/planning
6. Multi-tenant issues

7. Unknown risk profiles
8. Unsynchronized system clocks
9. Loss of access to operational/security logs
10. Hardening of infrastructure conflicting with cloud environment
11. Malicious insiders
12. Inappropriate access to cloud resources
13. Negative impact to business due to co-tenant activities
14. Hardware failure
15. Supply-chain failure
16. Exposure/modification of cloud traffic
17. Isolation failure
18. Cloud provider acquisition
19. Management network failure/interface compromise
20. Loss/compromise of encryption keys
21. Vendor lock-in
22. Virtual Machine vulnerabilities
23. Jurisdictional issues based on changing geographic boundaries
24. Theft of infrastructure from cloud data center
25. Cloud service termination/failure
26. E-discovery/subpoena
27. Improper/incomplete data handling & disposal
28. Loss/modification of backup data
29. Compliance risk
30. DoS/DDoS attacks based on resource consumption & overcharges

What are Cloud Computing attacks that I should be aware of?

1. Service hijacking via Social Engineering & network sniffing

2. Session hijacking using Cross Site Scripting (XSS)

3. DNS attacks

4. Side channel attacks

5. Cross VM attacks

6. SQL injection

7. Cryptanalysis attacks

8. Wrapping attacks - performed during the translation of SOAP messages in the TLS layer; attackers duplicate the body of the message and send it to the targeted server impersonating the legitimate user

9. DoS/DDoS attacks

10. Man-in-the-Cloud attacks - abuse of cloud file synchronization services by tricking the user into installing malicious software that places the attacker's synchronization token for the service on their machine, allowing the attacker to steal the user's token and gain access to their files

What are the OWASP Top 10 Application Security Risks - 2017?

A1 - Injection: Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

```
a. Input Validation
b. Limit Account Privileges
```

A2 - Broken Authentication: Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

A3 - Sensitive Data Exposure: Attackers may steal or modify weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

A4 - XML External Entities (XXE): Many older or poorly configured XML processors
evaluate external entity references within XML documents. External entities can
be used to disclose internal files using the file URI handler, internal file
shares, internal port scanning, remote code execution, and denial of service
attacks.

Applications and in particular XML-based web services or downstream integrations
might be vulnerable to attack if:

```
• The application accepts XML directly or XML uploads, especially from
```

untrusted sources, or inserts untrusted data into XML documents, which is then
parsed by an XML processor.

```
• Any of the XML processors in the application or SOAP based web services has
```

document type definitions (DTDs) enabled. As the exact mechanism for disabling
DTD processing varies by processor, it is good practice to consult a reference
such as the OWASP Cheat Sheet 'XXE Prevention’.

```
• If your application uses SAML for identity processing within federated
```

security or single sign on (SSO) purposes. SAML uses XML for identity
assertions, and may be vulnerable.

```
• If the application uses SOAP prior to version 1.2, it is likely susceptible
```

to XXE attacks if XML entities are being passed to the SOAP framework.

NOTE: Being vulnerable to XXE attacks likely means that the application is
vulnerable to denial of service attacks including the Billion Laughs attack.

A5 - Broken Access Control: Restrictions on what authenticated users are allowed
to do are often not properly enforced. Attackers can exploit these flaws to
access unauthorized functionality and/or data, such as access other users'
accounts, view sensitive files, modify other users’ data, change access rights,
etc...

A6 - Security Misconfiguration: The most commonly seen issue. The result of
insecure default configurations, incomplete or ad hoc configurations, open cloud
storage, misconfigured HTTP headers, and verbose error messages containing
sensitive information. Not only must all operating systems, frameworks,
libraries, and applications be securely configured, but they must be patched and
upgraded in a timely fashion.

A7 - Cross-Site Scripting (XSS) - XSS flaws occur whenever an application
includes untrusted data in a new web page without proper validation or escaping,
or updates an existing web page with user-supplied data using a browser API that
can create HTML or JavaScript. XSS allows attackers to execute scripts in the
victim’s browser which can hijack user sessions, deface web sites, or redirect
the user to malicious sites.

There are three forms of XSS, usually targeting users' browsers:

```
1. Reflected XSS - The application or API includes unvalidated and unescaped
```

user input as part of HTML output. A successful attack can allow the attacker
to execute arbitrary HTML and JavaScript in the victim’s browser. Typically
the user will need to interact with some malicious link that points to an
attacker controlled page, such as malicious watering hole websites,
advertisements, or similar.

```
2. Stored XSS - The application or API stores unsanitized user input that is
```

viewed at a later time by another user or an administrator. Stored XSS is
often considered a high or critical risk.

```
3. DOM XSS - JavaScript frameworks, single-page applications, and APIs that
```

dynamically include attacker-controllable data to a page are vulnerable to DOM XSS. Ideally, the application would not send attacker-controllable data to unsafe JavaScript APIs.

Typical XSS attacks include session stealing, account takeover, MFA bypass, DOM node replacement or defacement (such as trojan login panels), attacks against the user's browser such as malicious software downloads, key logging, and other client-side attacks.

A8 - Insecure Deserialization: Often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

Serialization is the process of translating data structures or object state into a format that can be stored or transmitted and reconstructed later. When the resulting series of bits is reread according to the serialization format, it can be used to create a semantically identical clone of the original object.

A9 - Using Components with Known Vulnerabilities: Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

A10 - Insufficient Logging & Monitoring: Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

Additional Attacks to be concerned about:

```
1. Directory Traversal (../) - moving from one directory to others without the
```

knowledge of the system owner

```
2. Cross-Site Request Forgery (CSRF) - forces a logged-on victim’s browser to
```

send a forged HTTP request, including the victim’s session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim’s browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

What are the Cloud Security Control Layers?

1. Applications - SDLC, WAF

2. Information - DLP, Encryption

3. Management - GRC, IAM, Patch & Configuration

4. Network - NIDS/NIPS, DNSSEC, QoS

5. Trusted Computing - Roots of Trust (RoT) is a set of functions in the trusted computing module that is always trusted by the computer’s operating system (OS). The RoT serves as separate compute engine controlling the trusted computing platform cryptographic processor on the PC or mobile device it is embedded in.

6. Computer & Storage - Host-based firewalls, HIDS/HIPS, Encryption

7. Physical - Guards, Guns and Gates

What is Cloud Pen Testing? Actively evaluating the security of the cloud system. The scope of the test is governed by the service model used.