Attack Attack! Threat Modeling

Beware of Geeks bearing gifts!

Larry England and John Krautheim

04 May 2022

Table of Contents

- Who are we?
  - Intro & Goals
    - Approach a simple 4-step tango - Dancing anyone?
      - Examine Log4j vulnerability

BROADCOM

Who are the presenters? Who are these guys anyway?

John Krautheim, PhD, CISSP-ISSEP

Broadcom Software Engineer - Pittsburgh

20 years experience in Computer Security

Enjoys cycling and camping

Is an audiophile on the cheap

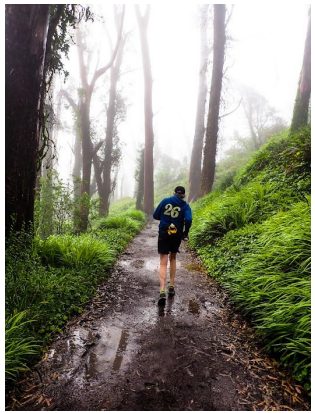
Larry England

Broadcom Software Engineer with experience across many technologies - sunny (rainy?) California

Enjoys hiking, trail running (ultras), crashing bikes, xcountry skiing, photography, music

Very amatuer piano player

Participant in the witness protection program



BROADCOM

Once upon a time, there was Log4j…

BROADCOM

Where were you when Log4j hit you? …

On or about Dec 10th, 2021, while at work, I **awoke to the question** by one of our "Security Champions" to hear

"Who is using Log4j()???"... and

The Slack Channel is going nuts

See CVE2021-44228 CVSS score 10!! Outline of steps to exploit see this.

Estimated number of attempted attacks was 10 million to exploit this zero-day vulnerability!

Could this have been prevented? How might this have been prevented?

we'll look at this question later ….

BROADCOM

Intro

BROADCOM

Goals

Assuming **every system is penetrable** (if turned on)

and you are either a general user or external user

ask yourself these questions:

How would you breach these systems?

Where are the vulnerabilities?

What are the risks for our users?

What is a Threat?

A potential event that has “unwelcome/unintended consequences”.

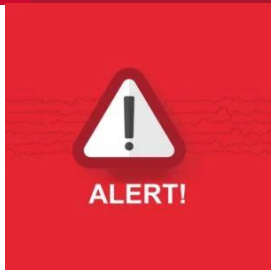
Or an individual or org from which an attack can originate.

> Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. Also, the potential for a threat-source to successfully exploit a particular information system vulnerability.
>
> - NIST SP 1800-15B

The big questions to ask:

- What are we working on?
- What could go wrong?
  - What is the probability an identified threat will happen?
  - What would the consequences/costs be if it does happens?
- What are we going to do about it?
- Did we do a good job?

What is Threat Modeling?



Set of approaches and techniques to identify risks affecting a system based on

- how it is architected
- how is it coded
- how it is supposed to work

Through **software design analysis**, threat modeling identifies weaknesses by juxtaposing various design views against threat agents. This includes

- Security controls and boundaries
  - ex: not mixing authorized and unauthorized code in an address space
- Weak or ineffective encryption
  - ex: use of DES-56 encryption - or using your own encryption (even modified standard encryption)
- Potential vulnerabilities
- Risk Scenarios - What could go wrong go wrong go wrong go wrong go wrong ?

Think like a hacker. Break the rules. If not us, someone else will.

- **Goal**: get the system(s) to behave in an unintended manner … somehow!
- Assume you can obtain a general user credentials (even on z)
  - How can you impersonate another (elevated) id?
  - Are credentials passed in the clear?
  - Can the various credentials be compromised/spoofed?
  - What are the credentials for datastores? Can datastores be compromised? What could I find in datastore?
- What are the opportunities for input by users?
  - How do you ensure your caller is whom you expect/allow?
  - What does the protocol allow? How is it protected from abuse/unauthorized actions?
- Can a man-in-the-middle attack be conducted due to the number of hops?
  - Does any information get 'leaked' that could be used in combination with other info to form an attack (like leaking userids, (potentially sensitive) applications, etc.)?
- Policy enforcement - where defined? how to circumvent? how enforced?
  - How is an aberrant/malicious application prevented from being scheduled on an agent(s)?
  - Is it possible to install a Trojan horse into a system?
- Are there vulnerabilities open on z/OS due to this structure?

BROADCOM

4 easy steps to Security Sobriety (threat modeling)

1. Decompose the application / product
2. Find the threats - think like a hacker
3. Rank the threats
4. Determine steps to take - reduce the threats

Step 1 - Decompose the Application / Product

BROADCOM

Steps - How to approach this task?

Model the application to understand how a system works - Asking “what could go wrong?” **looking at “4+1 architecture”**

1) Identify trust boundaries/zones
2) Identify actors (internal and external)
3) Add data flows
4) Identify potential entry points
5) Identify risks / assess impacts
6) Authentication flows
7) Dependencies
8) Entry & exit points
9) Assets (ex: databases, files)

Things to think about - can you create a doomsday scenario - what’s the worst that could happen?

4+1 Architectural model - provides different perspectives of the arch

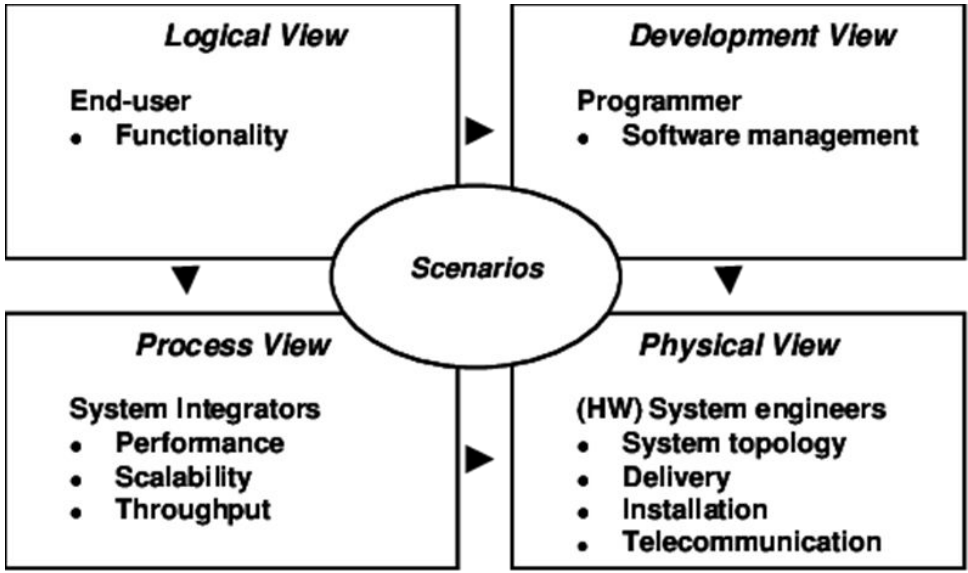
Logical View - functionality of the systems for the end-users

Process View - The system processes and how they communicate / trust boundaries

Development View - implementation view

Physical View - Deployment view / topology

Scenarios/use cases



BROADCOM

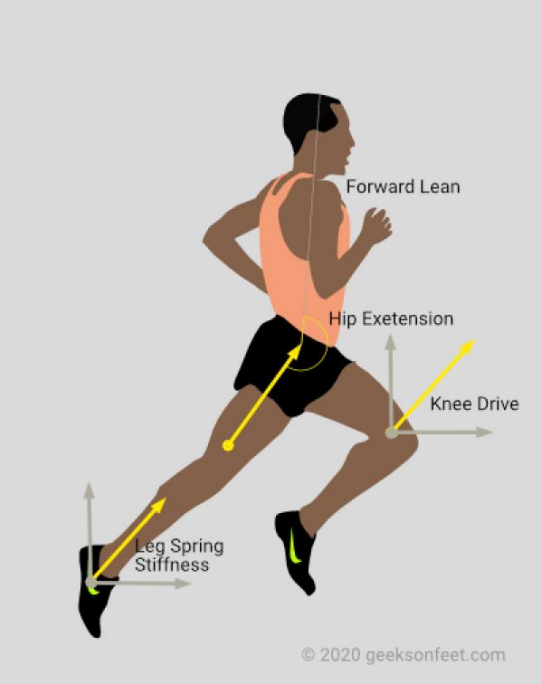
Step 2 - Find the Threats

BROADCOM

Security properties to consider

property	description
Confidentiality	Data is only available to the people intended to access it.
Integrity	Data and system resources are only changed in appropriate ways by appropriate people.
Availability	Systems are ready when needed and perform acceptably.
Authentication	The identity of users is established (or you're willing to accept anonymous users).
Authorization	Users are explicitly allowed or denied access to resources.
Non-repudiation	Users can't perform an action and later deny performing it

Use a well-known, structured approach - STRIDE



Spoofing

Tampering

Repudiation

Information disclosure

Denial of Service

Elevation of privilege

BROADCOM

STRIDE = **S**poofing, **T**ampering, **R**epudiation, **I**nfo disclosure, **D**oS, **E**levation of Privilege

A structure to consider when performing a threat assessment

Spoofing	accessing and use of another user's credentials	authentication
Tampering	maliciously change or modify persistent data, such as a database	integrity
Repudiation	performing prohibited operations in a system that lacks the ability to trace the operations	non-repudiation
Info Disclosure	intending to read a file that one was not granted access to, or to read data in transit.	confidentiality
Denial of Service	attempting to deny access to valid users, such as by making a web server temporarily unavailable or unusable	availability
Elevation of Privilege	intending to gain privileged access to resources in order to gain unauthorized access to information or to compromise a system	authorization

Why do we need all of the ‘4 models/flows’?

Element	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Data Flows		X		X	X	
Data Stores		X		X	X	
Processes	X	X	X	X	X	X
Interactors	X		X			
Trust Zones						X

BROADCOM

Step 3 - Rank the Threats

BROADCOM

DREAD - how to assess impact of a vulnerability

- **D**amage: How big would the damage be if the attack succeeded?
- **R**eproducibility: How easy is it to reproduce an attack?
- **E**xploitability: How much time, effort, and expertise is needed to exploit the threat?
- **A**ffected Users: If a threat were exploited, what percentage of users would be affected?
- **D**iscoverability: How easy is it for an attacker to discover this threat?

BROADCOM

MITRE ATT&CK®

What is MITRE ATT&CK® (Adversarial Tactics, Techniques and Common Knowledge) ?

It's a knowledge base of **tactics** and **techniques** designed for threat hunters, defenders and red teams to help classify attacks, identify attack attribution and objectives, and assess an organization's risk.

- Constantly evolving based upon new attack vectors discovered
- Very detailed (eye chart if shown in a slide!) find it here -> https://attack.mitre.org/
- It is presented in a tabular form
  - columns that represent the tactics (or desired outcomes) used during the life of an attack
  - rows that represent of techniques that are utilized to achieve their tactical goals.
  - over 400 attack patterns identified

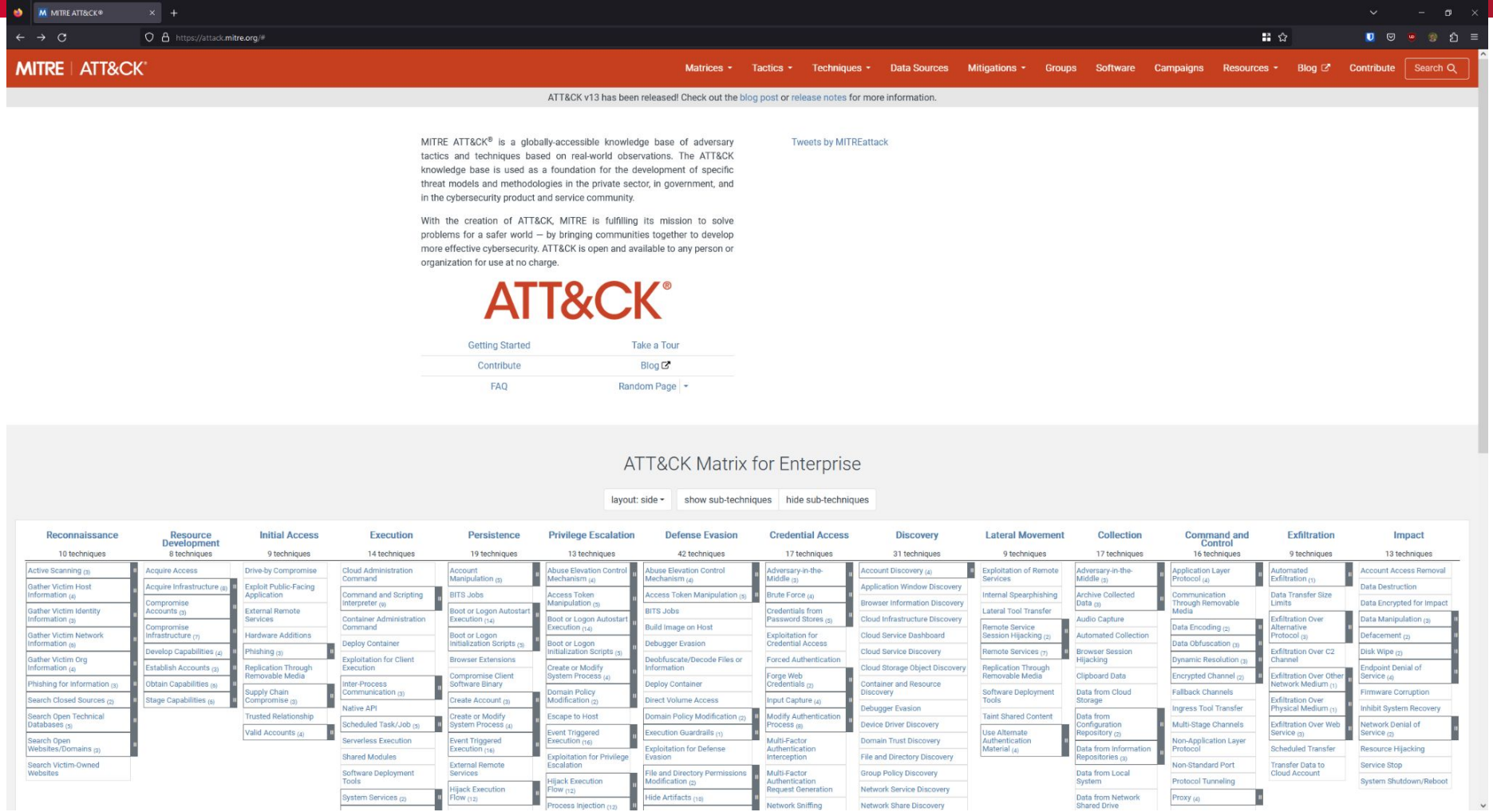
Why even look at MITRE ATT&CK?

- Adopt an attacker's perspective
- The benefit of the ATT&CK framework is that organizations can gain an understanding of how adversaries operate, the steps they might plan to take to gain initial access, discover, move laterally, and exfiltrate data.

how to use the MITRE ATT&CK framework?

- Good career :-)
- See this doc - from the US Center for Cybersecurity and Infrastructure Security Agency (CISA)

BROADCOM



 |

BROADCOM

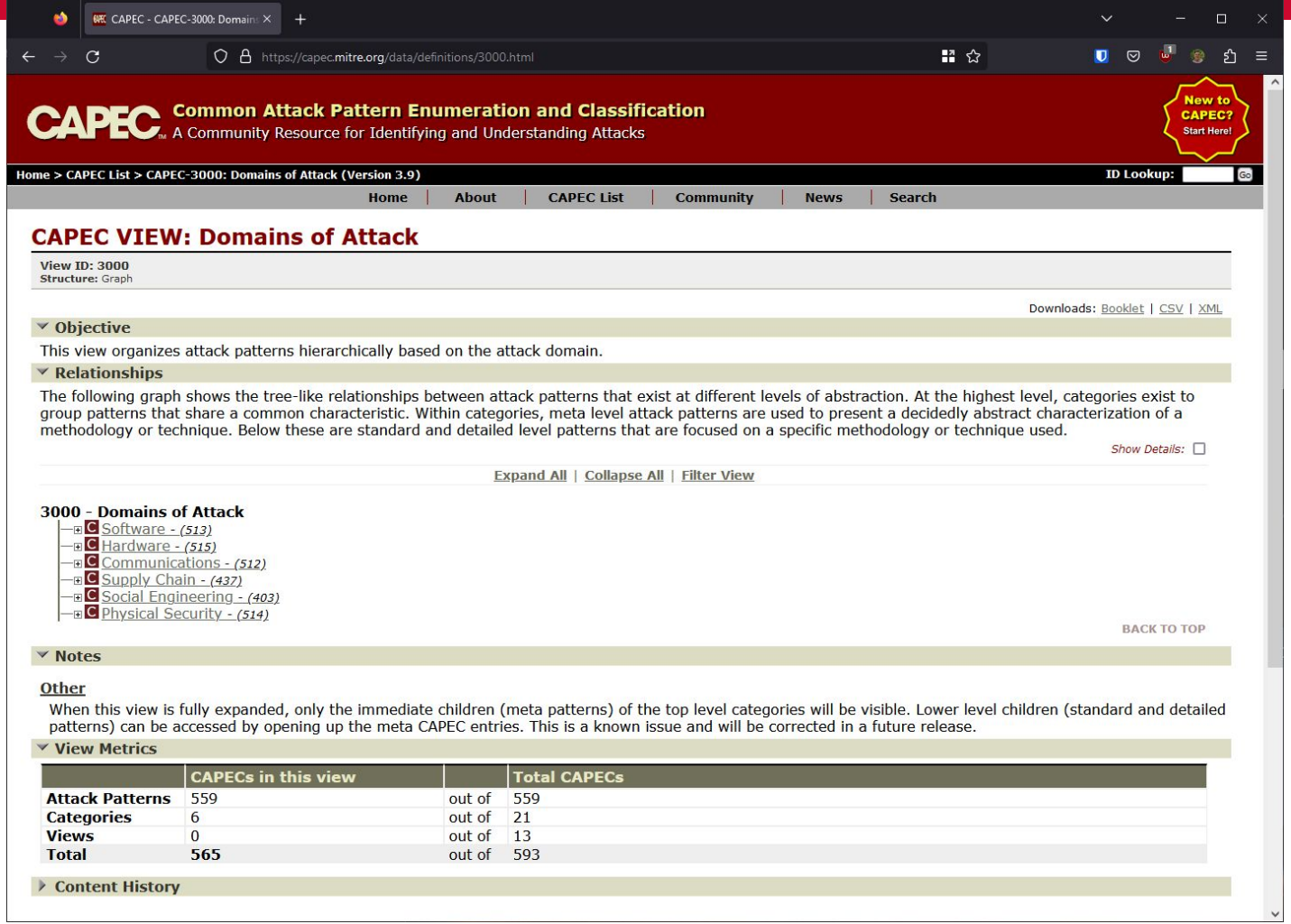
MITRE CAPEC™

What is MITRE CAPEC™ (Common Attack Pattern Enumerations and Classifications) ?

A comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. It can be used by analysts, developers, testers, and educators to advance community understanding and enhance defenses.

- CAPEC is focused on application security and describes the common attributes and techniques employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. (e.g., SQL Injection, XSS, Session Fixation, Clickjacking)
  - Focus on application security
  - Enumerates exploits against vulnerable systems
  - Includes social engineering / supply chain
  - Associated with Common Weakness Enumeration (CWE)
- Domains of Attack
  - Software
  - Hardware
  - Communications
  - Supply Chain
  - Social Engineering
  - Physical Security
- Details are here -> https://capec.mitre.org/

BROADCOM



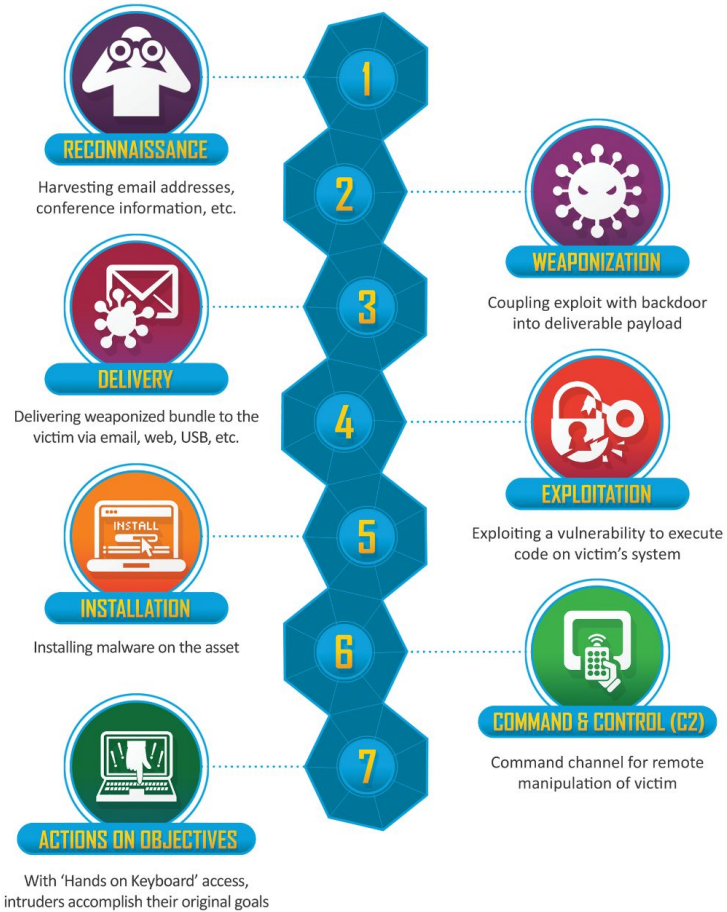
BROADCOM

Using ATT&CK and CAPEC

- Use ATT&CK for:
  - Comparing computer network defense capabilities
  - Defending against the Advanced Persistent Threat
  - Hunting for new threats
  - Enhancing threat intelligence
  - Adversary emulation exercises

- Use CAPEC for:
  - Application threat modeling
  - Developer training and education
  - Penetration testing

Kill Chain



https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/photo/cyber/THE-CYBER-KILL-CHAIN-body.png.pc-adaptive.full.medium.png

BROADCOM

Step 4 - Actions to Take

BROADCOM

Actions to take

- **Open a work item to track the work (this may be in a separate bug repo)**

- **Report the findings so we can learn from the experience/exercise - ie provide feedback to the engineering team to learn an elevate their expertise**

Revisit the infamous Log4j

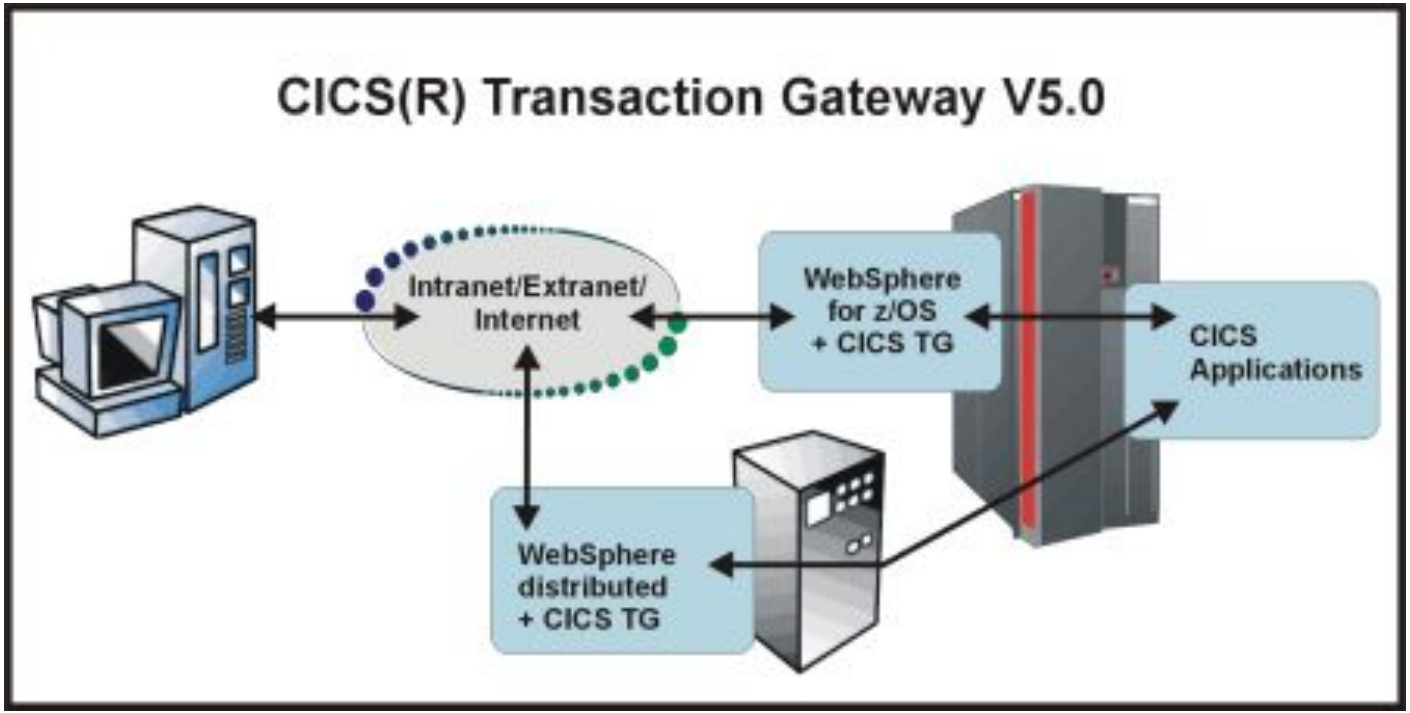
BROADCOM

Can we use Log4j as an example??

What if threat model was performed against Log4j … would the huge hole been discovered?

See Threat Modeling as a way of Thinking about Design Flaws

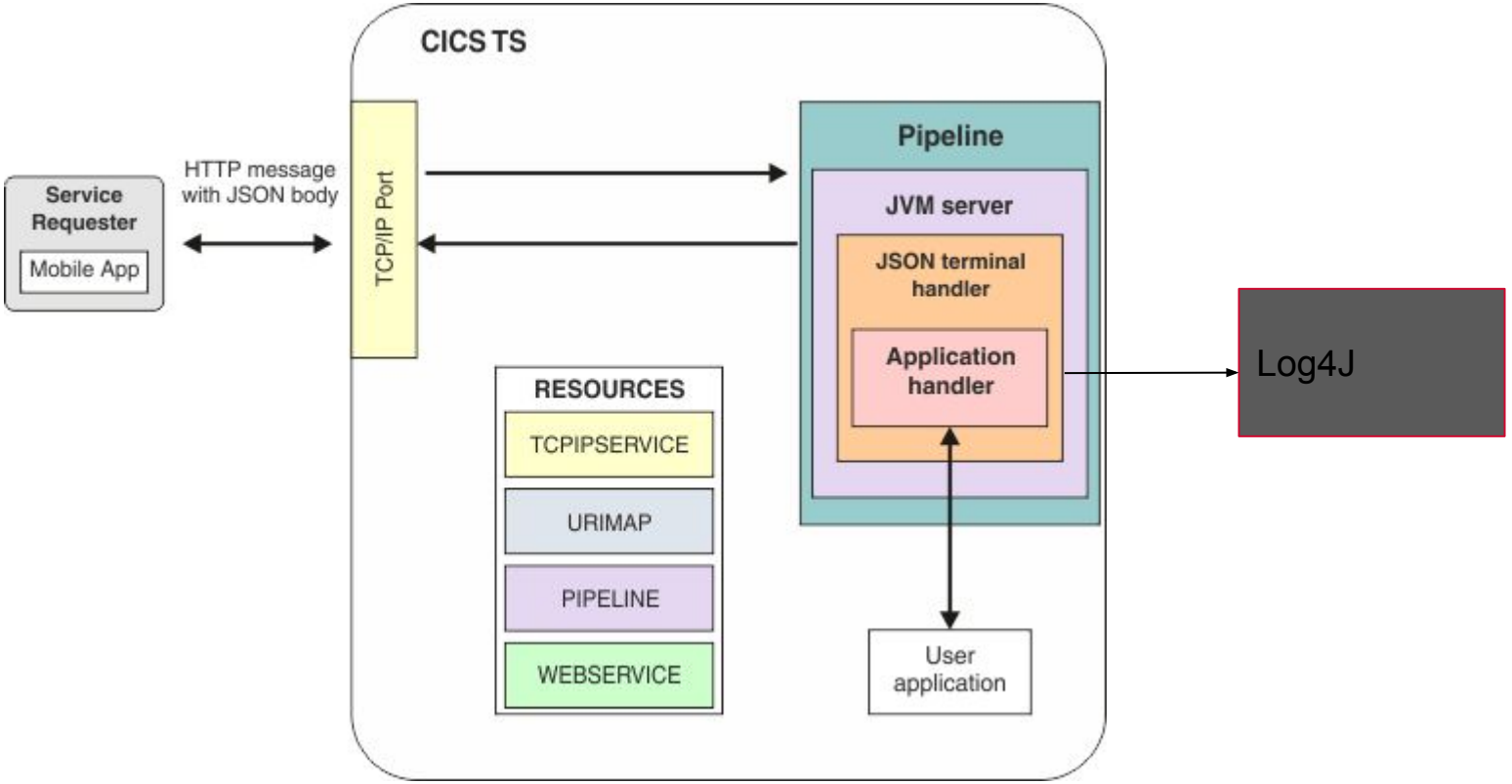
Is Log4j in your system?



https://www.ibm.com/common/ssi/GIF/ALET/AIM00010.GIF

BROADCOM

CICS using Java Pipeline



https://www.ibm.com/docs/en/cics-ts/5.3?topic=caspjr-cics-as-service-provider-json-requests-using-cics-java-pipelines

BROADCOM

CICS Configuration File

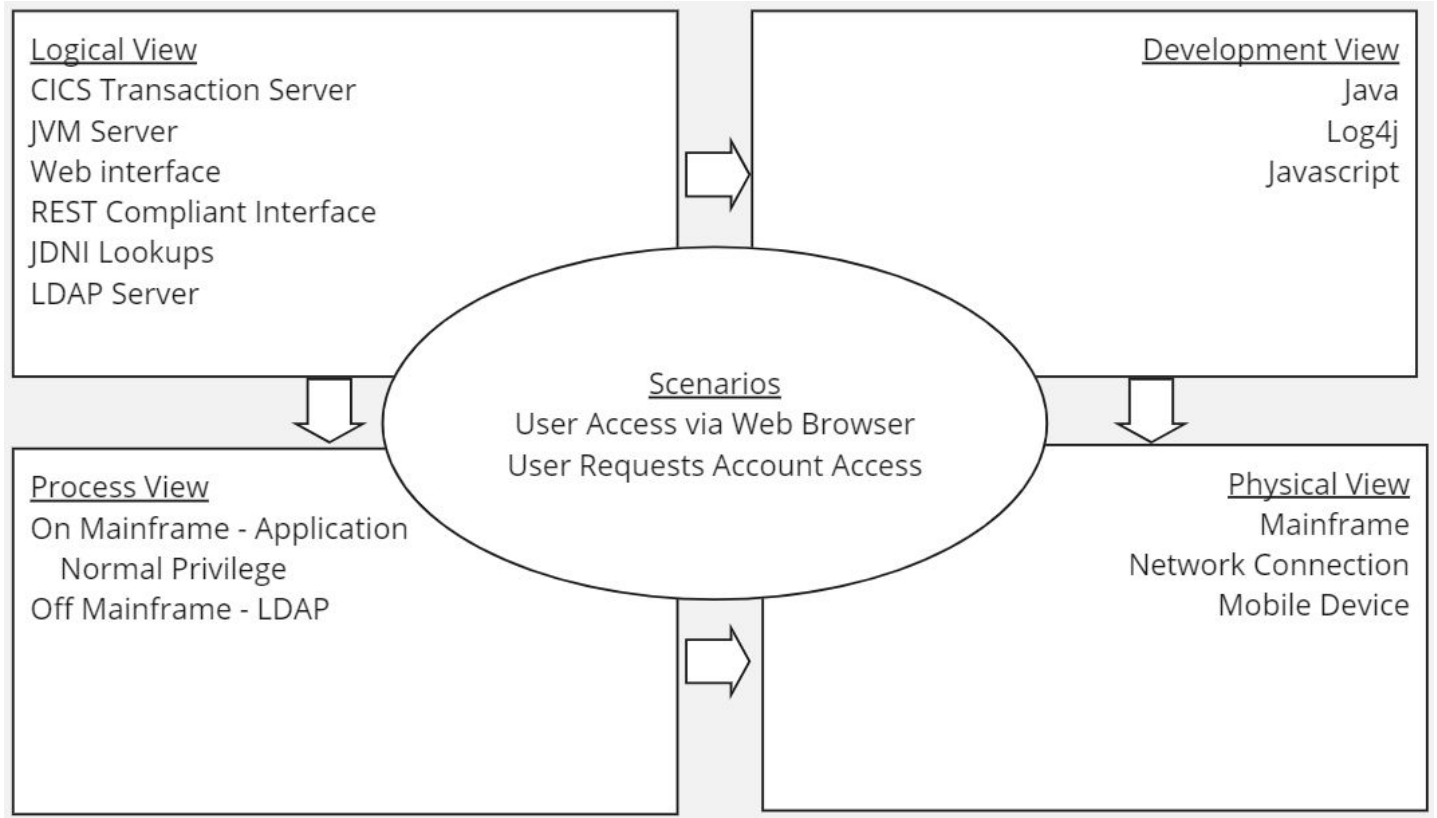
Project setup:

- Log4jBundle project (OSGI bundle)
  - contains MANIFEST.MF
  - contains log4j.jar
  - contains log4j.properties

- HelloWorldBundle project (OSGI bundle)
  - contains MANIFEST.MF

- HelloWorldCICSBundle project (CICS bundle)
  - contains cics.xml
  - contains Log4jBundle
  - contains HelloWorldBundle

4-in-1 Model for Java application with Log4j

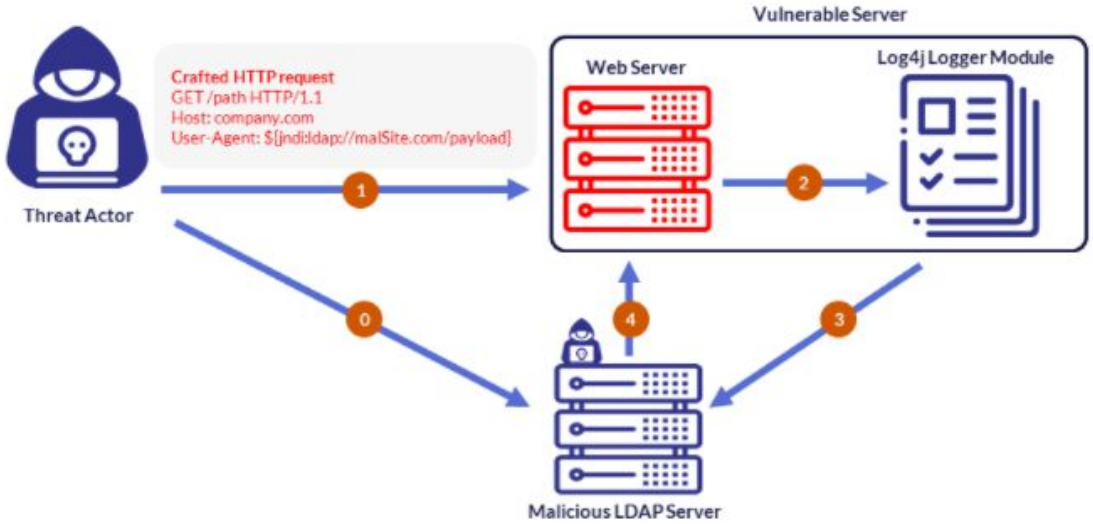


BROADCOM

What could possibly go wrong?

- What if the log messages include JNDI lookups?
- What if we use JNDI to connect to an external LDAP server?
- What if the LDAP server is controlled by a third actor?
- What if the LDAP server responds with directory information pointing to another external service under an attacker's control?
- What if JNDI can be used to cause remote code execution via deserialization?

This could happen



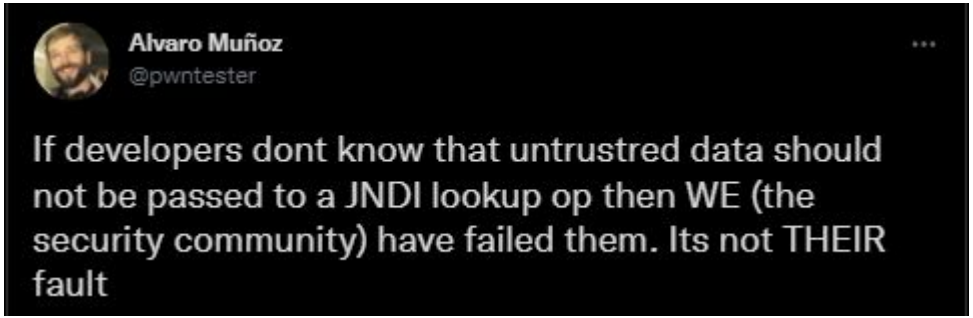
0. Threat actor setting up his malicious LDAP server with malicious Java class
1. Threat actor sends malicious payload that is likely to be logged by the application
2. Payload passed to Log4j for logging
3. Log4j parse the payload and make a query to the malicious LDAP server
4. The LDAP server responds with content that holds the malicious java class

https://cyberint.com/blog/research/log4j-incident-update/

BROADCOM

Summary

- Think like a hacker! - Nothing takes the place of thinking!
- Adopt Threat Modeling
- ToDo: Implement Threat Modeling into your culture - think like a hacker!



BROADCOM

Thank You

Bibliography

- Threat Modeling Manifesto
- https://www.iriusrisk.com/resources-blog/threat-modeling-as-a-way-of-thinking-about-design-flaws-log4j-case
- Threats - What Every Engineer Should Learn from Star Wars
- https://attack.MITRE.org/
- https://capec.MITRE.org/
- https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats
- https://satoricyber.com/glossary/threat-modeling-with-microsoft-dread/
- https://shapingsoftware.com/4-1-view-model-of-software-architecture/
- https://owasp.org/www-community/Threat_Modeling_Process

BROADCOM