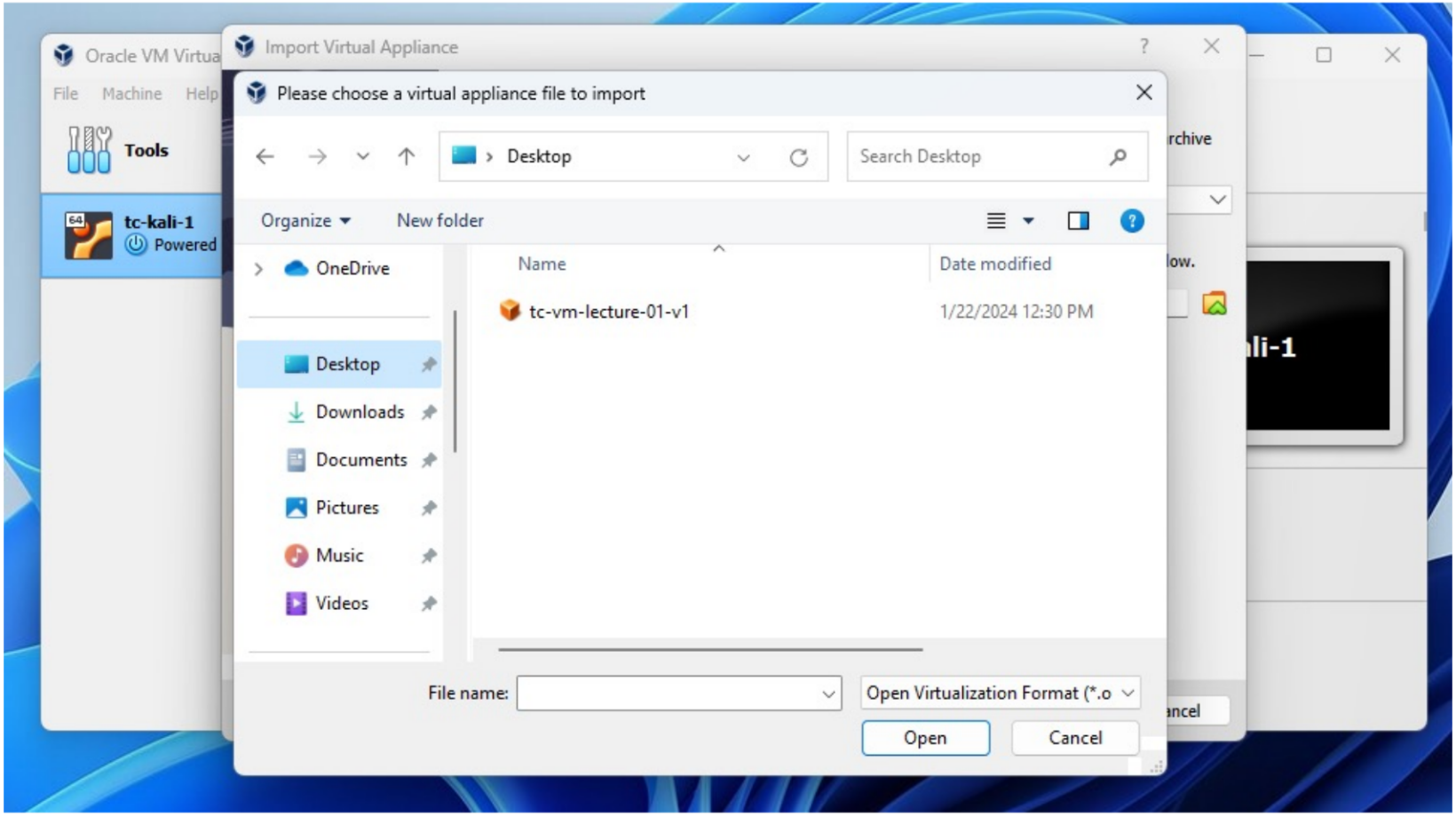
Threats and Countermeasures

Lecture 04:
Resource Development & Initial Access

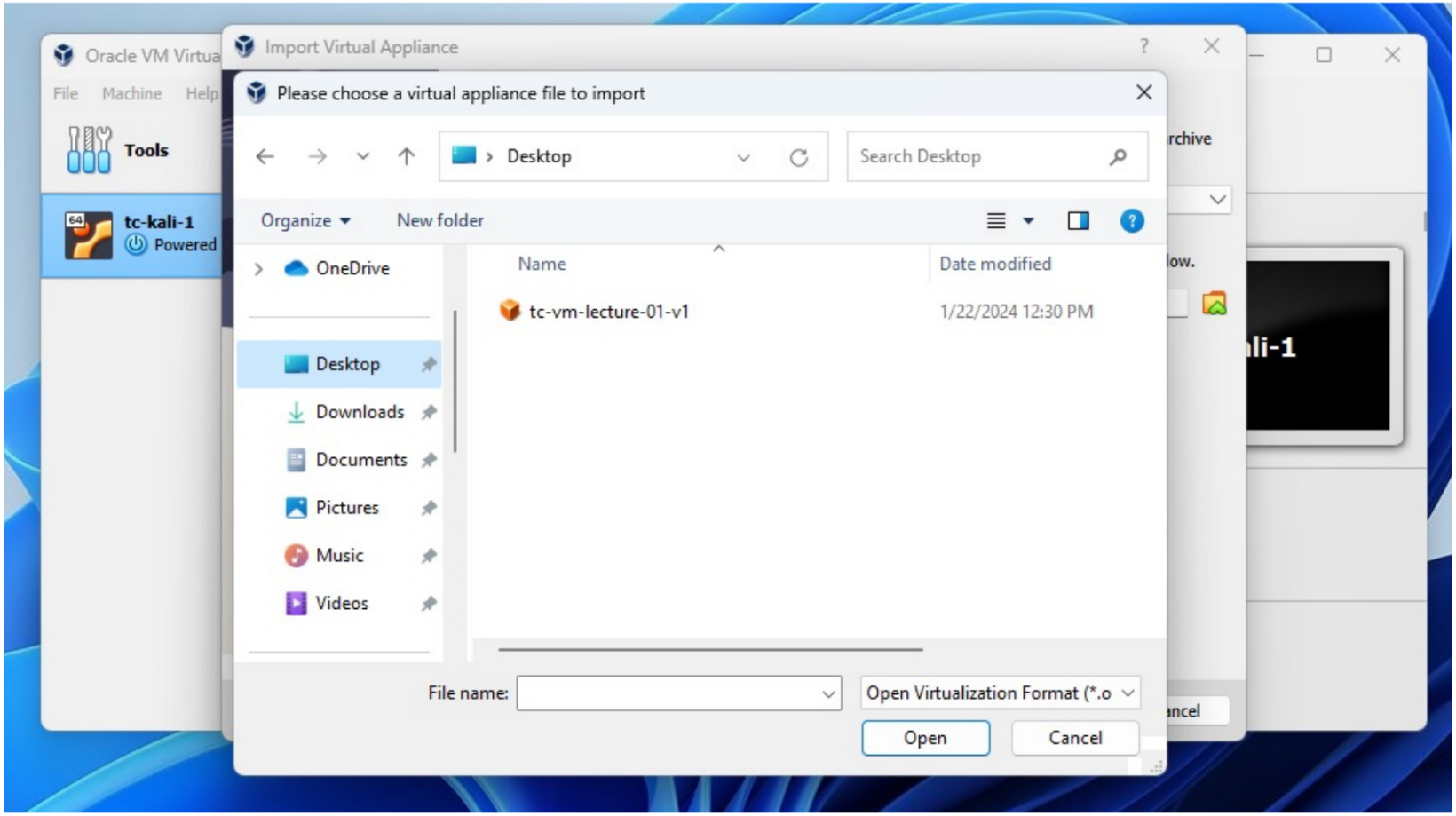
COMP-5830/-6830
Spring 2025

Today: tc-vm-03_r06



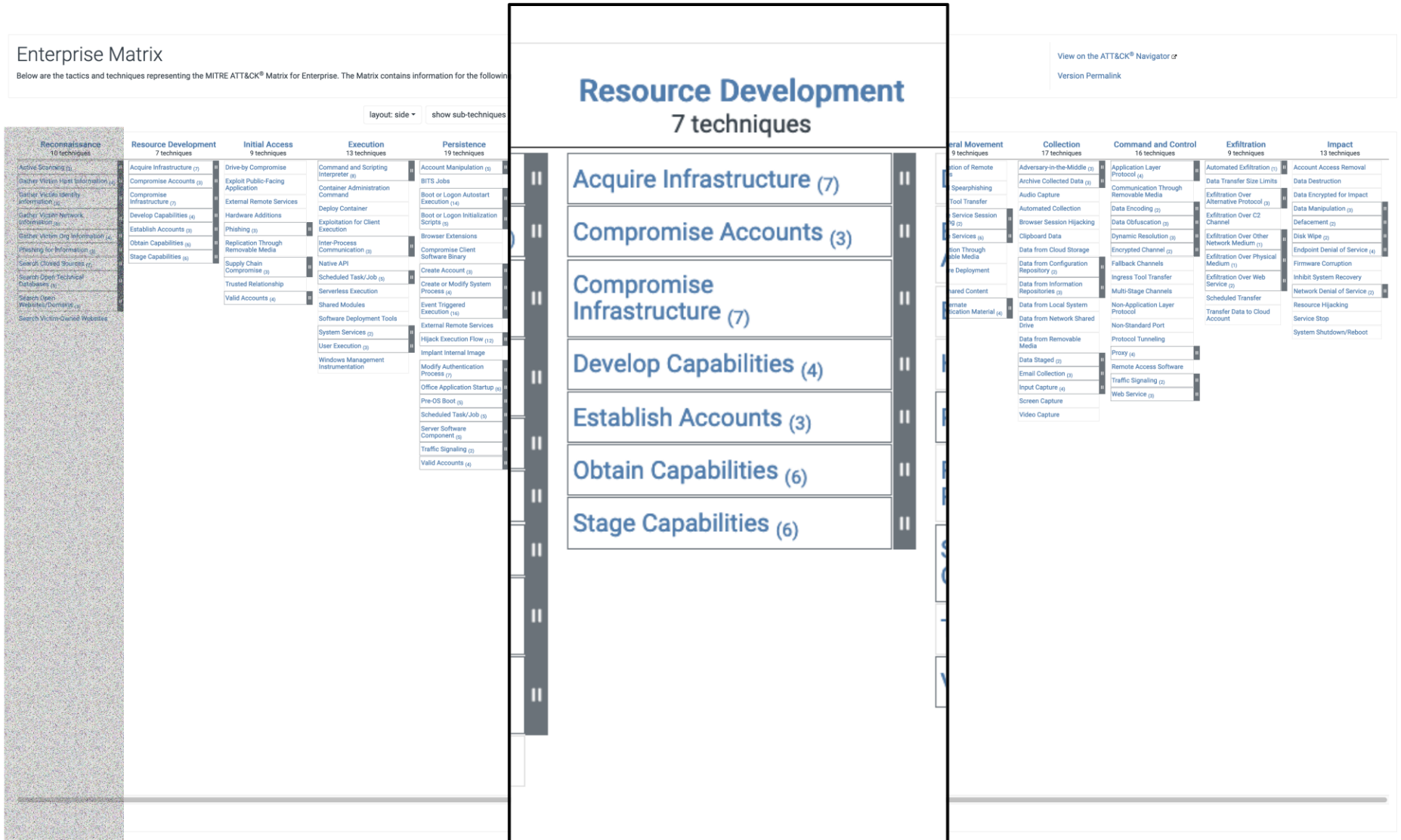
Disk Crypto: ZVcNabdysWUnmo32MRkDZ4TC

Today: tc-vm-01_r11

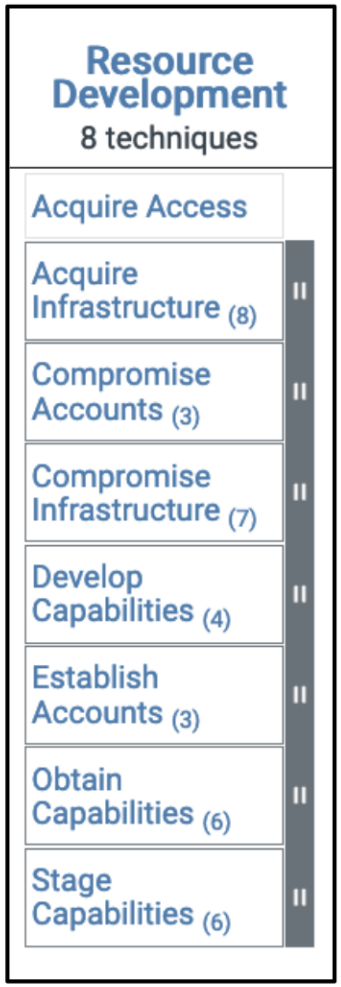


Disk Crypto: cY9hFa2hXuSgePTQNjhbsfob

MITRE ATT&CK

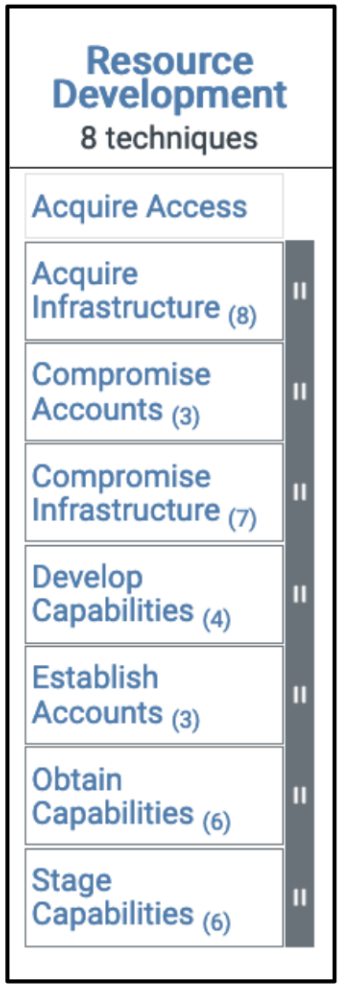

Enterprise Matrix
Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for Enterprise. The Matrix contains information for the followin
View on the ATT&CK® Navigator
Version Permalink
layout: side
show sub-techniques
Reconnaissance
10 techniques
Resource Development
7 techniques
Initial Access
9 techniques
Execution
13 techniques
Persistence
19 techniques
Collection
17 techniques
Command and Control
16 techniques
Exfiltration
9 techniques
Impact
13 techniques
Resource Development
7 techniques
Acquire Infrastructure (7)
Compromise Accounts (3)
Compromise Infrastructure (7)
Develop Capabilities (4)
Establish Accounts (3)
Obtain Capabilities (6)
Stage Capabilities (6)

Resource Development



Resource Development consists of techniques that involve adversaries creating, purchasing, or compromising/stealing resources that can be used to support targeting. Such resources include infrastructure, accounts, or capabilities. These resources can be leveraged by the adversary to aid in other phases of the adversary lifecycle, such as using purchased domains to support Command and Control, email accounts for phishing as a part of Initial Access, or stealing code signing certificates to help with Defense Evasion.

Resource Development (simplified)



- Obtaining (directly or indirectly) resources to leverage in the attack
- In support of and enabling other phases of the attack

Actions

Actions

- Develop/Adapt exploits, malware, and toolchains

Actions

- Develop/Adapt exploits, malware, and toolchains
- Stage exploits, malware, and/or toolchains for access

Actions

- Develop/Adapt exploits, malware, and toolchains
- Purchase domains and/or cloud resources
- Stage exploits, malware, and/or toolchains for access

Actions

- Develop/Adapt exploits, malware, and toolchains
- Purchase domains and/or cloud resources
- Stage exploits, malware, and/or toolchains for access
- Poison/Optimize SEO for improved reachability

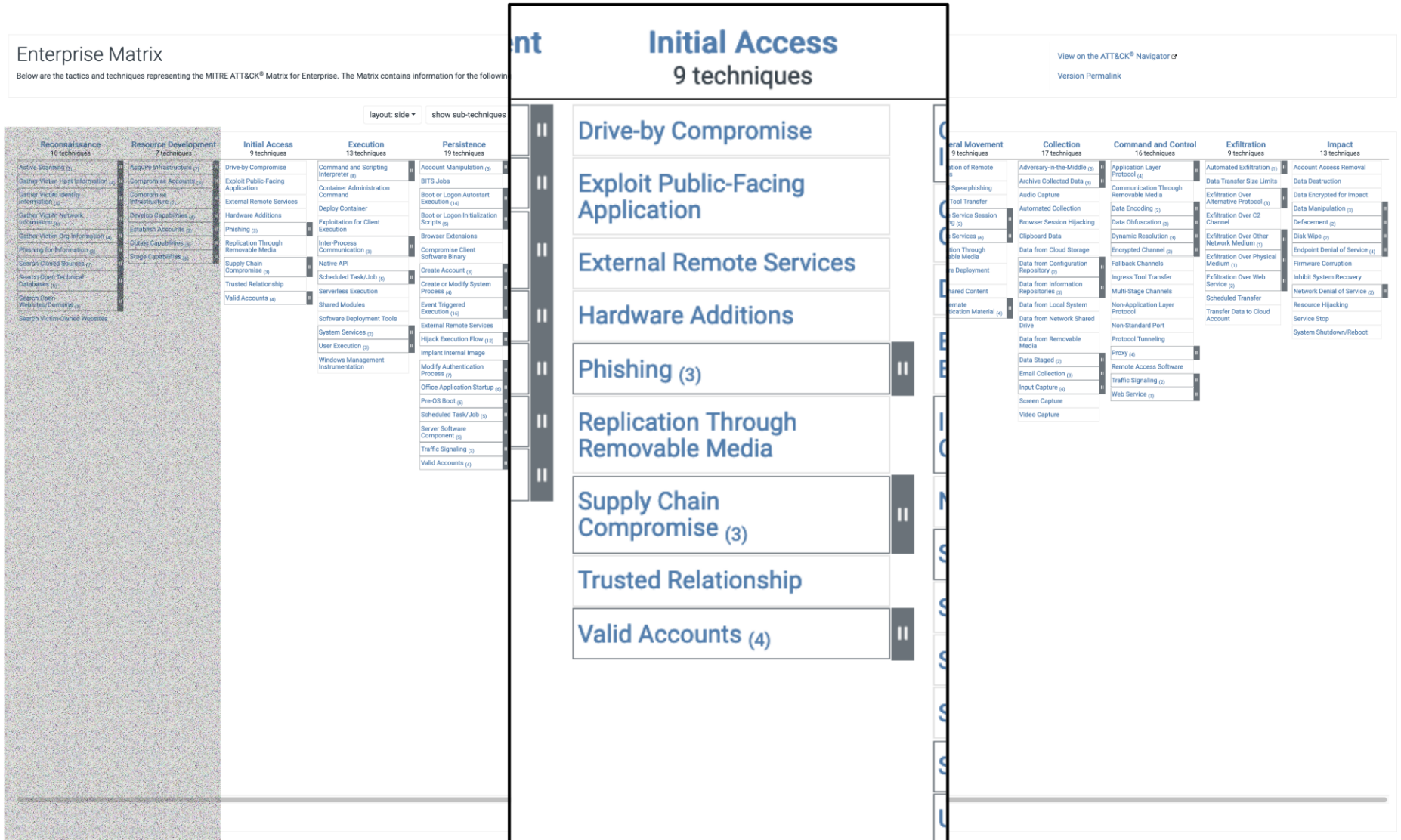
Actions

- Develop/Adapt exploits, malware, and toolchains
- Purchase domains and/or cloud resources
- Add credibility to accounts and infrastructure
  - Often called “aging”
- Stage exploits, malware, and/or toolchains for access
- Poison/Optimize SEO for improved reachability

Actions

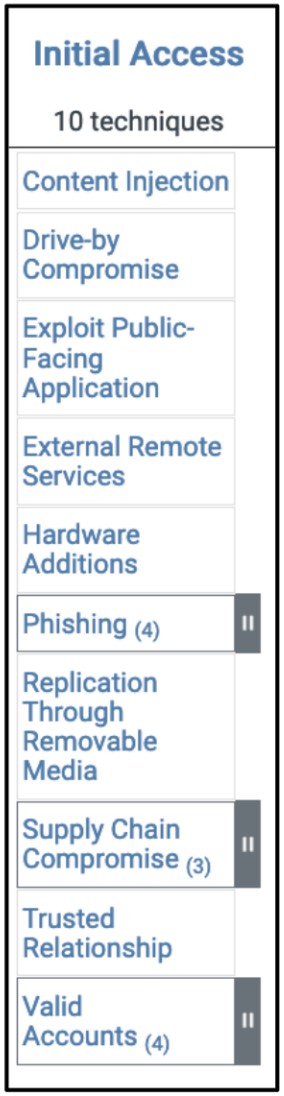
- Develop/Adapt exploits, malware, and toolchains
- Purchase domains and/or cloud resources
- Add credibility to accounts and infrastructure
  - Often called “aging”
- Stage exploits, malware, and/or toolchains for access
- Poison/Optimize SEO for improved reachability
- Design exfiltration mechanism for target network
  - (talk about later)

MITRE ATT&CK


Enterprise Matrix
Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for Enterprise. The Matrix contains information for the followin
View on the ATT&CK® Navigator
Version Permalink
layout: side
show sub-techniques
Reconnaissance
10 techniques
Resource Development
7 techniques
Initial Access
9 techniques
Execution
13 techniques
Persistence
19 techniques
Drive-by Compromise
Exploit Public-Facing Application
External Remote Services
Hardware Additions
Phishing (3)
Replication Through Removable Media
Supply Chain Compromise (3)
Trusted Relationship
Valid Accounts (4)
Command and Scripting Interpreter (8)
Container Administration Command
Deploy Container
Exploitation for Client Execution
Inter-Process Communication (3)
Native API
Scheduled Task/Job (5)
Serverless Execution
Shared Modules
Software Deployment Tools
System Services (2)
User Execution (3)
Windows Management Instrumentation
Account Manipulation (5)
BITS Jobs
Boot or Logon Autostart Execution (14)
Boot or Logon Initialization Scripts (5)
Browser Extensions
Compromise Client Software Binary
Create Account (3)
Create or Modify System Process (4)
Event Triggered Execution (16)
External Remote Services
Hijack Execution Flow (12)
Implant Internal Image
Modify Authentication Process (7)
Office Application Startup (6)
Pre-OS Boot (5)
Scheduled Task/Job (5)
Server Software Component (5)
Traffic Signaling (2)
Valid Accounts (4)
Initial Access
9 techniques
Drive-by Compromise
Exploit Public-Facing Application
External Remote Services
Hardware Additions
Phishing (3)
Replication Through Removable Media
Supply Chain Compromise (3)
Trusted Relationship
Valid Accounts (4)
Collection
17 techniques
Command and Control
16 techniques
Exfiltration
9 techniques
Impact
13 techniques
Adversary-in-the-Middle (3)
Archive Collected Data (3)
Audio Capture
Automated Collection
Browser Session Hijacking
Clipboard Data
Data from Cloud Storage
Data from Configuration Repository (2)
Data from Information Repositories (3)
Data from Local System
Data from Network Shared Drive
Data from Removable Media
Data Staged (2)
Email Collection (3)
Input Capture (4)
Screen Capture
Video Capture
Application Layer Protocol (4)
Communication Through Removable Media
Data Encoding (2)
Data Obfuscation (3)
Dynamic Resolution (3)
Encrypted Channel (2)
Fallback Channels
Ingress Tool Transfer
Multi-Stage Channels
Non-Application Layer Protocol
Non-Standard Port
Protocol Tunneling
Proxy (4)
Remote Access Software
Traffic Signaling (2)
Web Service (3)
Automated Exfiltration (1)
Data Transfer Size Limits
Exfiltration Over Alternative Protocol (3)
Exfiltration Over C2 Channel
Exfiltration Over Other Network Medium (1)
Exfiltration Over Physical Medium (1)
Exfiltration Over Web Service (2)
Scheduled Transfer
Transfer Data to Cloud Account
Account Access Removal
Data Destruction
Data Encrypted for Impact
Data Manipulation (3)
Defacement (2)
Disk Wipe (2)
Endpoint Denial of Service (4)
Firmware Corruption
Inhibit System Recovery
Network Denial of Service (2)
Resource Hijacking
Service Stop
System Shutdown/Reboot

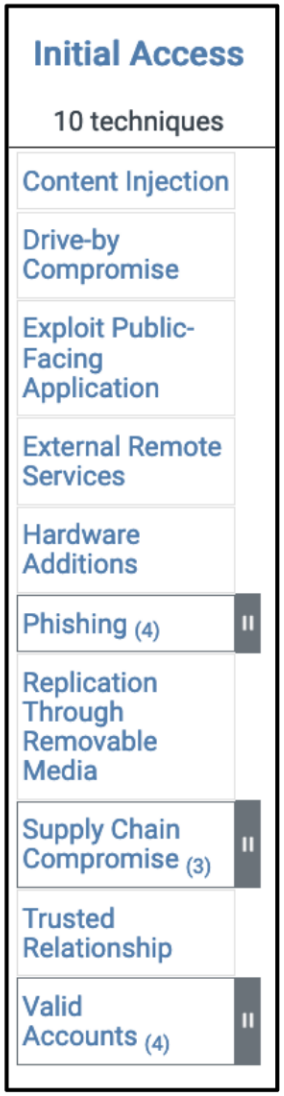
Initial Access

Initial Access consists of techniques that use various entry vectors to gain their initial foothold within a network. Techniques used to gain a foothold include targeted spearphishing and exploiting weaknesses on public-facing web servers. Footholds gained through initial access may allow for continued access, like valid accounts and use of external remote services, or may be limited-use due to changing passwords.



Initial Access (simplified)

- Goal: Initial foothold in a target network
- Doesn't have to be big or even impactful, only an improvement over no-access



Valid Credentials

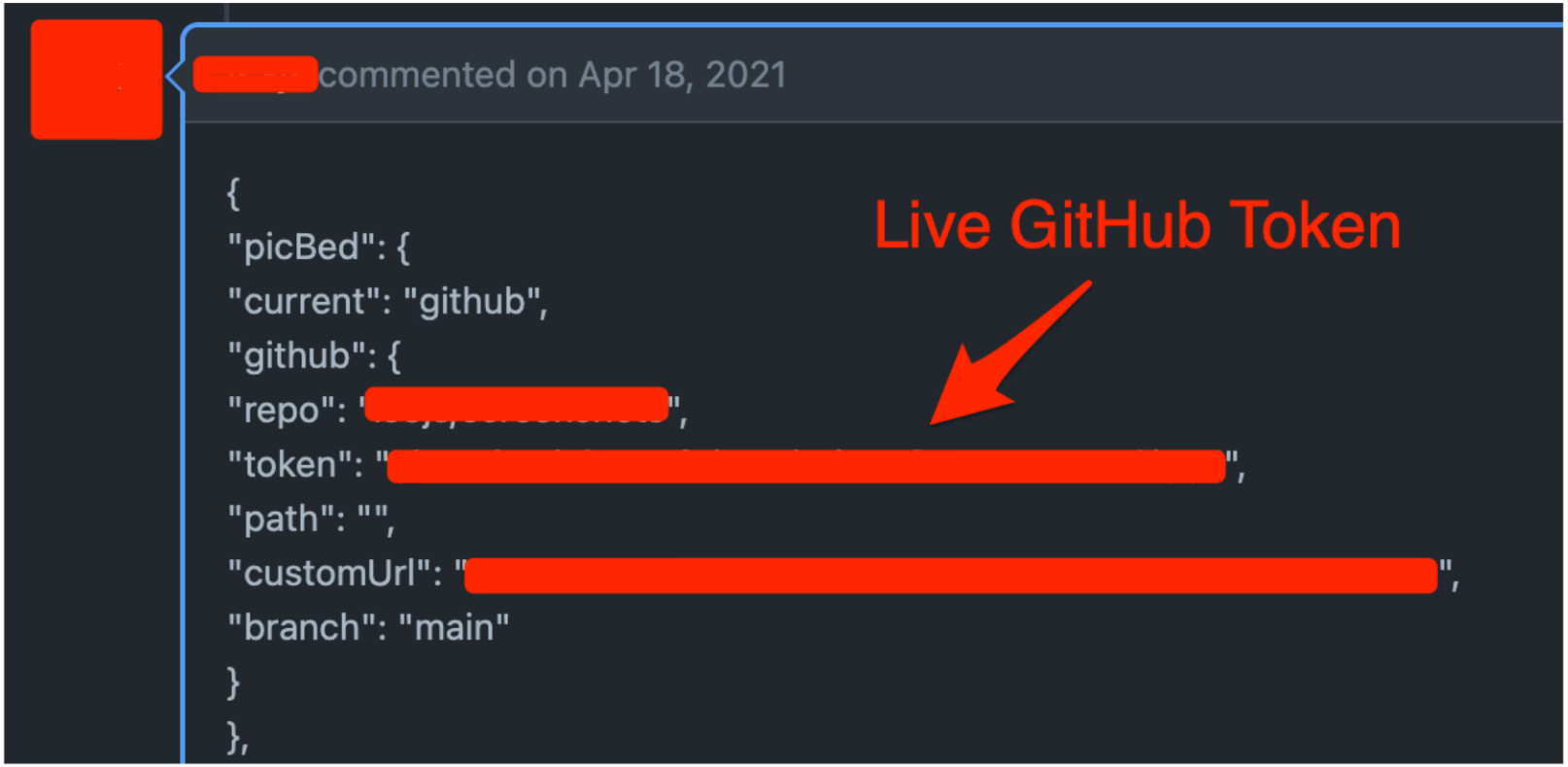
If the attacker can obtain/locate valid creds for the target, life begins on easy-mode.

B/c attacker is treated as legitimate user:

- Can bypass defenses entirely
- Can gain access into internal network
- Can come-and-go as they please

Valid Credentials

If the attacker can obtain/locate valid creds for the target, life begins on easy-mode.



Valid Credentials

If the attacker can obtain/locate valid creds for the target, life begins on easy-mode.

The Case of the RSA [redacted]-A-DUCK certificate

Nadia Heninger
Zakir Durumeric
Eric Wustrow
J. Alex Halderman

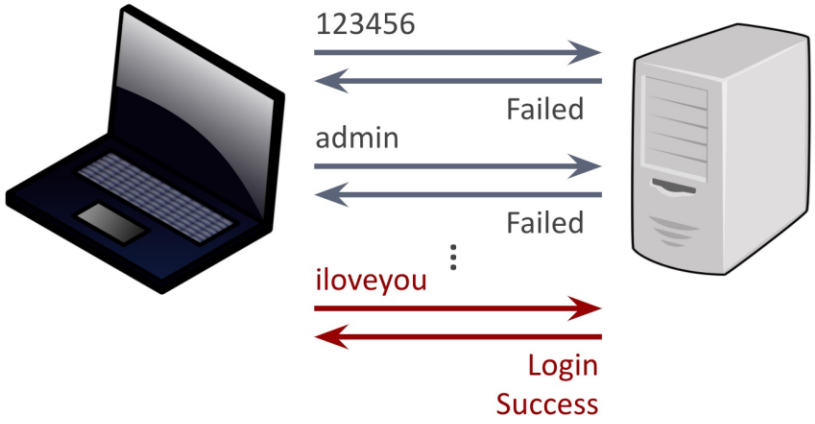
N=pq

Valid Credentials

If the attacker can obtain/locate valid creds for the target, life begins on easy-mode.

Online Password Guessing

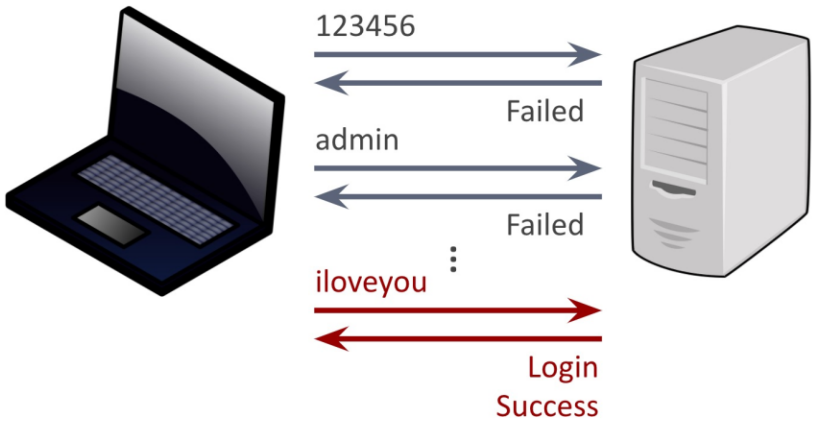
An **Online Password Guessing Attack** is where an actor treats the authentication server as an *oracle* to determine when it has found the correct password.



Online Password Guessing

An **Online Password Guessing Attack** is where an actor treats the authentication server as an *oracle* to determine when it has found the correct password.

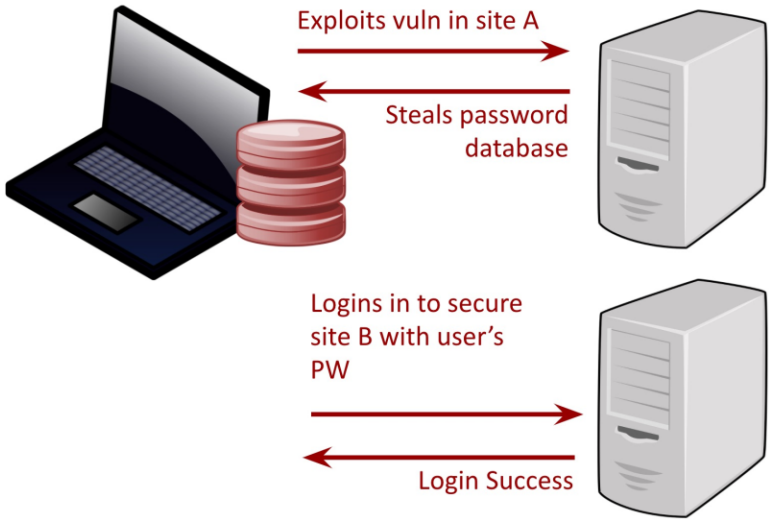
- Defenses:
  - Lock-out threshold
  - Rate limits
  - (re)CAPTCHAs
  - Anomaly detection



Credential Stuffing

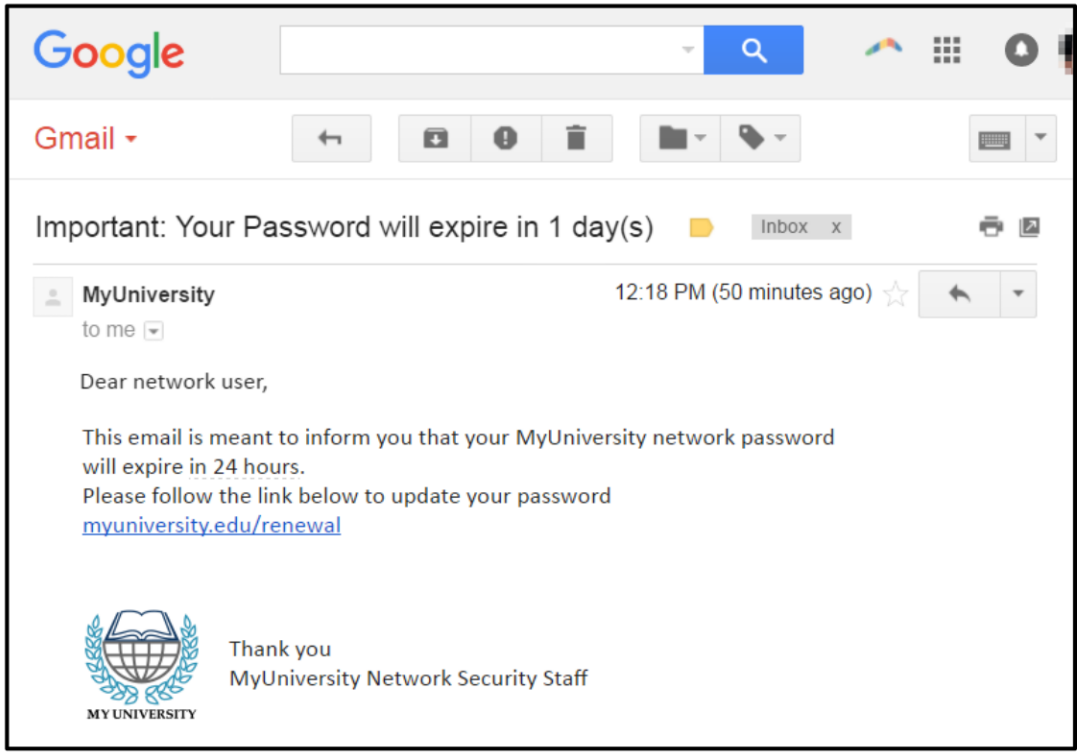
Credential stuffing attacks involve reusing known username-password combinations from one breach on a separate service.

- Relies on:
  - Ubiquity of accounts
  - Static email addresses
  - Password reuse
  - Laziness



Phishing for Access

Attempt to socially engineer benign users into giving direct or indirect access.



Phishing for Access

Attempt to socially engineer benign users into giving direct or indirect access.

Comes in many, many forms:

- Untargeted vs. spear-phishing
- Credentials vs. direct-access
- Attachments vs. downloads vs. browsing

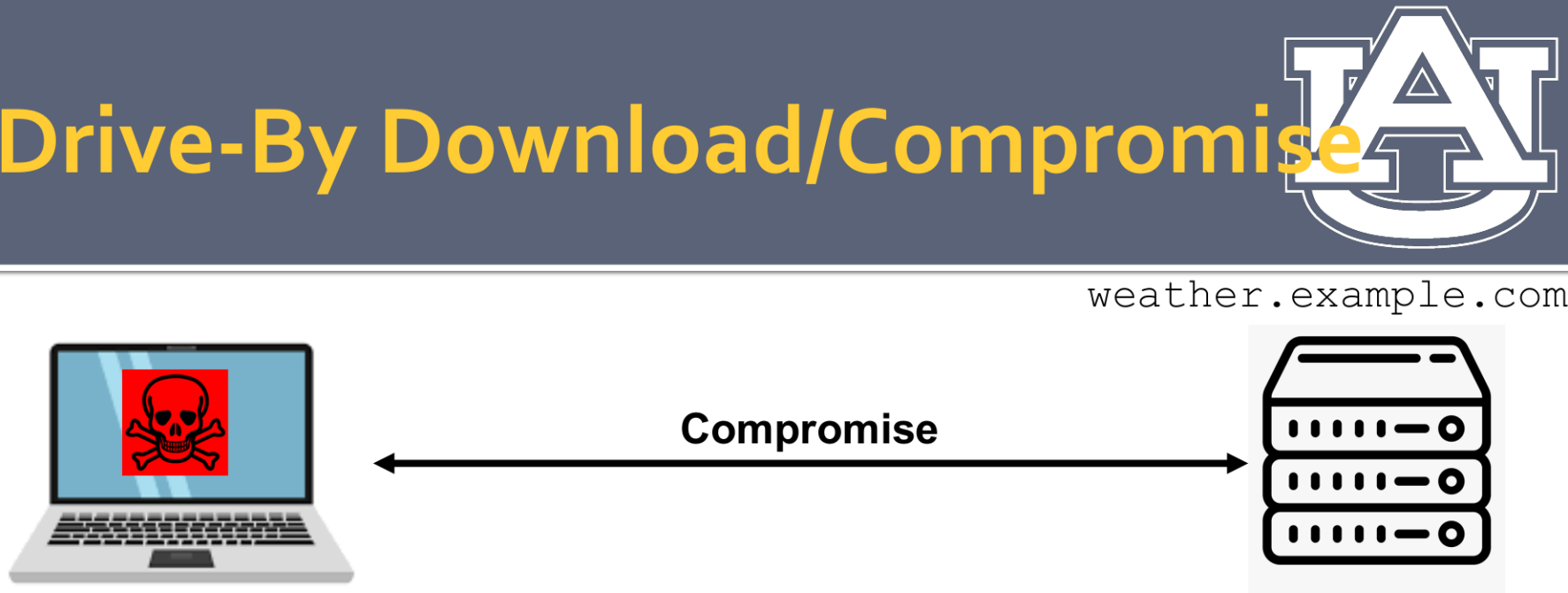
Drive-By Download/Compromise

- Legitimate user accesses a malicious web server and downloads/executes malicious functionality
  - JavaScript, iframe, etc

Drive-By Download/Compromise

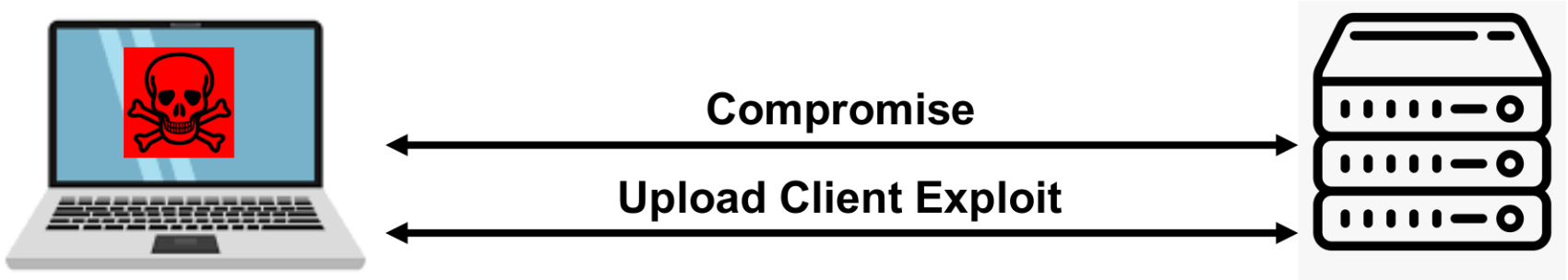
- Legitimate user accesses a malicious web server and downloads/executes malicious functionality
  - JavaScript, iframe, etc
- Vulns allow modification of webserver/app/database
  - Chat sessions, forum posts, comments, etc

Drive-By Download/Compromise

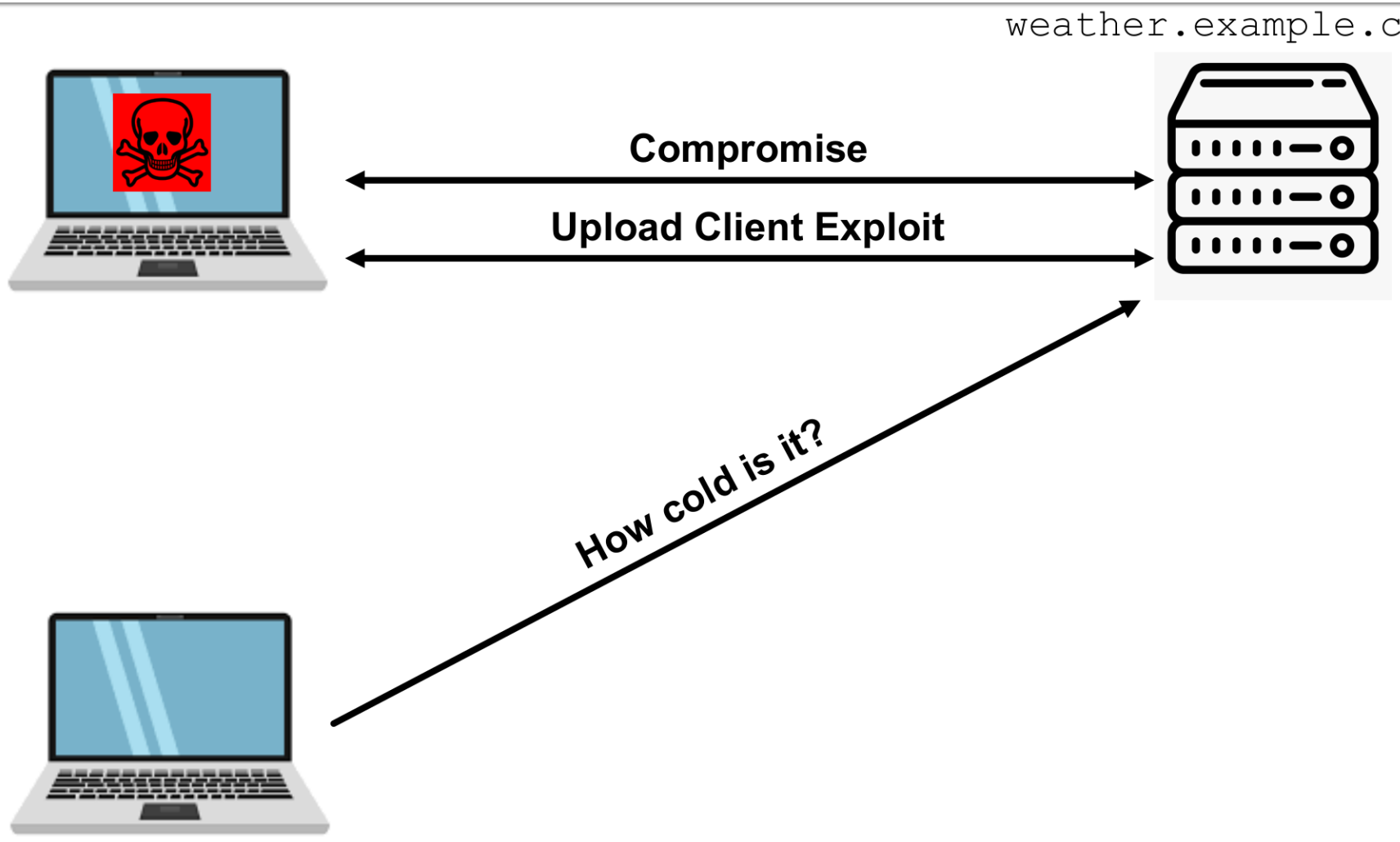


Drive-By Download/Compromise

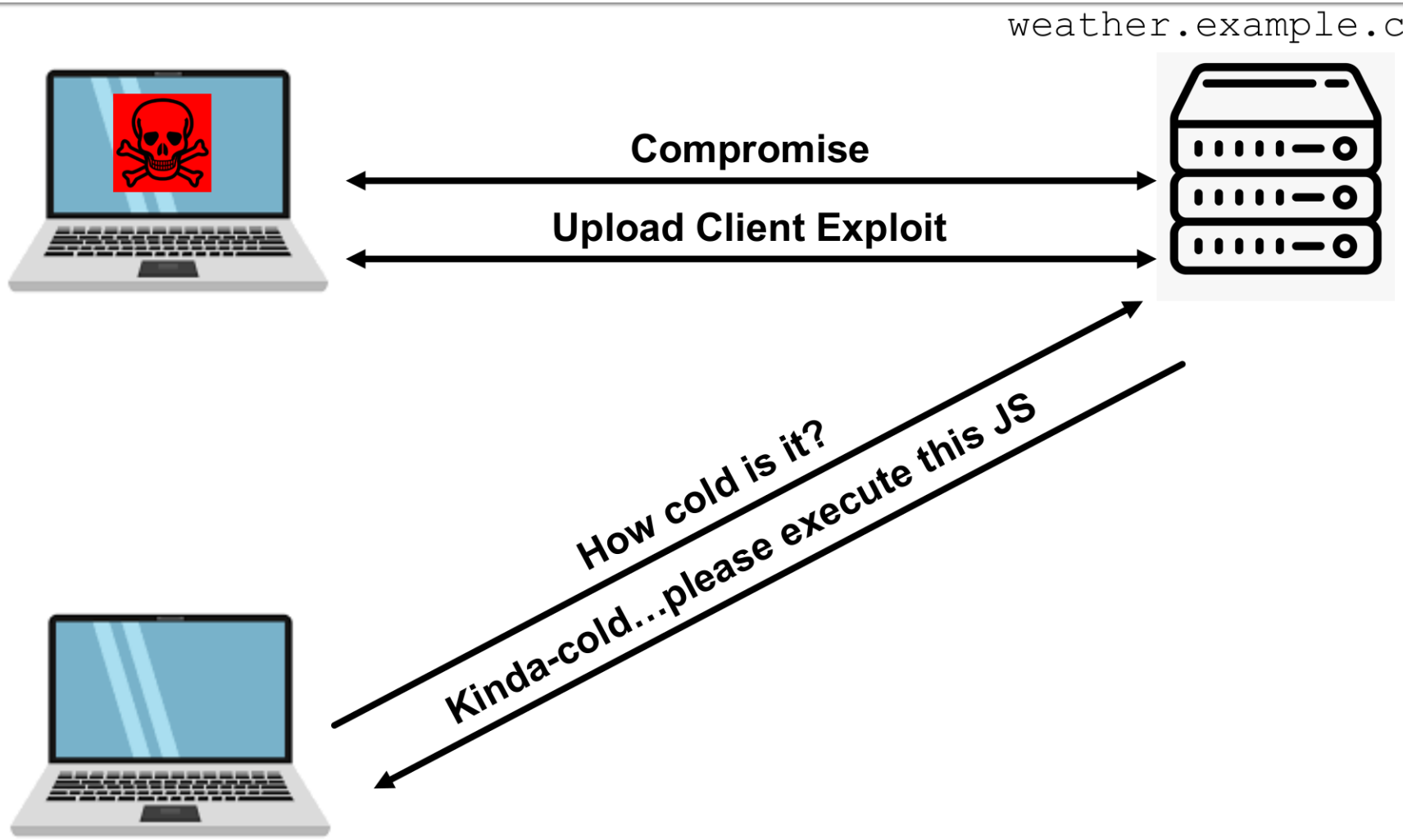
weather.example.com



Drive-By Download/Compromise



Drive-By Download/Compromise



Example: URL data

Template:

```
<div>
    Hello
    <script>
    (new URLSearchParams(
        window.location.search
    )).get('name');
    </script>
</div>
```

Example: URL data

Template:

```
<div>
    Hello {get name from url}
</div>
```

https://example.com?name=Alice

```
<div>
    Hello Alice
</div>
```

Example: URL data

```
<div>
    Hello {get name from url}
</div>
```

https://example.com?name=Alice%0DWelcome+to+the+Thunderdome%21

```
<div>
    Hello Alice
Welcome to the Thunderdome!
</div>
```

Example: URL data

```
<div>
    Hello {get name from url}
</div>
```

https://example.com?name=Alice%0D%3Cimg%20src%3D%22https%3A%2F%2Fexample.com%2Fdog-picture.png%22%3E

```
<div>
    Hello Alice
<img src='https://example.com/dog-picture.png'>
</div>
```

Example: XSS

```
<div>
    Hello {get name from url}
</div>
```

https://example.com?name=Alice%3Cscript%3E%7Bstuff%7D%3C%2Fscript%3E

```
<div>
 Hello Alice<script>stuff</script>
</div>
```

Example: XSS

```
<div>
    Hello {get name from url}
</div>
```

https://example.com?name=Alice%3Cscript%3E%7Bstuff%7D%3C%2Fscript%3E

```
<div>
 Hello Alice<script>stuff</script>
</div>
```

Cross-Site Scripting (XSS)

Cross-site scripting (XSS) is a class of attacks which extends Content Injection to execute JavaScript in the victim's browser.

Source: https://github.com/oxsobky/HackVault/wiki/Unleashing-an-Ultimate-XSS-Polyglot

Cross-Site Scripting (XSS)

Cross-site scripting (XSS) is a class of attacks which extends Content Injection to execute JavaScript in the victim's browser.

- Often use "polyglots" to test websites

- `jaVasCript:` : A label in ECMAScript; a URI scheme otherwise.
- `/*-/*`/*\`/*'/*"/**/` : A multi-line comment in ECMAScript; a literal-breaker sequence.
- `(/* */oNcliCk=alert() )` : A tangled execution zone wrapped in invoking parenthesis!
- `//%0D%0A%0d%0a//` : A single-line comment in ECMAScript; a double-CRLF in HTTP response headers.
- `</stYle/</titLe/</teXtarEa/</scRipt/--!>` : A sneaky HTML-tag-breaker sequence.
- `\x3csVg/<sVg/oNloAd=alert()//>\x3e` : An innocuous svg element.

Source: https://github.com/oxsobky/HackVault/wiki/Unleashing-an-Ultimate-XSS-Polyglot

SQL Injection

SQL injection occur when untrusted input is not properly sanitized resulting in the addition of server-executed code.

```
def get_account_info(username):
    query = 'SELECT * FROM acct_table'
    query += 'WHERE username == "' + username + '"'
    query += ';'

    sql_db.execute(query)
```

SQL Injection

SQL injection occur when untrusted input is not properly sanitized resulting in the addition of server-executed code.

```
def get_account_info(username):
    query = 'SELECT * FROM acct_table'
    query += 'WHERE username == "' + username + '"'
    query += ';'

    sql_db.execute(query)
```

```
alice
```

SQL Injection

SQL injection occur when untrusted input is not properly sanitized resulting in the addition of server-executed code.

```
def get_account_info(username):
    query = 'SELECT * FROM acct_table'
    query += 'WHERE username == "' + username + '"'
    query += ';'

    sql_db.execute(query)
```

```
alice
bob
```

SQL Injection

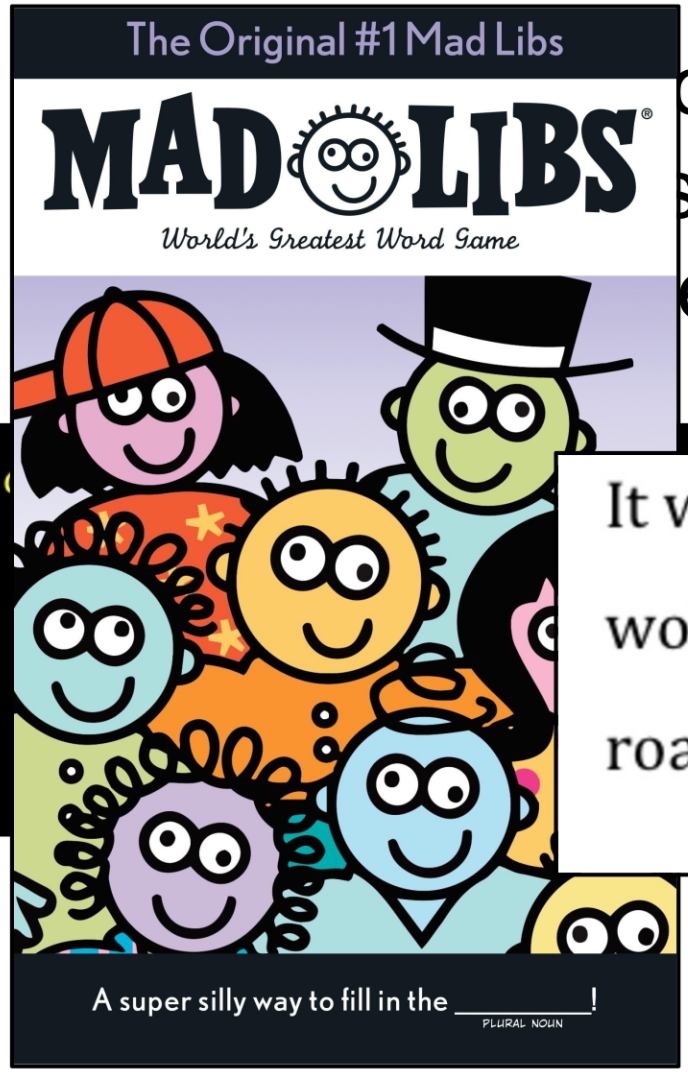
SQL injection occur when untrusted input is not properly sanitized resulting in the addition of server-executed code.

```
def get_account_info(username):
    query = 'SELECT * FROM acct_table'
    query += 'WHERE username == "' + username + '"'
    query += ';'

    sql_db.execute(query)
```

```
alice
bob
" OR 1==1
```

SQL Injection



occur when untrusted input
sanitized resulting in the
er-executed code.

It was a _____ , cold November day. I
adjective
woke up to the _____ smell of _____
adjective type of bird
roasting in the _____ downstairs. I
room in a house
down the stairs to see if I could

**** Injection

SQL injection-style attacks are **NOT** limited to SQL or even database queries.

- Commonly in the form of “shelling out”

```
def log_event(username):
    cmd = 'echo "' + username + ' did a thing" >> event.log'
    os.system(cmd)
```

Directory/Path Traversal

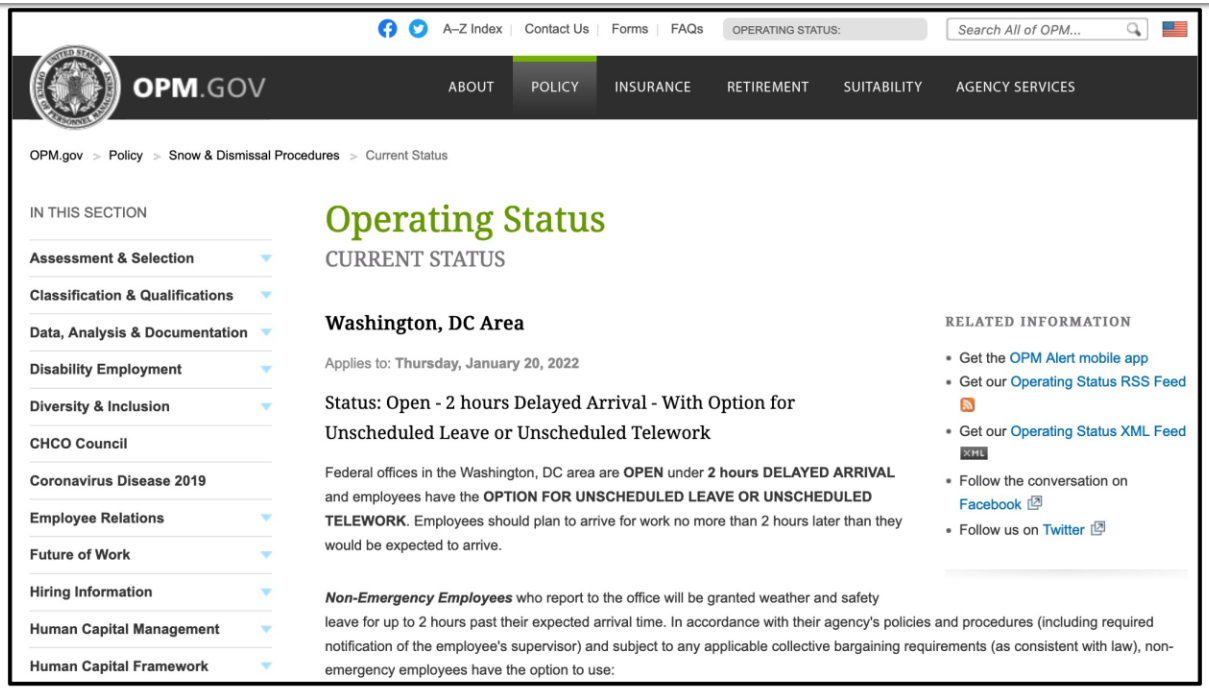
Directory/Path Traversal occurs when a *filesystem path* is unexpectedly altered to the attacker's target location.

```
def load_image_file(image_name):
    path = '/www/img/' + image_name + '.png'
    handle = open(path, 'r')
    data = handle.read()
    handle.close()
    return data
```

Drive-By Download/Compromise

- Legitimate user accesses a malicious web server and downloads/executes malicious functionality
  - JavaScript, iframe, etc
- Vulns allow modification of webserver/app/database
  - Chat sessions, forum posts, comments, etc
- **Watering Hole Attack**: Resource used by a target community and existing usage serves as advantage and vector

Watering Hole Example



A–Z Index | Contact Us | Forms | FAQs | OPERATING STATUS: | Search All of OPM...

OPM.GOV

ABOUT | POLICY | INSURANCE | RETIREMENT | SUITABILITY | AGENCY SERVICES

OPM.gov > Policy > Snow & Dismissal Procedures > Current Status

IN THIS SECTION

- Assessment & Selection
- Classification & Qualifications
- Data, Analysis & Documentation
- Disability Employment
- Diversity & Inclusion
- CHCO Council
- Coronavirus Disease 2019
- Employee Relations
- Future of Work
- Hiring Information
- Human Capital Management
- Human Capital Framework

Operating Status

CURRENT STATUS

Washington, DC Area

Applies to: **Thursday, January 20, 2022**

Status: Open - 2 hours Delayed Arrival - With Option for Unscheduled Leave or Unscheduled Telework

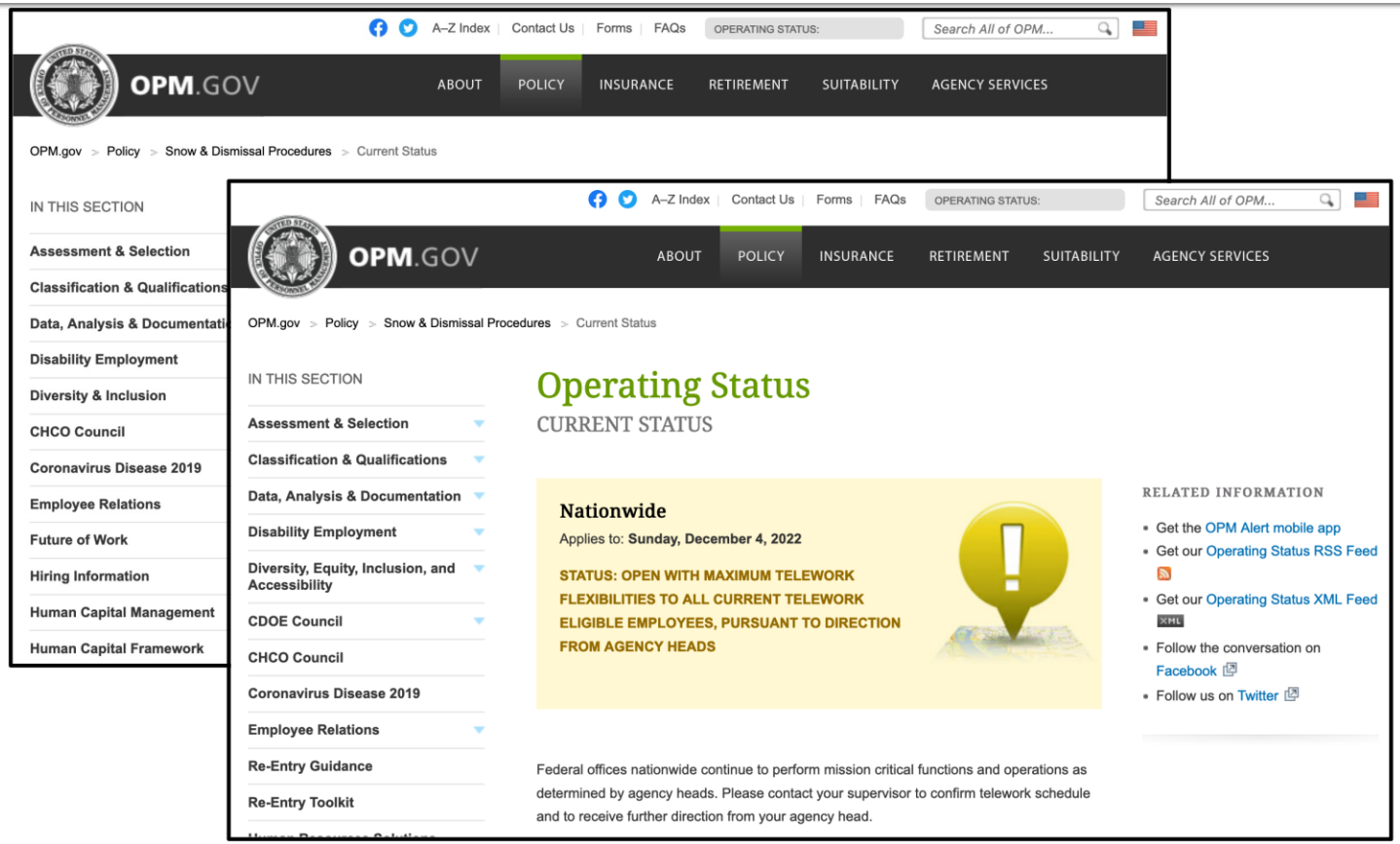
Federal offices in the Washington, DC area are **OPEN** under **2 hours DELAYED ARRIVAL** and employees have the **OPTION FOR UNSCHEDULED LEAVE OR UNSCHEDULED TELEWORK**. Employees should plan to arrive for work no more than 2 hours later than they would be expected to arrive.

Non-Emergency Employees who report to the office will be granted weather and safety leave for up to 2 hours past their expected arrival time. In accordance with their agency's policies and procedures (including required notification of the employee's supervisor) and subject to any applicable collective bargaining requirements (as consistent with law), non-emergency employees have the option to use:

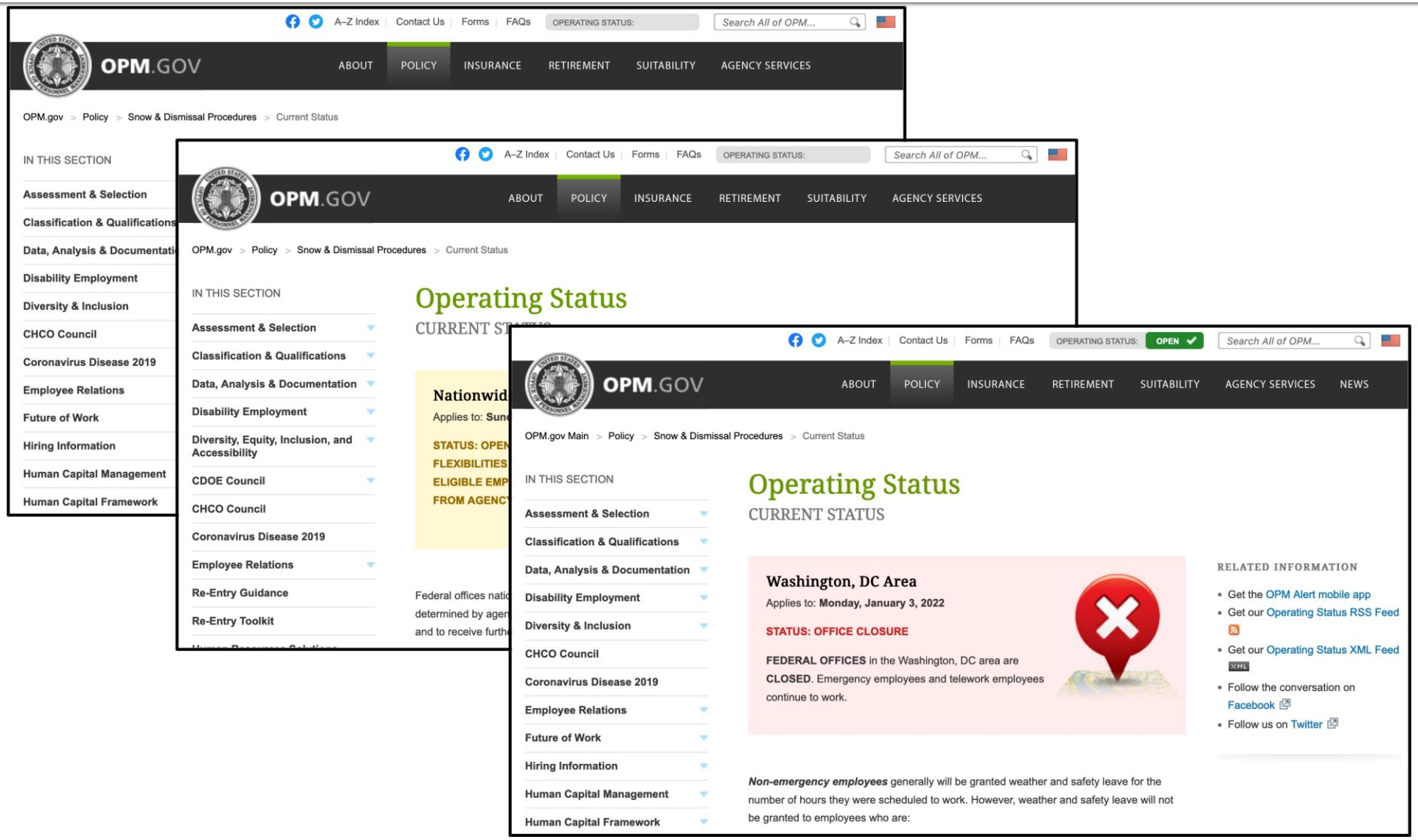
RELATED INFORMATION

- Get the OPM Alert mobile app
- Get our Operating Status RSS Feed
- Get our Operating Status XML Feed
- Follow the conversation on Facebook
- Follow us on Twitter

Watering Hole Example



Watering Hole Example



Supply-Chain Compromise

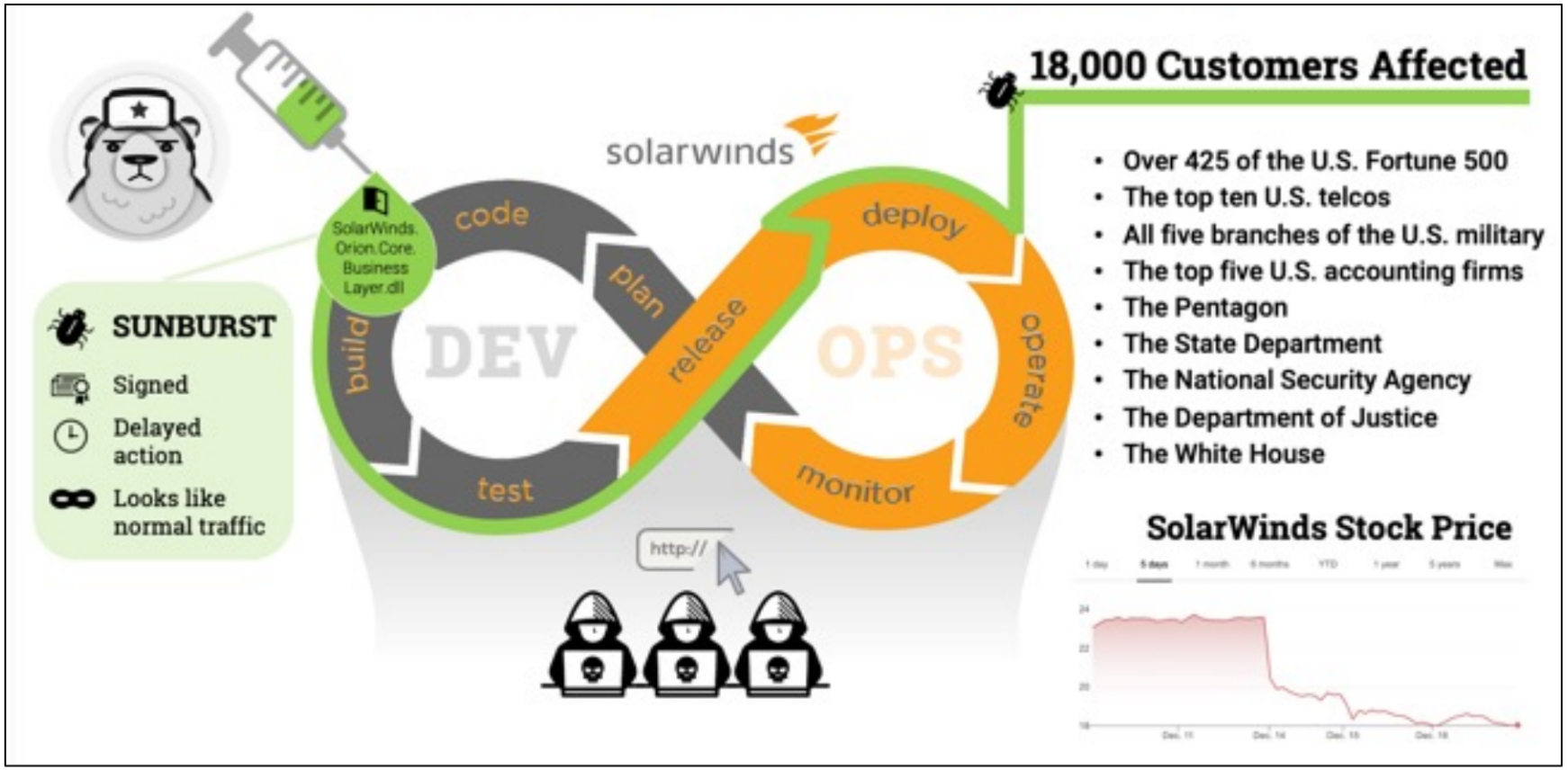
A **supply-chain compromise** occurs when malicious functionality is added to products *prior to target receiving them*.

Supply-Chain Compromise

A **supply-chain compromise** occurs when malicious functionality is added to products *prior to target receiving them*.

- Can occur at any point prior to the target
  - Modify code/config/build-tools during dev
  - Replace product prior to distribution
  - Replace product during distribution
- Can be via software or hardware

Supply-Chain Compromise



Supply-Chain Compromise

(TS//SI//NF) Such operations involving **supply-chain interdiction** are some of the most productive operations in TAO, because they pre-position access points into hard target networks around the world.

(TS//SI//NF) Left: Intercepted packages are opened carefully; Right: A "load station" implants a beacon

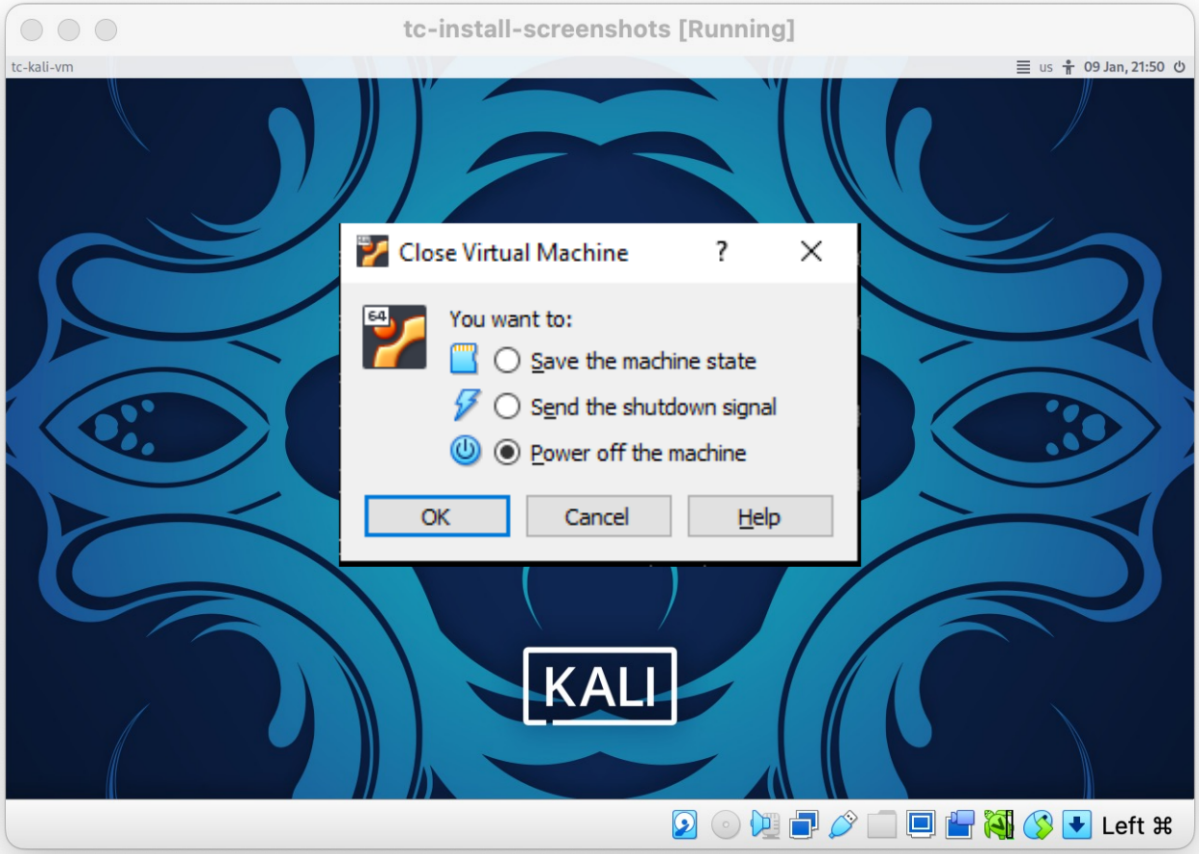
https://arstechnica.com/tech-policy/2014/05/photos-of-an-nsa-upgrade-factory-show-cisco-router-getting-implant/

Threats and Countermeasures

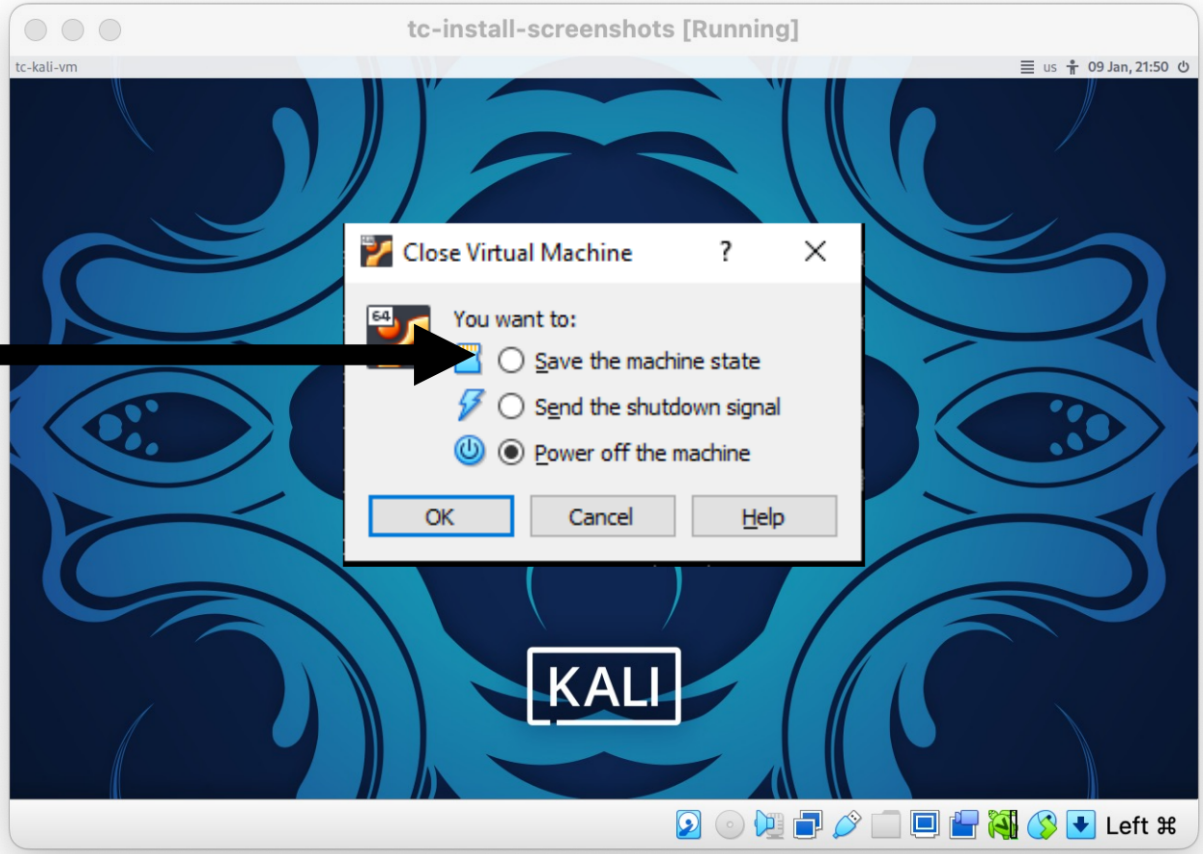
Lecture 04:
Resource Development & Initial Access

COMP-5830/-6830
Spring 2025

Kali NIC Configuration

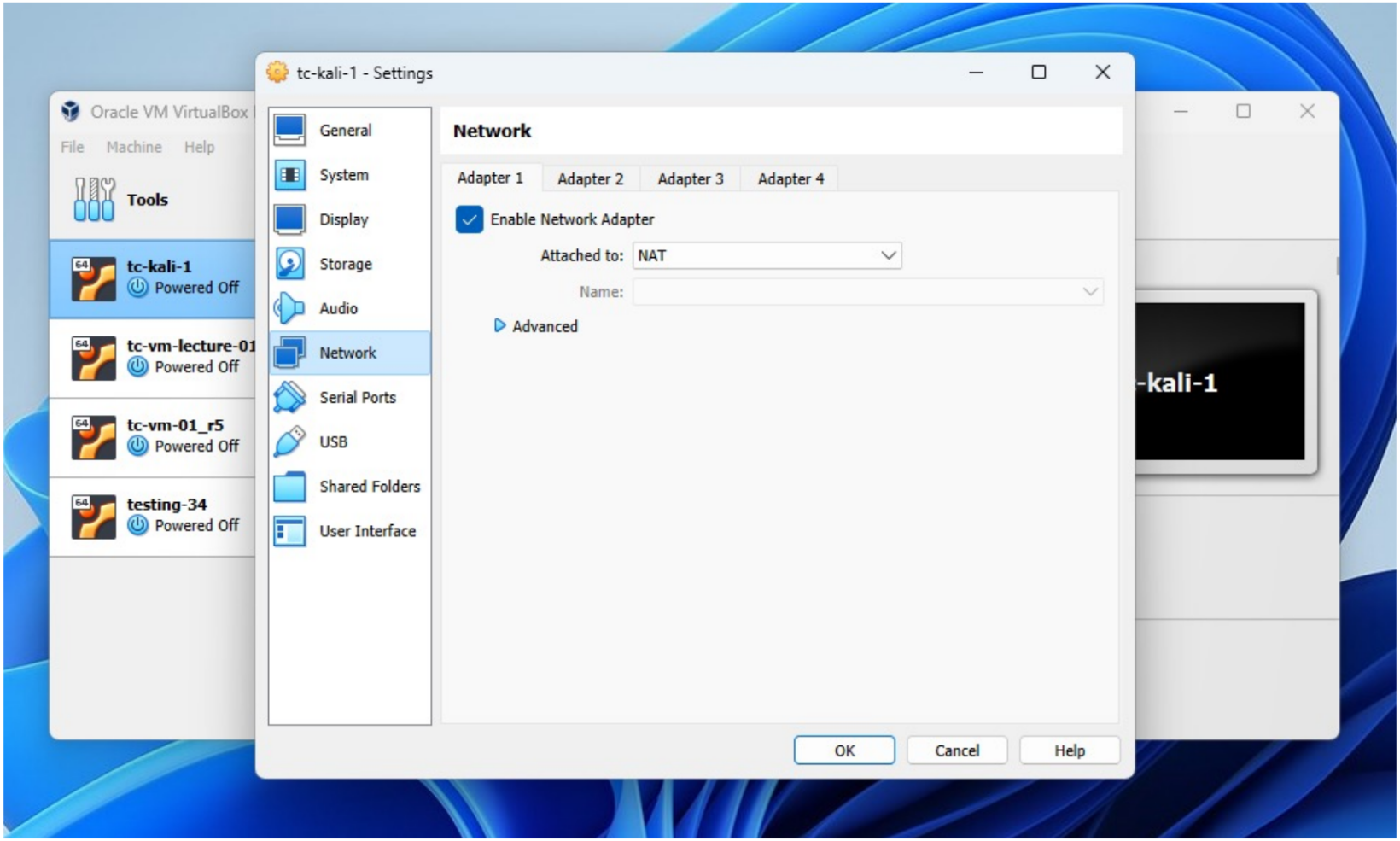


Kali NIC Configuration

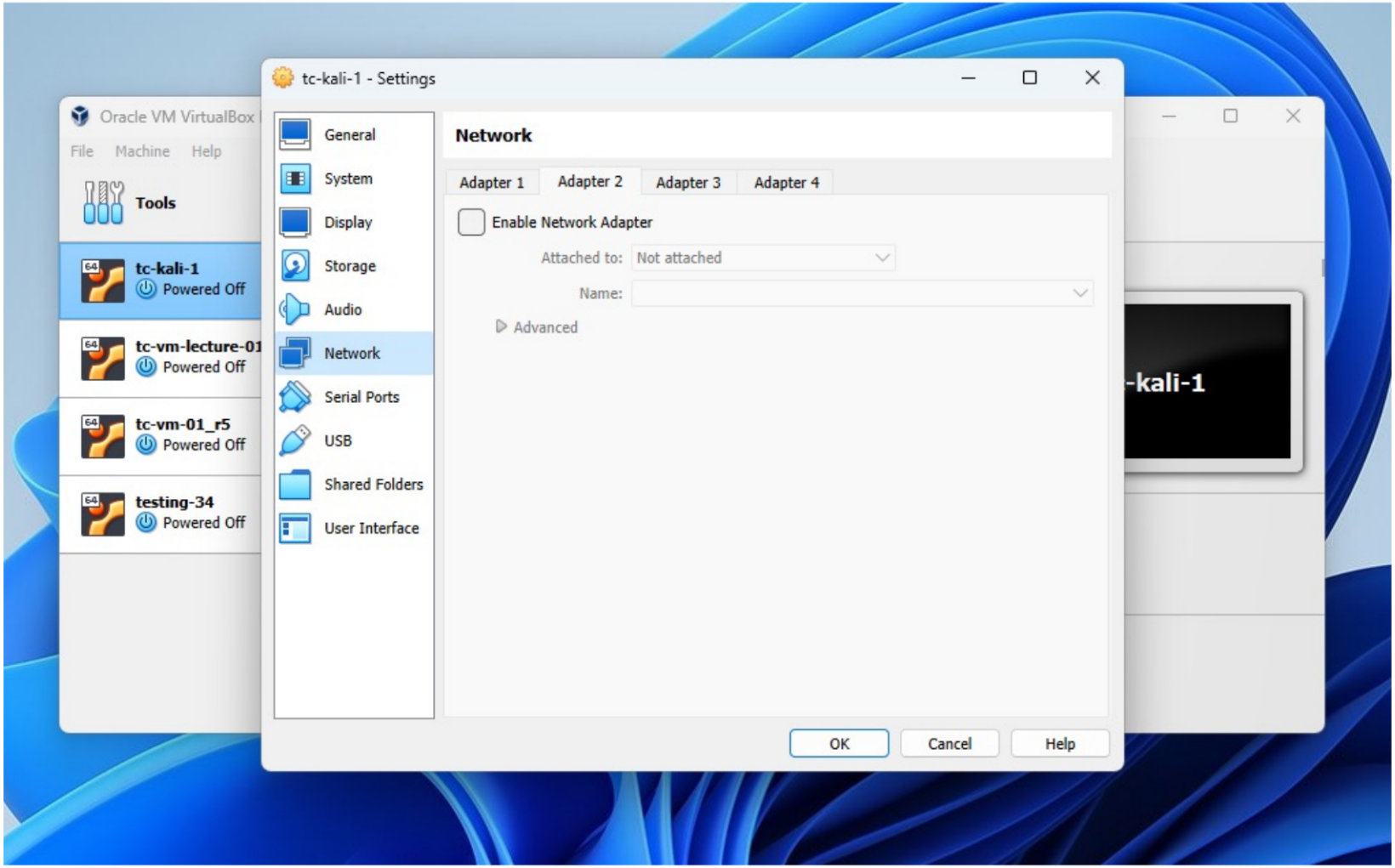


No

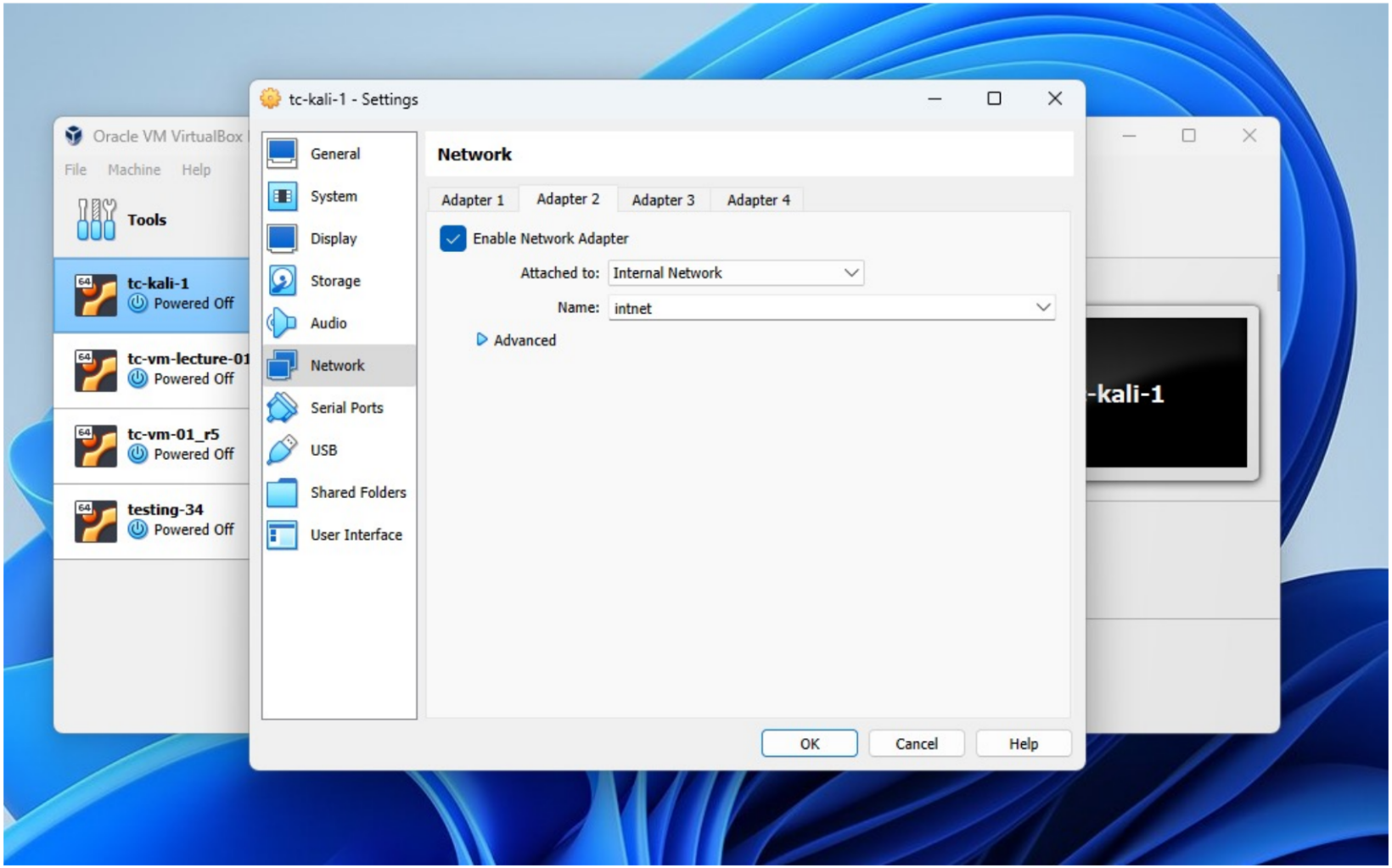
Kali NIC Configuration



Kali NIC Configuration



Kali NIC Configuration



Per-Startup Kali Config

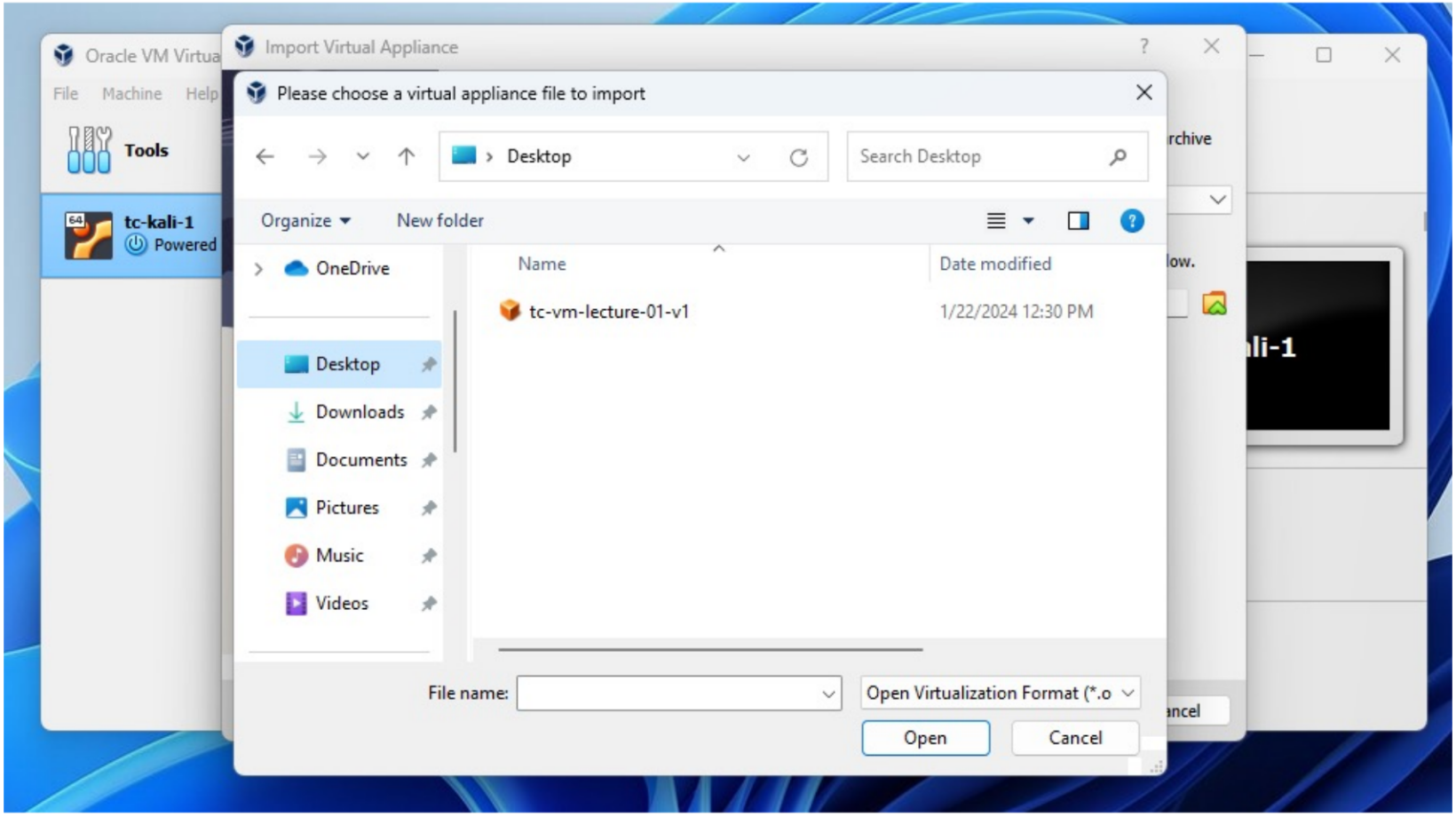
When connecting to the local VM, have to setup Kali networking ***every time you boot***.

- Stop auto-detect
  - `ifconfig eth0 down`
- Set Kali VM's IP address
  - `ifconfig eth0 192.168.66.XXX netmask 255.255.255.0`

May have to set 1x

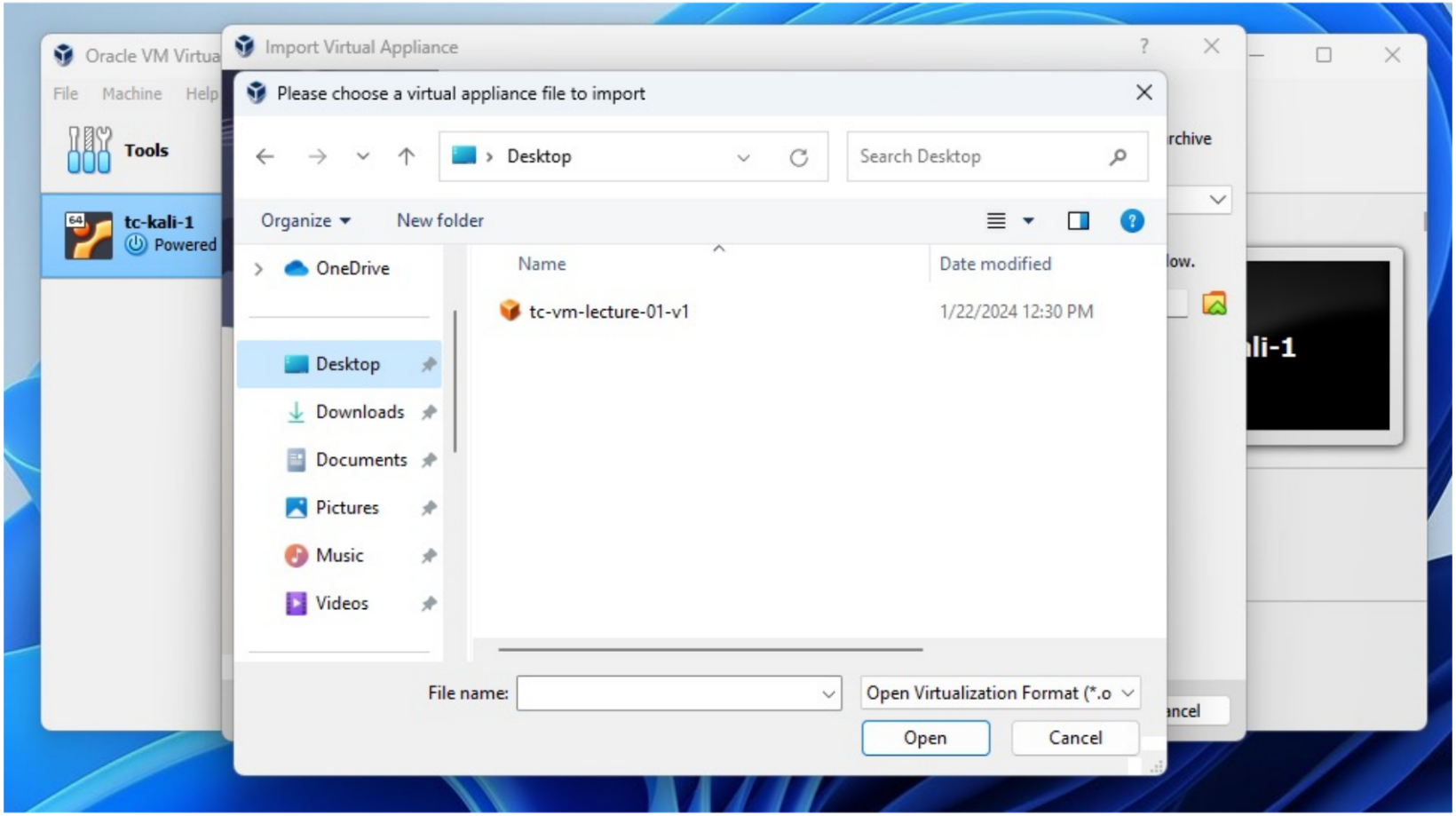
- `ip route add 192.168.66.0/24 dev eth0`

Today: tc-vm-03_r06



Disk Crypto: ZVcNabdysWUnmo32MRkDZ4TC

Today: tc-vm-01_r11



Disk Crypto: cY9hFa2hXuSgePTQNjhbsfob

Hardening Acct

```
Username: corpadmin
Password: password123
```

Ideas from Last Week

- Setup Firewall Rules
- Remove Password SSH Access
- User Permissions Audit/Downgrade
- App/Service Audit/Minimize