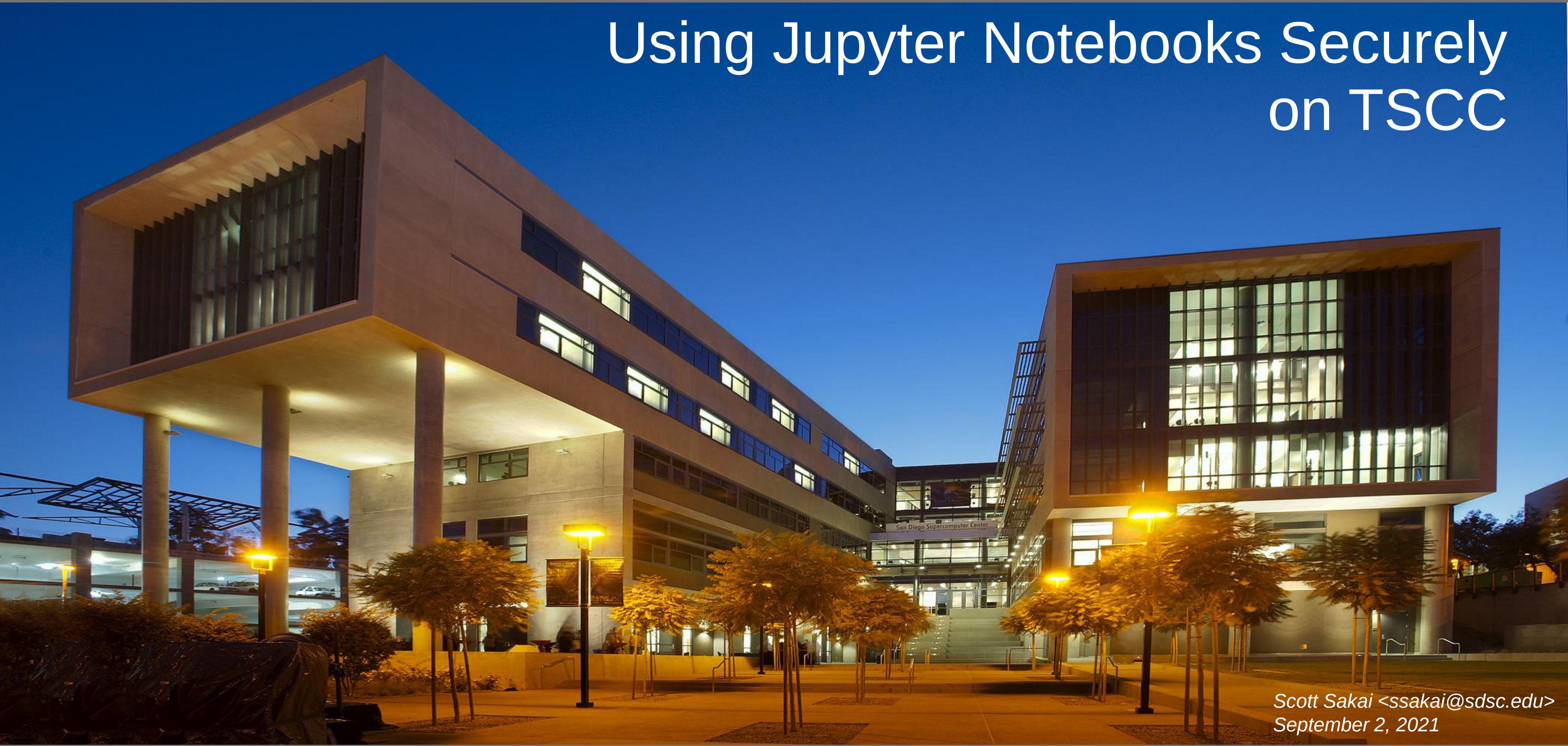
Using Jupyter Notebooks Securely on TSCC



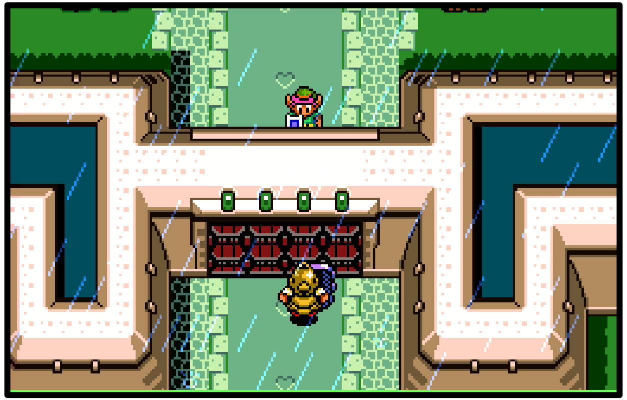
Scott Sakai <ssakai@sdsc.edu>

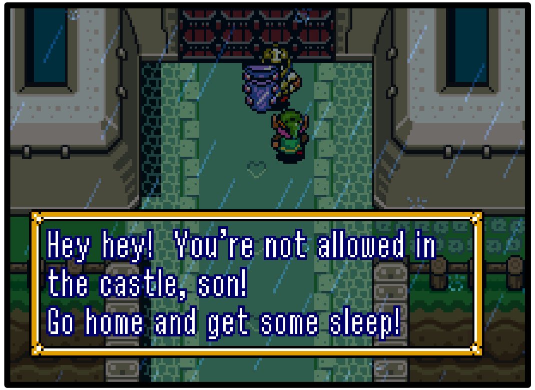
September 2, 2021

SDSC SAN DIEGO SUPERCOMPUTER CENTER

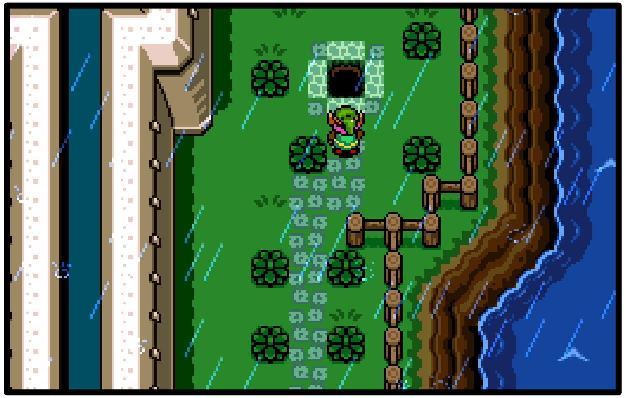
UC San Diego

Understanding Backdoors



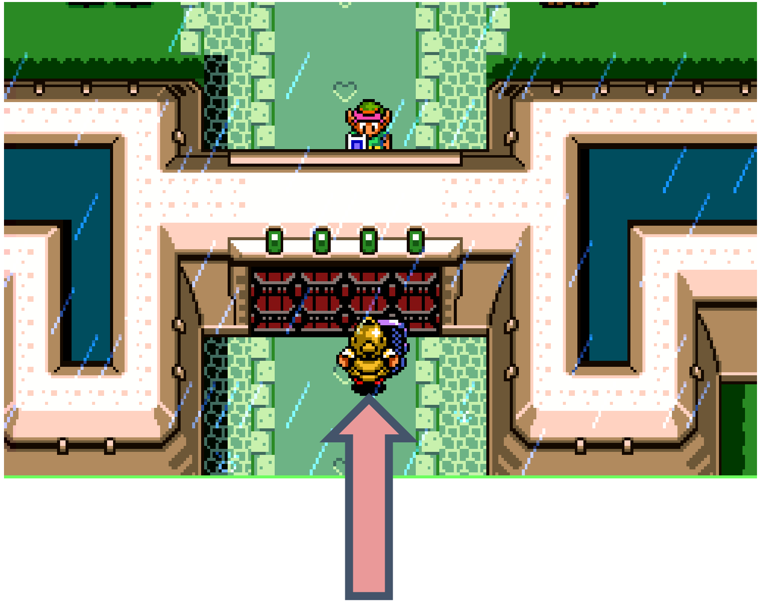


Hey hey! You’re not allowed in
the castle, son!
Go home and get some sleep!



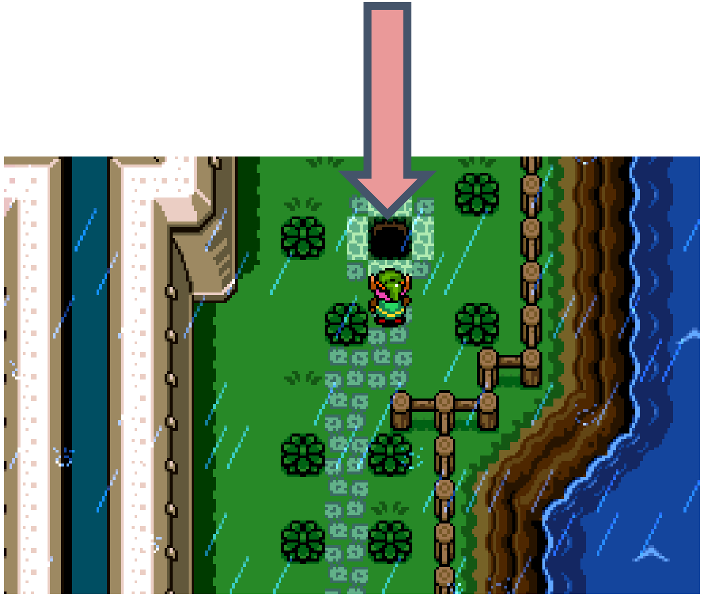
Game images from *The Legend of Zelda: A Link to the Past*

Understanding Backdoors (cont)



SSH

Jupyter Notebook



Game images from *The Legend of Zelda: A Link to the Past*

Jupyter Notebooks Aren't Bad

Use them safely and securely.

Why Security Matters

- There’s nothing interesting here.
- Nobody wants to attack me.

¯_(ツ)_/¯

Why Security Matters (cont)

There’s nothing interesting here.

- Research / Proprietary Data

  Collect it all, figure out how to monetize it later.

- Network Identity / Resources

  Launch attacks from TSCC since it’s “On Campus”.
  Launch attacks on other places to hide real identity.

- CPU/GPU Cycles

  Cryptocurrency!

Why Security Matters (cont)

Nobody wants to attack me.

- Unlikely to be Targeted

  But this doesn’t mean nothing will happen...

- You Can Still be a *Victim*!

  Most attacks involving an account are opportunistic. The wrong place at the wrong time.

  There may be nothing interesting, but the attacker needs to break in to learn that.

Why Security Matters (cont)

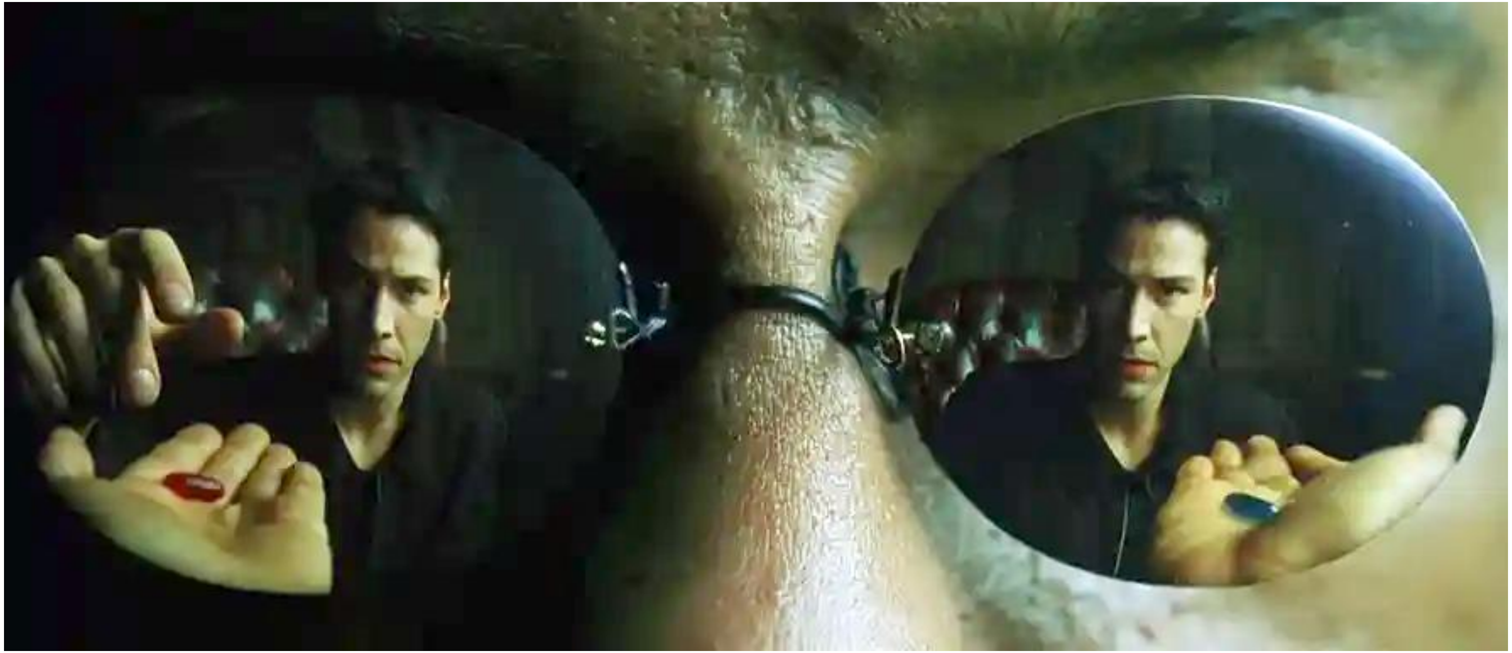
We Don’t Know Their Motives

Would you trust research based on tampered data?

Do you have backups?

Down the Rabbit Hole

Real-Life Experiences



Down the Rabbit Hole (cont)

The Internet is Nosy

In the Last 24 Hours, TSCC:

- Had 2.5 *million* SSH log-in attempts with invalid username/password.

- Received over 10,000 connection attempts to the default Jupyter Notebook port (8888).

Down the Rabbit Hole (cont)

Sometimes *They* Get In

“There’s strange files in my home directory,
I didn’t put them there.”

Down the Rabbit Hole (cont)

Sometimes *They* Get In

When were the files created?

“... BTW, I already deleted them.”

Down the Rabbit Hole (cont)

Sometimes *They* Get In

Do we have snapshots with the files?

Yes, we do!

- When were the files created?

  A very specific date and time, “T”

- Was the account logged in to around time T?

  Yes, from a foreign IP “F”, using an SSH key.

Down the Rabbit Hole (cont)

Sometimes *They* Get In

- Were any new SSH keys added?
  No.

- Are there any SSH keys in the account's home directory?
  Yes…

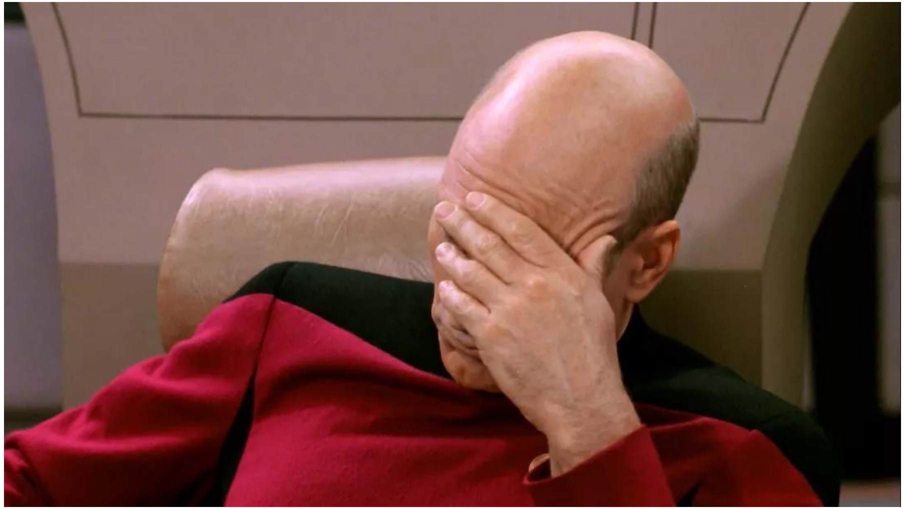
- When was the SSH key file last accessed?
  Just before time T.

Down the Rabbit Hole (cont)

Sometimes *They* Get In

- Is there a passphrase on it?

  … No.



Down the Rabbit Hole (cont)

Sometimes *They* Get In

- What else was going on around time T from host F?

  Connections to port 8888 just before time T.

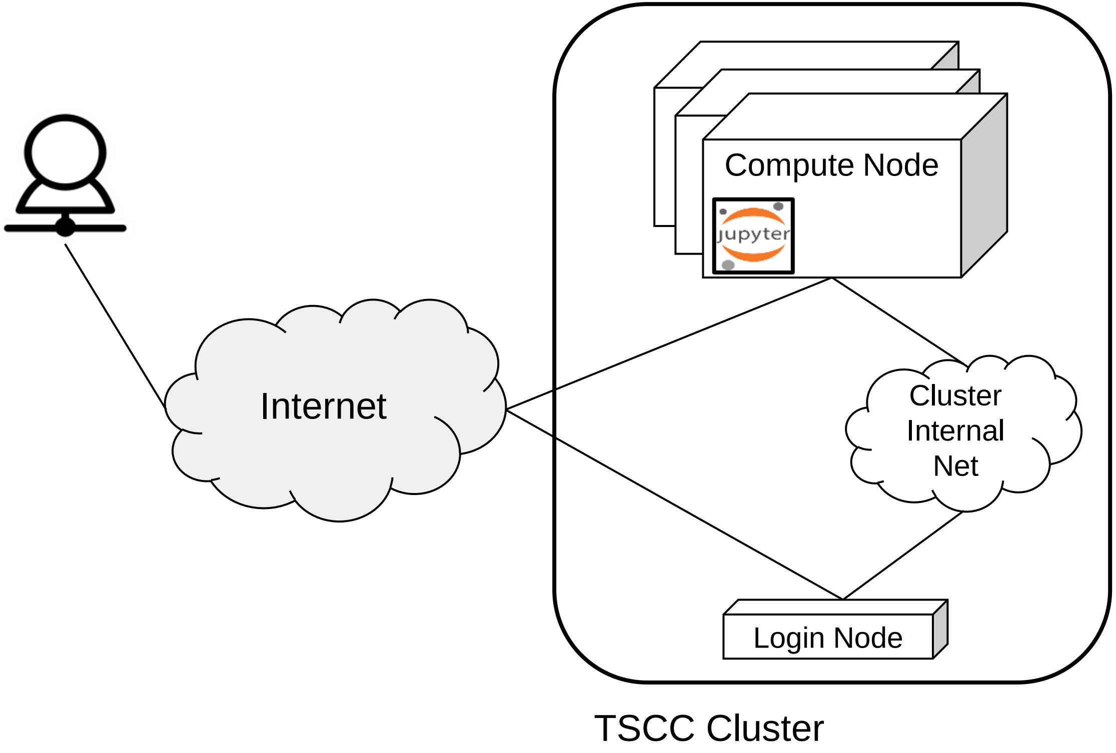
Down the Rabbit Hole (cont)

Sometimes *They* Get In

Sequence of Events:

- Attacker finds Jupyter Notebook and uses it to retrieve SSH key from account’s home directory.
- Attacker logs in to account using SSH key.
- Attacker installs crypto-miner software.
- User discovers software.

Jupyter Notebooks and TSCC



Questions?

Up next: How to Use Jupyter Notebooks **Insecurely**

How to Use Jupyter Notebooks Insecurely

Don’t Set A Password / Use A Weak Password

- Remember: TSCC is **always** being probed.
- A notebook will be found, it’s just a matter of time.

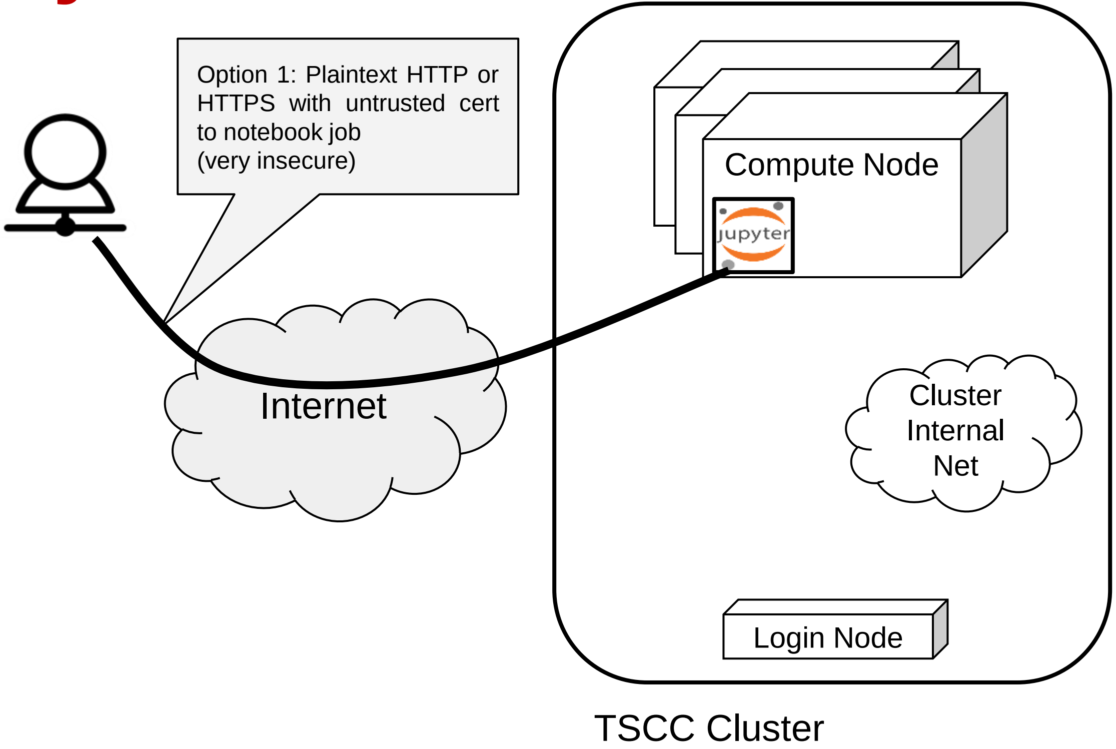
How to Use Jupyter Notebooks Insecurely

Don’t Set A Password / Use A Weak Password

- Demo: Creating an insecure notebook with no password

- Demo: Discovering an insecure notebook with no password

How to Use Jupyter Notebooks Insecurely



Option 1: Plaintext HTTP or HTTPS with untrusted cert to notebook job (very insecure)

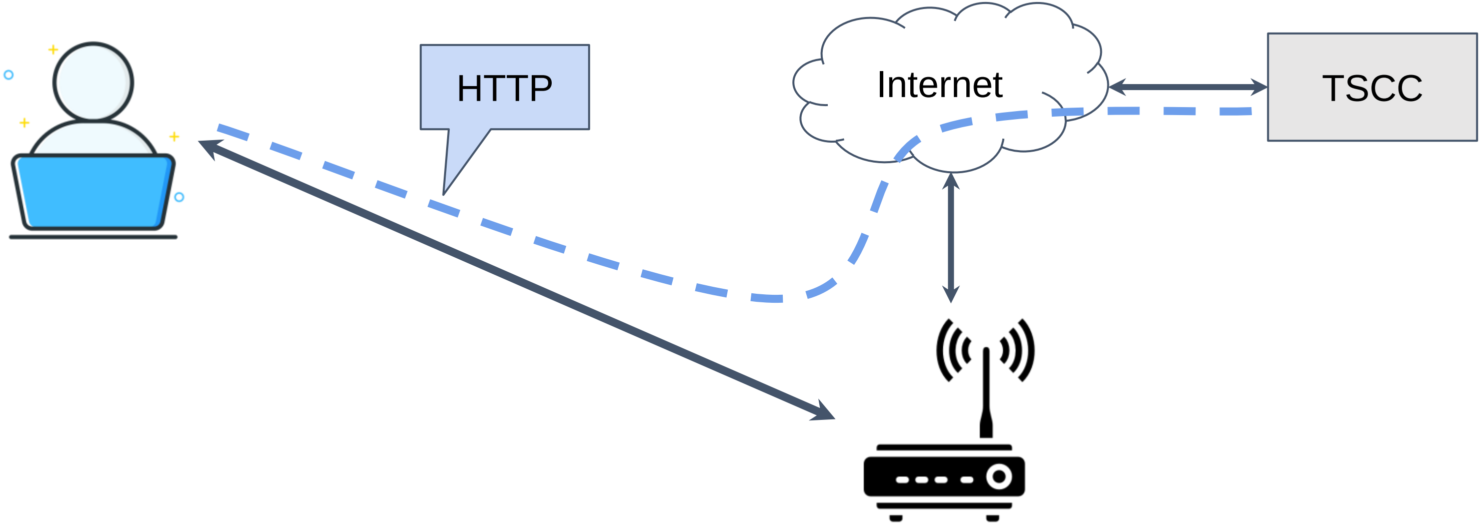
How to Use Jupyter Notebooks Insecurely

Use Plaintext HTTP (or untrusted SSL)

- Plain HTTP (no S) transfers web site data without protection against viewing or alteration.
- An attacker can capture passwords.
- An attacker can capture cookies, bypassing passwords.

How to Use Jupyter Notebooks Insecurely

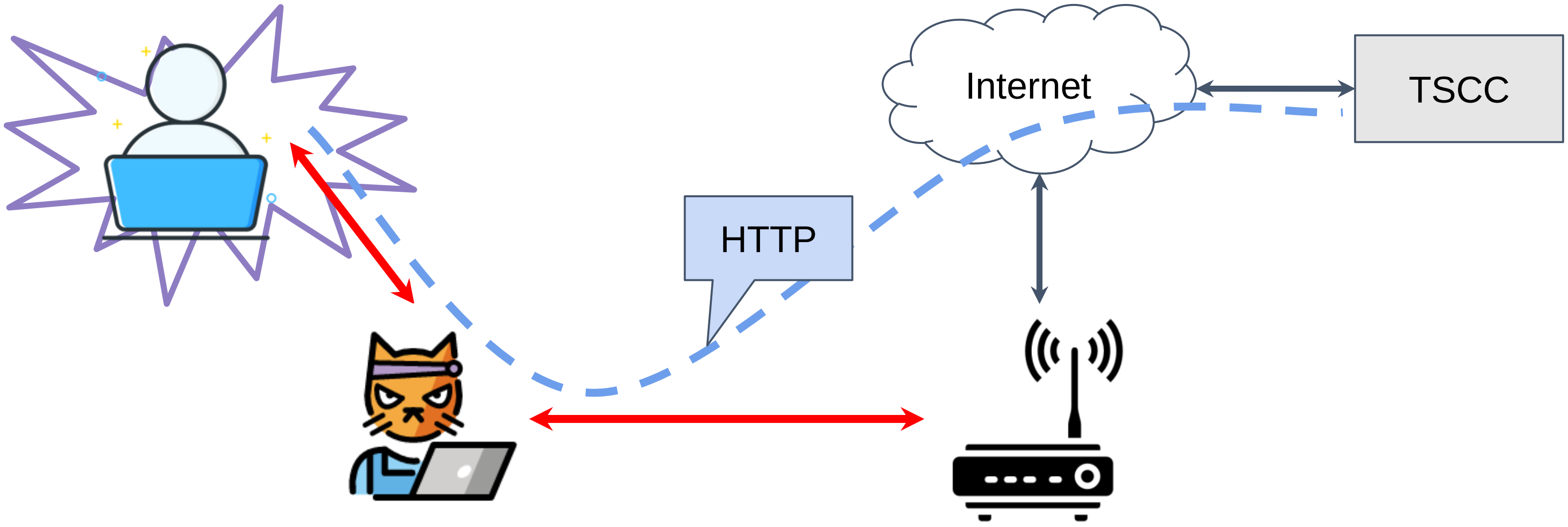
Use Plaintext HTTP (or untrusted SSL)



Background: “Man-In-the-Middle” (MITM) Attack

How to Use Jupyter Notebooks Insecurely

Use Plaintext HTTP (or untrusted SSL)



Background: “Man-In-the-Middle” (MITM) Attack

How to Use Jupyter Notebooks Insecurely

Use Plaintext HTTP (or untrusted SSL)

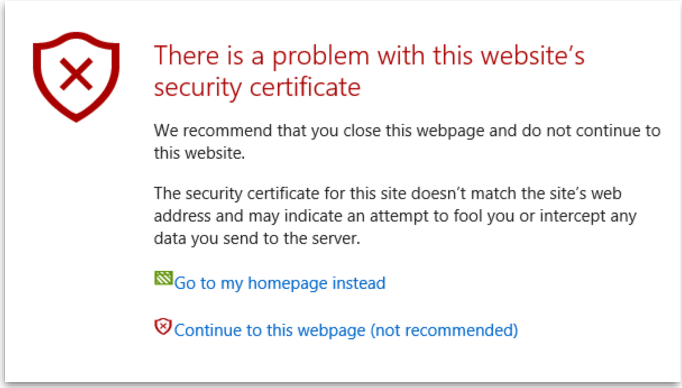
- Demo: Breaking into a passworded notebook with MITM

How to Use Jupyter Notebooks Insecurely

Use Plaintext HTTP (or untrusted SSL)

- SSL is essential.

- Browser-trusted certificate also essential.



There is a problem with this website's security certificate

We recommend that you close this webpage and do not continue to this website.

The security certificate for this site doesn't match the site's web address and may indicate an attempt to fool you or intercept any data you send to the server.

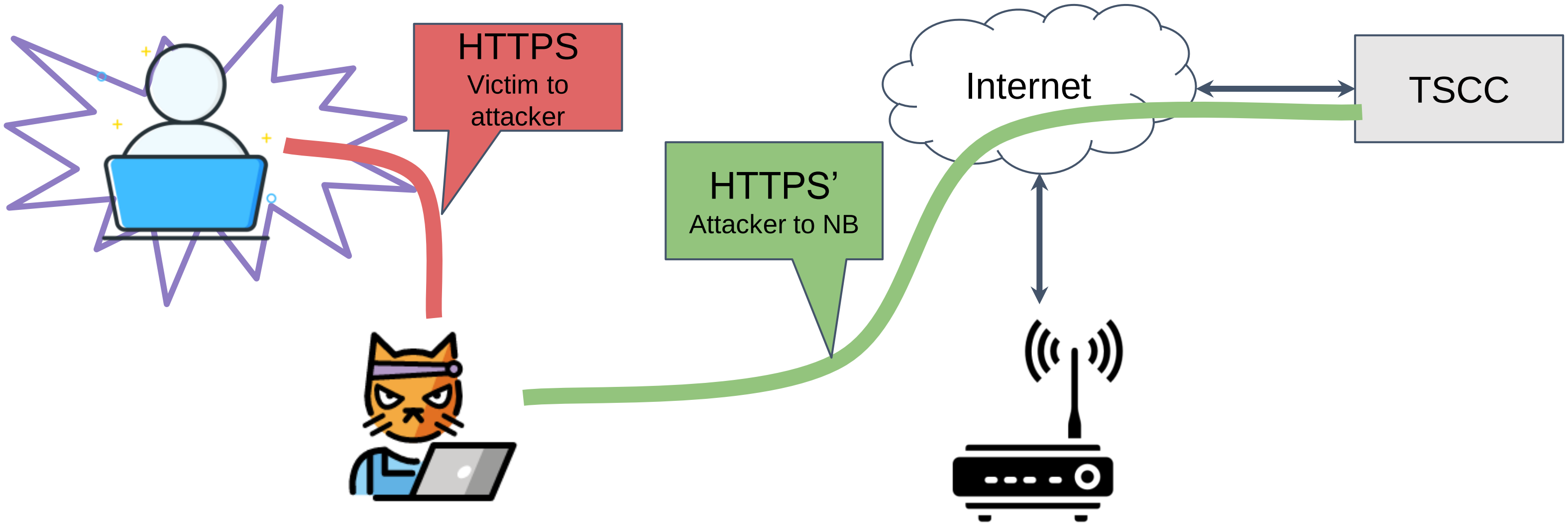
Go to my homepage instead

Continue to this webpage (not recommended)

- MITM still works if the certificate warning is ignored!

How to Use Jupyter Notebooks Insecurely

Use Plaintext HTTP (or untrusted SSL)



Background: “Man-In-the-Middle” (MITM) Attack with HTTPS

Questions?

Up next: How to Use Jupyter Notebooks Securely

How to Use Jupyter Notebooks Securely

- Set a long password, or better yet, use the auto-generated token.
- Launch the notebook in a way that prevents it from being directly reachable from the Internet.
- Use encryption over *unfriendly* networks.
  (The Internet is unfriendly, so is wifi.)

How to Use Jupyter Notebooks Securely (cont)

- Set a long password, or better yet, use the auto-generated token.

- Launch the notebook in a way that prevents it from being directly reachable from the Internet.

- Use encryption over *unfriendly* networks.
  (The Internet is unfriendly, so is wifi.)

How to Use Jupyter Notebooks Securely (cont)

- The `jupyter notebook` command takes an `--ip` argument.
- Use the node’s internal IP or internal hostname.
- The internal hostname can be found by adding `.local` to the short hostname.

```
jupyter notebook --ip $(hostname -s).local
```

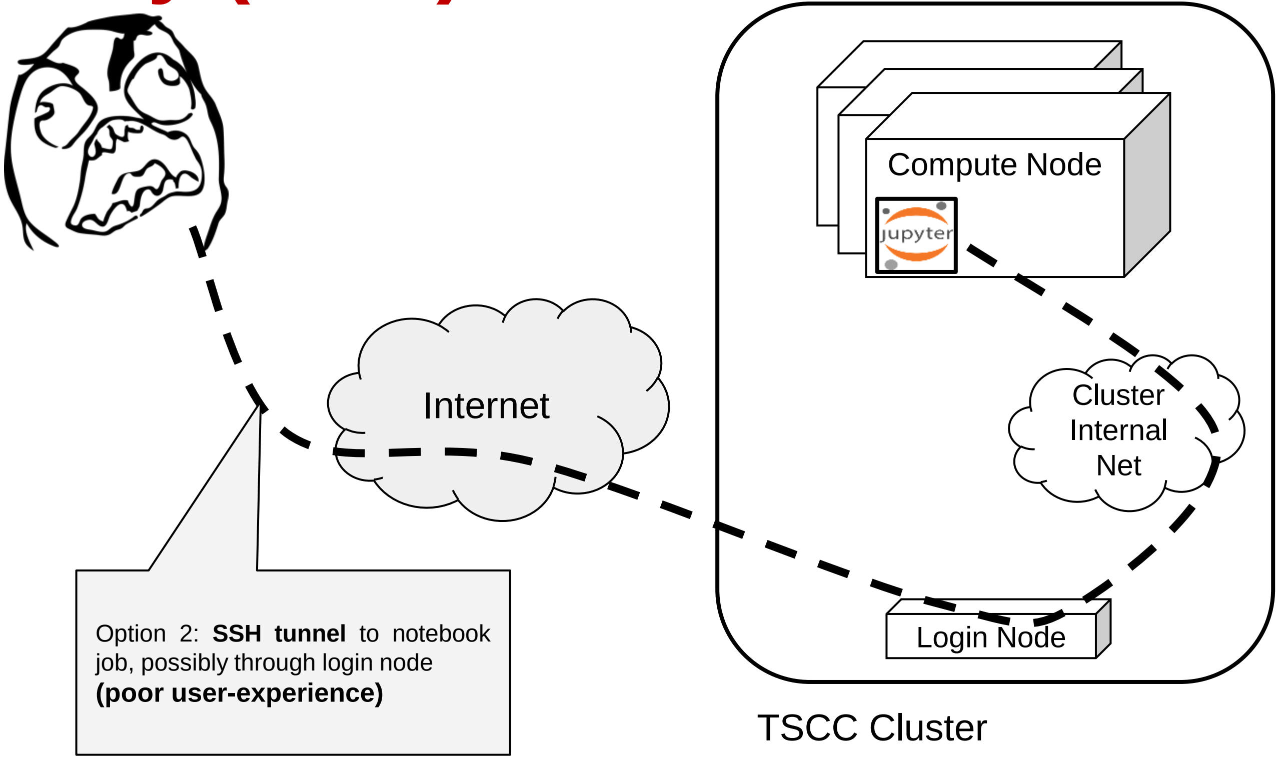
How to Use Jupyter Notebooks Securely

- Set a long password, or better yet, use the auto-generated token.

- Launch the notebook in a way that prevents it from being directly reachable from the Internet.

- Use encryption over *unfriendly* networks.
  (The Internet is unfriendly, so is wifi.)

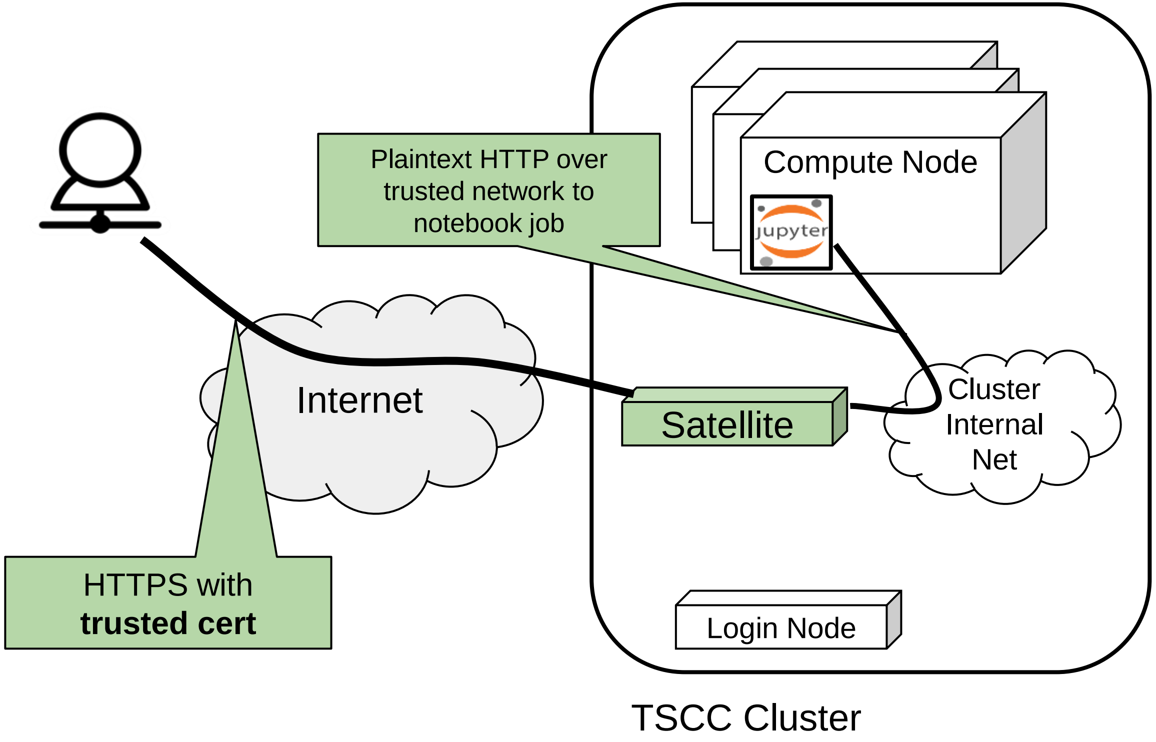
How to Use Jupyter Notebooks Securely (cont)



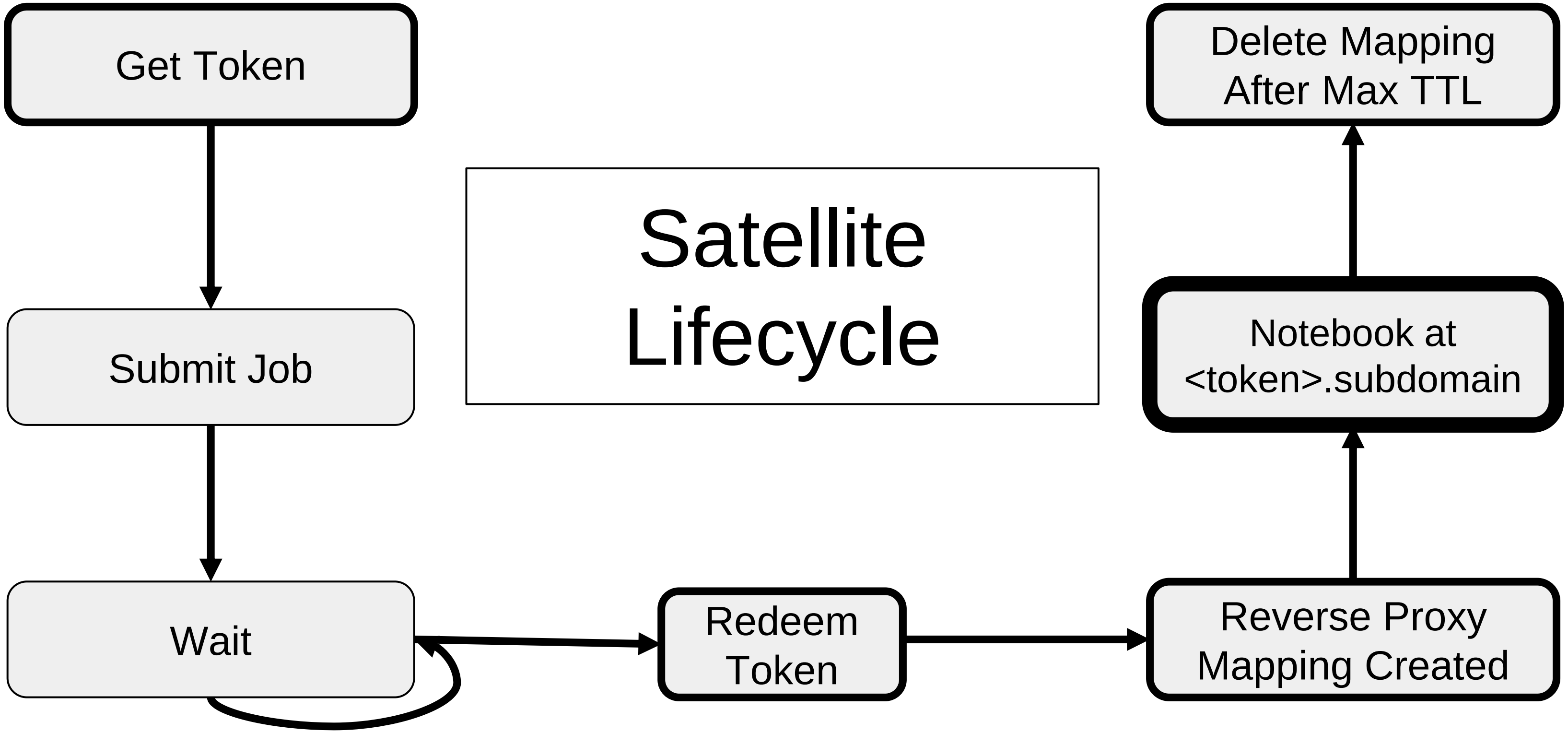
Option 2: **SSH tunnel** to notebook job, possibly through login node **(poor user-experience)**

TSCC Cluster

How to Use Jupyter Notebooks Securely (cont)


Compute Node
jupyter
Plaintext HTTP over trusted network to notebook job
Internet
Satellite
Cluster Internal Net
HTTPS with **trusted cert**
Login Node
TSCC Cluster

Satellite Lifecycle


Get Token
Submit Job
Wait
Redeem Token
Reverse Proxy Mapping Created
Notebook at <token>.subdomain
Delete Mapping After Max TTL

How to Use Jupyter Notebooks Securely (cont)

- Demo: Manual example of Satellite API

Parting Thoughts

- There’s more to security than this presentation!
- Did you backup your data?
- Patch everything! (Python modules get security updates too!)
- Protect your account to protect your research.

Accidents can happen, **it’s not your fault**.

Questions?

Thank You!