Cybersecurity user requirements analysis: the ECHO approach.

KATOS, V., KI-ARIES, D., FAILY, S., GENCHEV, A., BOZHILOVA, M. and STOIANOV, N.

2022

Cybersecurity user requirements analysis: the ECHO approach

Vasilis Katos[1][0000-0001-6132-3004], Duncan Ki-Aries[1][0000-0002-2859-1143], Shamal Faily[1][0000-0002-2859-1143], Angel Genchev[2][1111-2222-3333-4444], Maya Bozhilova[2][1111-2222-3333-4444] and Nikolai Stoianov[2][0000-1111-2222-3333]

[1] Fern Barrow, Talbot Campus, Poole, Bournemouth, Dorset BH12 5BB, GB
[2] Defence Institute "Prof. Tsvetan Lazarov", bul. "Prof. Tsvetan Lazarov" 2, Sofia, Bulgaria
vkatos@bournemouth.ac.uk dkiaries@bournemouth.ac.uk
sfaily@bournemouth.ac.uk a.genchev@di.mod.bg
m.bozhilova@di.mod.bg n.stoianov@di.mod.bg

Abstract: Cyber defense requires research and investment in advanced technological solution as well as in the development of effective methods and tools for identifying cyber threats and risks. This implies a need for a well-defined process for user requirements elicitation. The paper presents a structured approach for the identification of cybersecurity knowledge and elicitation of user needs, based on the development of specific use cases. Employing use cases is an effective way to identify the cyber security gaps. Example use case descriptions of the attacks on a general computer network are given. The proposed use cases are analyzed within CAIRIS platform. The modelling process confirms that CAIRIS is a powerful tool to enrich the context of threat models and UML class diagrams. Also, the modelling with CAIRIS could support using security-by-design principles. The research is conducted under the activities of "The European network of Cybersecurity centres and competence Hub for innovation and Operations" (ECHO) project.

Keywords: Cyber defense, User requirements elicitation, Use case analysis, CAIRIS modelling.

1 Introduction

Cyber defense of critical systems and citizen is a challenging task at the national and regional level. Cyber attacks can devastate Critical Infrastructure organizations, such as those part of the Health Care, Energy, or Security sectors, where damage can lead to loss of life. Because such attacks may not be limited to a single organization or state, collaboration is necessary to address common cyber threats and challenges.

Recognizing the need for collaboration to address increasing cyber threat, the European Union (EU) established a programme to create a European cyber security ecosystem. The ECHO (European network of Cybersecurity centres and competence Hub for innovation and Operations) project is one of the four pilot projects, funded by the European Commission, to create a cybersecurity competence network [1].

ECHO involves 30 industry and research partners from 14 member states. It aims to strengthen the EU's proactive cyber defense, improve the technological capabilities of a secure digital market, and protect the European citizens against cyber attacks more generally. With such a diverse range of partners and ambitious objectives, a consistent approach for eliciting and specifying requirements is important. The techniques used or capturing requirements needs to be easy for different partners to adopt, and requirements need to be managed using affordable tools that are likely to remain maintainable both now and in the future.

The use-cases are widely used as a technique for eliciting requirements for software systems, because of the benefits they provide. They can be used to elicit and specify requirements a user's perspective, they are effective for communicating with stakeholders, bringing hidden requirements in the minds of the stakeholders to the surface where they can be specified [2]. Nasr et al. [ibid] illustrate such benefits by using use cases to specify requirements for the embedded systems in the avionics domain. Faily et al. [12] demonstrate the usability of the use cases approach designing security and usability into a non-trivial software system for a research and development project.

Because use cases are an effective way to identify the user needs and cyber security knowledge gaps in different sectors, Task 2.1 of ECHO used use cases to understand the scope and complexity of different problem domains, share knowledge of domain-specific business knowledge and threat models, including the modus operandi of actors and attackers. In doing so, they helped standardize and streamline the storytelling and narrative of attacks in the sectors of concern, using a structure that enabled (i) requirements elicitation for information sharing models, (ii) the development of the curricula and modelling for enhancing cybersecurity/cyberdefense skills through the federated cyber range, (iii) inputs for the identification of sector-specific and inter-sector security challenges, (iv) inputs for the identification of technological challenges and opportunities, (v) the scoping and development of demonstration cases.

In this paper, we present our approach for use-case driven requirements analysis for ECHO. In Section 2, we present the definitions adopted for the specific task's needs. In Section 3, we provide an overview of the main features of the CAIRIS platform [4], which we used to model, analyze and validate the ECHO use cases. We illustrate the approach taken to elicit storylines and use cases in Section 4, and our approach for subsequently modelling and analyzing them with CAIRIS in Section 5. We conclude in Section 6 by briefly describing benefits and implications of our approach.

2. ECHO approach for a use case definition

The ECHO concept for representation of the cybersecurity vulnerabilities includes three components: scenario, storyline and use case. Each identified sector contains a single scenario, which aggregates multiple storylines. The storyline includes a set of use cases based on the common type of attacks or the common infrastructure (asset). This set of use cases is about user needs that should be addressed. Figure 1 presents the hierarchical relation between the three terms – scenario, storyline, use case.

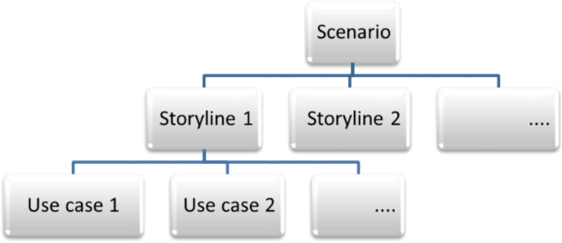


Fig. 1. Relations: scenario – storyline – use case.

The words scenario, storyline and use case are being used differently in different areas and perceived differently by different authors. For the goals of the ECHO project, we have used the following definitions:

- Use case – A use case describes an interaction between attackers and the system/systems under attacks. Each use case has the following mandatory attributes: a name, a unique identifier, and a step-by-step description of a basic course of action. Some use cases also describe the system's states at each step, exception conditions and variant paths. (adapted from "A technical discussion on Use Case Best Practices", 06/11/03, IBM) [3]. Although the definition originates from Software Engineering, the clear way in which it describes the attacker-system interactions and corresponding outcomes makes it suitable for cyber security needs analysis.
- Storyline – A storyline groups several use cases sharing common infrastructure.
- Scenario – A set of storylines, related to the activities of malicious actors aimed to damage, theft or destroy assets in a specific sector (domain).

One of the first challenges we encountered when developing a use case template was the ambiguity over how to write use case descriptions, and choosing the right level of detail. Several books have been written on appropriate ways to write a good use case description. Based on an analysis of the best practice, we concluded that the usability of use cases is greatly enhanced by adopting a common structure for writing the descriptions. To address the project's objectives and needs, the template for ECHO use cases description was proposed. A template example for particular application is presented in Section 4.

3. Use case modelling with CAIRIS

Many tools support the management of use cases, but few support the management and analysis of complementary security design concepts used by ECHO.

CAIRIS (Computer Aided Integration of Requirements and Information Security) is an open-source platform for designing both security and usability into system specifications [4], and is a tool exemplar for how security and usability engineers might collaboratively address security and privacy problems at the earliest stages of the design process [5].

CAIRIS does not prescribe any particular design technique or methodology, and supports a wide variety of usability, security, and specification concepts. These include many of the concepts proposed in our approach. It supports the specification of use cases, assets and all the elements feeding into a risk analysis process. It also supports threat modelling using Data Flow Diagrams (DFDs) and attack trees. Faily et al. [13] present an approach for reasoning about tainted data flows in design-level DFDs.

CAIRIS supports a number of features which make it useful as a collaborative design platform. First, it supports the automatic generation of visual models, which can be dynamically created as model elements are added, updated, or deleted. Many visual models scale poorly because the bounded rationality bias makes it difficult to manage models that have become too complex to comprehend. By generating models, providing the ability to filter model, and providing different views of the same system, CAIRIS overcomes this problem. The most relevant visual models for our objectives include:

(I) Asset models: these are based on UML class diagrams;

(II) Task models: these are augmented UML use case models, which include additional information on task usability and misuse case risk impact;

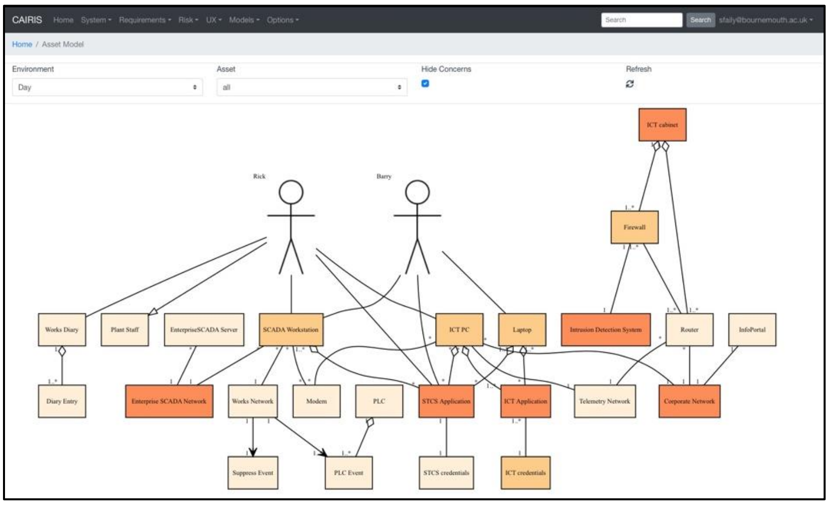
(III) Risk models: these quickly visualize the elements contributing to risk, and can be categorized by metadata, e.g. ATT&CK tactic [6];

(IV) Goal models: based on the KAOS [7] modelling language, these can show the system goals that exemplar system need to satisfy, and obstacles that obstruct these. These obstacles are attack trees.

(V) DFDs: these illustrate data flows between entities, processes and data stores in exemplar systems, and trust boundaries these cross.

Figure 2 illustrates the CAIRIS asset model, and shows how security information is used to augment the UML class model elements. The actor figures indicate personas that interact with tasks that use assets in this particular context. The shading of red indicates the asset attack surface based on vulnerabilities the assets are currently exposed to; the darker the shade of red, the more exposed the attack surface is.

CAIRIS has been designed for interchangeability. CAIRIS model files are based on XML, and are intentionally both human and machine-readable. CAIRIS supports import/export is supported for a number of different file interchanges, specifications can be generated in PDF, OpenDocument Text and Word, and CAIRIS has a well-documented REST API [8].


Environment: Day

Asset: all

Hide Concerns ☑

Fig. 2. CAIRIS asset model example.

Third, CAIRIS has been released under a permissive open-source license allowing both ECHO partners and user to consume models without any cost. CAIRIS can be freely used and extended as part of the ECHO project without any restrictions. Finally, unlike many open-source projects, CAIRIS has been heavily documented and, as part of the CAIRIS source code, and its documentation and tutorials are revised frequently as features are added and updated.

4. An example storyline and use cases definitions

The approach used is based on end-users threats in corresponding sectors. The ECHO end-users are organizations, whose main business is in the selected sector of interest. They support the project with expert knowledge about the cybersecurity issues in their sector. Therefore, their knowledge of attacks are considered to be among the highest probable threats in the particular sector. The identified threats are modeled by use cases. The subsequent paragraphs present an example of use case concept applied to the military domain.

Storyline: Attacks on a HQ General computer network.

Story: There is an EU-supported peace-keeping mission in a zone of conflict. For the needs of the mission, a HQ is established. It is situated in a military base. A general computer network is deployed within the military base. Every officer has a workstation (mobile computer) with Ethernet connection to the network. The network topology is given in Figure 3. An Active Directory (AD) domain controller is used to manage the user accounts in the network. There are adversary governments, who want to create a bad image for the peace-keeping mission about being unsuccessful. So, they have decided to support a local terrorist organization (noted as TORG) with information about the technology needed to steal information from the HQ. TORG, for its part, is particularly interested in headquarters plans and the personal information of military staff. There is a hostile state (noted as HST), which is interested in compromising the EU mission image. HST agents support TORG with know-how. The Internet is provided by a DSL modem via domestic operator who provides Internet also for the needs of the national airport, located in that area. In the dormitory and the recreational places of the military base, a Wi-Fi network is deployed.

The HQ has planned to organize a convoy for fuel and other supplies to the local airport. It is scheduled for a predefined date. They have written the necessary documents and orders including a letter for the port authorities.

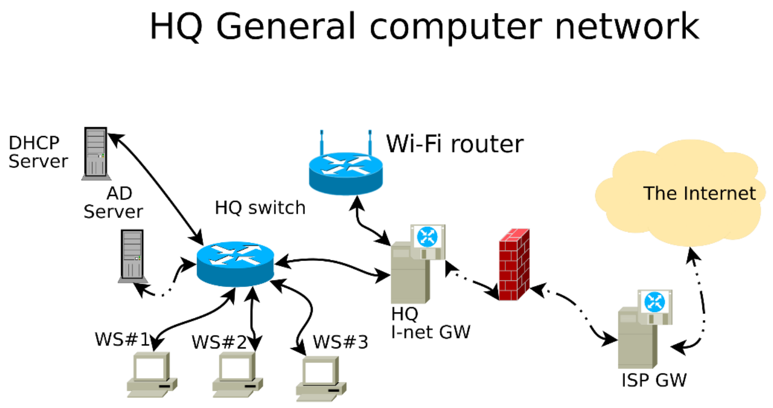


Fig. 3. The HQ network infrastructure.

Storyline's context: The storyline and its associated use cases will take place in the defense domain. An attack on the network would allow sensitive data ex-filtration towards the opposing military forces (TORG in this case).

Storyline's objectives:

Through realizing this storyline and its use cases:

- The strength of the general network security will be tested
- Possible future attacks towards computer networks will be prevented and/or damage will be mitigated
- The cyber defense staff will be educated to know about and defend against the described types of attacks.

Cyber-attack/s description/classification:

- Threat:
  - Computer viruses
  - Spyware
  - Hackers (High technological opposing forces)
  - Personnel (human factor).
- Exploited vulnerability:
  - Wi-Fi firmware vulnerabilities
  - Possible Man-in-the-middle (MITM)

- Personnel
- Missing data encryption
- OS command injection
- Unrestricted download of dangerous file types
- Reliance on untrusted inputs in a security decision
- Cross-site scripting and forgery
- URL redirection to untrusted sites
- Path traversal
- Weak passwords
- Software that is already infected with a virus.

Assets/processes affected: data, files, passwords, hardware (firmware)

Type of attacker: Terrorist group, supported by a hostile state

Purpose of attack: Compromising the public confidence in the peace-keeping mission via support of terrorist group activities (military attacks) with valuable information.

The storyline "Attacks on a HQ General Computer network" includes three use cases: UC01 "Wi-Fi router firmware attack", UC02 "Man-in-the-middle attack", and UC03 "USB flash stick malware attack".

4.1 UC01 "Wi-Fi router firmware attack"

Name: Wi-Fi router firmware attack

Initial stage: Everything works as desired.

Summary: Similar to the CIA's CherryBlossom [10]/Weeping Angel [11] toolkits are provided to TORG by its hidden supporters (agents from HST). In this UC, we call it "VapourMonger". The goal of TORG is to steal information. That's why they try to hack the Wi-Fi network and obtain information via faking common sites everybody uses. These sites are for example the venerable (in this UC) social network MasseHook. Because the attack needs to be performed remotely, TORG can use drones which they manage to land in the vicinity of the base or people carrying the equipment in backpacks.

Logical steps:

Step 1: A brute force WPA attack is started by TORG using its VapourMonger toolkit.

System state: Everything works as desired.

Step 2: Eventually the toolkit manages to find the Wi-Fi password on the 2nd day of the attack, and proceeds with Step#3.

System state: One of the Wi-Fi access points is under attack.

Step 3: A wi-fi admin password is found. The Wi-Fi access point attack phase#2 might find the password very quickly if a default password is held or might take some time if changed. Once found, the toolkit proceeds to the next step.

System state: The access point control panel page is under attack. Wi-Fi administrative password is known to the toolkit.

Step 4: The VapourMonger toolkit uploads a new firmware image in the access point. The malicious firmware supports MITM attack by re-routing the connections to the Internet via outside device back to the Wi-Fi router (because it needs Internet access). So, the toolkit starts a MITM attack.

System state: Wi-Fi password + Wi-Fi control panel page password are known to the toolkit.

Step 5: A fake (but having valid SSL certificate) clone of MasseHook social network page collects all passwords to the real MasseHook social network. Once a password is collected, the next TCP connections from the same IP/MAC address are passed-thru without intervention to allow the user to have normal service.

System state: Wi-Fi network password and Wi-Fi access point control panel page password are known to the toolkit. The firmware is altered. A MITM attack is in progress.

Initiating actor: TORG

Supporting actors: The HQ staff, HST government.

Final system state: On success: data leak. TORG knows the profiles of the EU mission staff, where do they live, their habits, who are their friends and family members.

Inputs: TORG knows where the military base is. (In fact, everybody knows that). Timing of the steps; MasseHook page requests from the users which are re-routed to the fake MasseHook page.

Outputs: HQ staff profile pages; friends lists and passwords for MasseHook.

4.2 UC02 “Man-in-the-middle attack”

Initial stage: HQ use DSL connection to a local internet provider. Computer network in the HQ operates as usual. The network administrator needs to download some drivers for a new scanner.

Summary: MITM Attack is performed with an aim to penetrate in HQ’s network, take control over workstations and AD controller, and download sensitive data.

Logical steps:

Step 1. A TORG attacker succeeds to mount a modified (for MITM) router on the communication line between the HQ and the Internet provider.

System state: Everything works as desired, but the communication passes through the attacker’s MITM device.

Step 2. A man-in-the-middle attack (MITM) is performed. Communication of HQ’s network with Internet is altered. URL, DNS responses are modified to point to a fake (mirror) site with modified binaries.

System state: The Internet access is controlled by the attacker.

Step 3. The administrator downloads a driver for his new scanner. His download request is redirected to a fake web site. The driver downloaded is an executable of self-unpacking installer type, crafted by the hackers to contain the real drivers and the advanced persistent threat (APT) malware.

System state: The driver executable is present on the administrator’s workstation (WS#1).

Step 4: The malware setup is executed on the administrator’s workstation (WS#1). It installs the malware and the drivers and starts the malware.

System state: The malware is active on the administrator’s workstation.

Step 5: The APT establishes a connection with its C&C server outside indicating its readiness for commands.

System state: The malware is active on the administrator's workstation. It's connected with the C&C server.

Step 6. By external commands, sensitive data from the Administrator's workstation is ex-filtrated. Hashes of user and Administrator passwords are sent to the C&C.

System state: The hashes and files of the administrator's folders are present on the attacker's side. The other state – no change.

Step 7: Attacker cracks the hashes and sends commands to gain administrative rights on the Administrator workstation.

System state: A process of the APT with elevated privileges is running on the administrator's workstation. The attacker has access to all active directory resources.

Step 8. Attacker succeed to download the plan for the petrol supply convoy and its schedule.

System state: Plans for the convoy route and schedule are known to TORG. Other processes: no change. The APT is ready for commands to disrupt the communication when TORG decides to launch a real attack on the convoy.

Initiating actor: TORG

Supporting actors: Northland intelligence team, HQ's system administrators, Local Internet Provider

Final system state: The HQ Infrastructure is infected. The APT is connected to C&C and serving requests from TORG.

Inputs: TORG intelligence team knows details about a local Internet provider and used cable connection to HQ; HQ's system administrator knows that he needs software drivers; Timings of the steps; The convoy plan

Outputs: Exfiltrated plans; User and Administrator passwords; HQ's network topology

4.3 UC03 "USB flash stick malware attack"

Initial stage: The described computer network infrastructure works normally without errors.

Summary: Local citizens are hired as support staff in the HQ. One of them – a poor citizen – Namir cleans the HQ office. A TORG agent recruits him for $ 200 with task to leave a USB flash drive in the HQ office. The flash stick is marked as EU unclassified information and is infected with malware. The unsuspecting officer on duty inserts the flash stick in his computer, which is Windows-based. Two months later, from a military unit, which defends the port at zone of conflict two platoon commanders are killed in Brussels on their annual leave. The TORG takes the responsibility.

Logical steps:

Step 1: The cleaner leaves the USB flash drive on the desk in the HQ office.

System state: The HQ computer network infrastructure is functioning normally.

Step 2: An hour later an officer on duty notices the "registered" USB stick and inserts it into WS#2 to see what it contains.

System state: The HQ computer still functioning normally.

Step 3: The worm accommodated at the flash stick is activated.

System state: The malware is active on WS#2.

Step 4: The malware (on WS#2) scans all files in the Documents folders. Then it uploads them to an attacker supplied site on the Internet.

System state: The malware is active on the user's workstation. The exfiltrated content is present on the attacker's host.

Step 5: System state: The malware is active on the user's workstation, waiting for new files to upload.

Initiating actor: The TORG hired employee.

Supporting actors: The officer on duty; HST cyber forces (with malware)

Final system state: A malware is active on WS#2. The uploaded content contains files of the HR with names and ranks of the staff with the plan for the annual leave of the staff.

Inputs: Timing of the steps; Plans (to be stolen); TORG knows who from locals has been hired by the EU mission.

Outputs: Exfiltrated files; Infected system.

5. An approach for modelling the example storyline within CAIRIS

This section presents an approach for turning an ECHO storyline and use cases into a CAIRIS model that validates and visualises described kill chains. Our approach surfaces assumptions that hidden in the storyline document and puts into context the threat intelligence contributing to the different risks leading to the final outcome.

The approach has three steps that identify and progressively refine the risk model elements.

Step 1: Initial Asset Identification

The first step entailed identifying explicit and implicit assets from the storyline and modelling the relationships between them. We identified an initial set of seven assets by browsing the storyline and use cases: Workstation, Domain controller, DSL modem, DHCP Server, HQ personal information, Wi-Fi access point, Base network.

In later steps, as we understood the storyline and the context a bit better, we elected additional assets being attacked, elaborated existing assets, and identified relationships between the assets. The final asset model is shown in Figure 4.

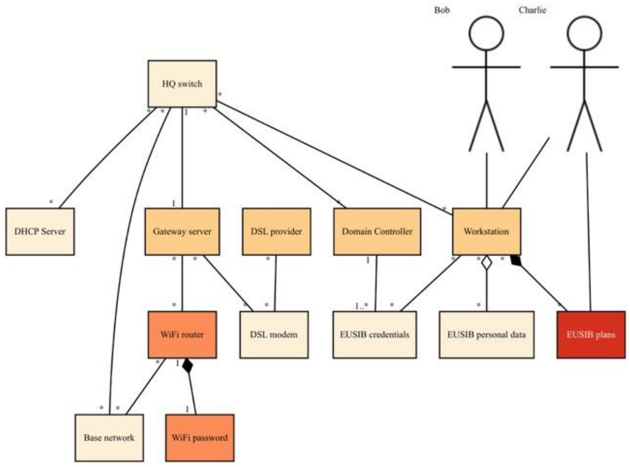


Fig. 4. Complete asset model for the storyline.

While modelling, it was also necessary to state assumptions in our model about things that were implicit. For example, references to the Wi-Fi password implied that a single credential was used to obtain access to the base network, which also included Internet access. However, deployment of active directory assumes additional credentials are needed to access resources on workstations.

Step 2. Rationalising the Attacker

We reviewed the storyline to understand the different system roles presented in the document. In Step 1, an ‘external user’ appeared to be present. In Step 2, the role inadvertently downloading the malware could be an administrator and, in Step 3, both roles are present. Given the context, we decided to define ’trusted’ and ‘untrusted’ roles.

It is then necessary to model attackers with the motivations and capabilities to carry out the attacks described. For this storyline, we identified three attackers:

- ‘Ardit’ (named after Ardit Ferizi) is an attacker with a certain amount of technical knowledge, and at least sufficient knowledge to properly operate the tools he has been given.
- ‘Bob’ is a network administrator.
- ‘Charlie’ is an officer on duty.
- ’Trudy’ is a cleaner.

Of these attackers, only Ardit and Trudy have any malicious intent. The other attackers are motivated primarily by productivity.

At this stage, we also modelled tasks and skeleton personas implicit from the storyline. Two tasks were identified. One of these (Patch kit) entails a skeleton persona (Bob) following procedures for doing a monthly update of all the workstations. This task was associated with the Workstation asset.

The other task (Starts shift) describes the procedures carried out by another skeleton persona (Charlie) who starts a shift as a duty operations officer - an activity that doesn’t preclude the checking of information on USB sticks. This task was associated with the Workstation asset and a newly identified HQ plans asset.

Normally, we would begin our analysis with some user research and the creation of personas. In this case, to avoid confusion, we name the attackers 'Bob' and 'Charlie' in CAIRIS as 'Bob Attacker' and 'Charlie Attacker'. The personas associated with these tasks are named Bob and Charlie respectively, to allow these user models to evolve should user research subsequently be carried out to better understand the user goals and expectations for the personas' associated roles.

Step 3. Modelling the Kill Chains

Each use case describes a kill chain, where each use case step describes a risk. Once the risk in each step is realized, it affords a possible vulnerability or threat that provides the foundation for the risk in the next step. Modelling the risk in each step entails identifying an exploitable vulnerability and the assets being exploited, the attack carrying out the threat, the threat itself and the threatened assets, and a misuse case scenario that puts the risk in context. When we applied this approach to the storyline, we produced the Risk model in Figure 5.

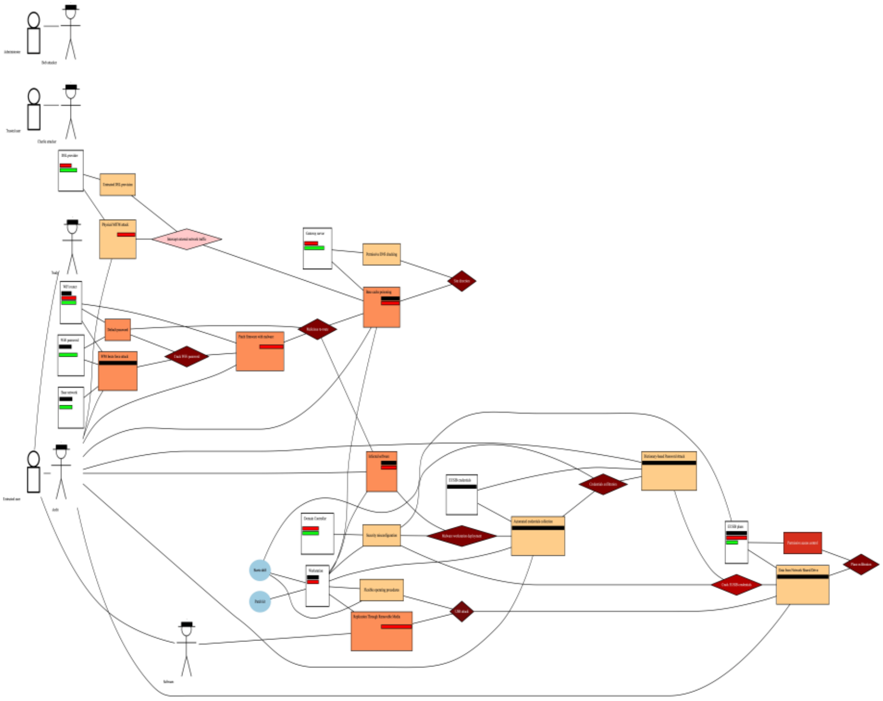


Fig. 5. Complete Risk model for the storyline.

The next figures and paragraphs describe the UC01 “Wi-Fi router firmware attack” risks’ modelling. The other two use cases’ cyber security risks were modelled using the same approach.

In the first step, corresponding with the risk ‘Crack Wi-Fi password’(Fig. 6). The threat described corresponds with a WPA brute force attack facilitated by the VaporMonger toolkit, and the vulnerability is the use of default passwords. Together, these form the basis of the ‘Crack Wi-Fi password’ risk, made possible by Ardit getting closer enough to the base perimeter to obtain the base Wi-Fi signal. On the basis of the brute force attack, the router password can be obtained in a short period of time.

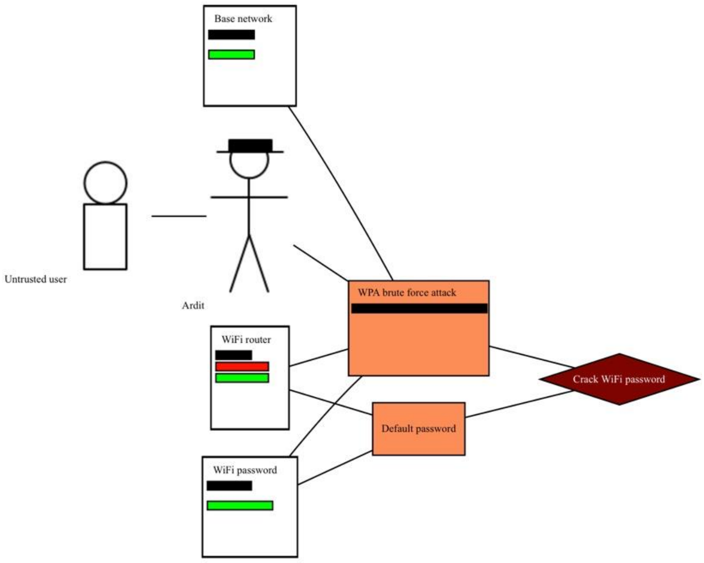


Fig. 6. Crack Wifi password risk.

The second step corresponds with the malicious re-route risk (Fig. 7). This is facilitated by the attacker obtaining the Wi-Fi password via the Crack Wi-Fi password risk and, using the VaporMonger toolkit, patching the firmware to poison the router’s DNS cache. Like the previous risk, this attack is possible due to the use of a default password for the Wi-Fi router, which we assume would be known to the VaporMonger toolkit.

The third step corresponds with the Site redirection risk (Fig. 8). The Malicious re-route risk facilitates the Base cache poisoning threat, where the malicious re-routing ensures MasseHook network requests are redirected to a fake site that collects credentials before re-directing traffic to the authentic site. This is made possible by permissive checking of DNS tracking.

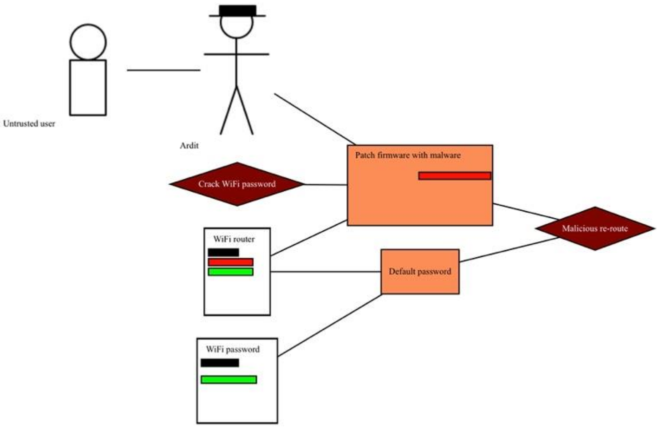


Fig. 7. Malicious re-route risk.

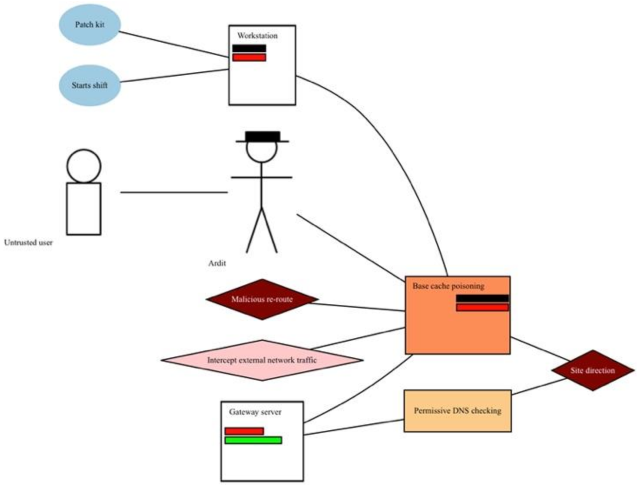


Fig. 8. Site re-direction risk.

7. Conclusion

In this paper, we presented the approach used by ECHO for devising use cases for subsequent analysis and validation using CAIRIS. Our approach helped elicit sector-specific cybersecurity knowledge available within the ECHO consortium, identify the user needs within the different industry sectors.

The benefit of our approach for use case definitions was the subsequent specification of user requirements that the ECHO outputs should address. Relevant use cases description provides some assurance for the development of the project assets, and useful guidelines for the validation of the solutions developed.

Our approach has a wider application than only requirements elicitation. The use cases could and should be used to drive the development process. The research has proved that CAIRIS is an effective platform for enriching the context of threat models. We are currently using CAIRIS to apply security-by-design principles to the design of ECHO Early Warning System and ECHO Federated Cyber Range. Our experiences designing these systems will be the subject of future work.

Acknowledgements

This work has been funded by the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no. 830943, ECHO (European network of Cybersecurity centres and competence Hub for innovation and Operations) project.
The authors are grateful to all partners who contributed to the deliverable D2.1.

References

1. European Network of Cybersecurity Centres and Competence Hub for Innovation and Operations (ECHO), https://echonetwork.eu/ last accessed 2021/04/12
2. E. Nasr, J. McDermid and G. Bernat, Eliciting and specifying requirements with use cases for embedded systems, In Proceedings of the Seventh IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, San Diego, CA, USA, pp. 350-357, (2002)
3. A technical discussion on Use Case Best Practices, 06/11/03, IBM, https://www.eg.bucknell.edu/~cs475/F04-S05/useCases.pdf, last accessed 2021/03/10
4. Computer Aided Integration of Requirements and Information Security. (CAIRIS) web site, https://cairis.org, last accessed 2021/03/11
5. Faily, S. Designing Usable and Secure Software with IRIS and CAIRIS, 1st edn., Springer, 2018
6. https://attack.mitre.org/, last accessed 2021/03/11
7. A KAOS Tutorial, http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf, last accessed 2021/03/11
8. CAIRIS API, https://app.swaggerhub.com/apis/failys/CAIRIS, last accessed 2021/03/11
9. CAIRIS documentation, https://docs.cairis.org, last accessed 2021/03/11
10. https://www.bleepingcomputer.com/news/security/cia-created-toolkit-for-hacking-hundreds-of-routers-models/, last accessed 2021/04/12
11. https://en.wikipedia.org/wiki/Vault_7, last accessed 2021/04/12
12. Faily, S., Lyle J, Fléchais I, Simpson A, Usability and Security by Design: A Case Study in Research and Development, USEC ’15, San Diego, CA, USA, Copyright 2015 Internet Society, ISBN 1-891562-40-1, http://dx.doi.org/10.14722/usec.2015.23012
13. Faily S., Scandariato R., Shostack A., Sion L., Ki-Aries D. (2020) Contextualisation of Data Flow Diagrams for Security Analysis. In: Eades III H., Gadyatskaya O. (eds) Graphical Models for Security. GraMSec 2020. Lecture Notes in Computer Science, vol 12419. Springer, Cham. https://doi.org/10.1007/978-3-030-62230-5_10