Appendix A. WEA Case Study: Evaluating Security Risks Using Mission Threads[1]

1. Originally published in *CrossTalk* September/October 2014.

by Carol Woody, PhD, and Christopher Alberts

Importance of Systems of Systems

Everything we do these days involves system and software technology: Cars, planes, banks, restaurants, stores, telephones, appliances, and entertainment rely extensively on technology. Much of this capability is supported by systems of systems—independent heterogeneous systems that work together to address desired functionality through complex network, data, and software interactions. The Wireless Emergency Alerts (WEA) service is a good example of a system of systems.

WEA enables local, tribal, state, territorial, and federal public safety officials to send geographically targeted text alerts to the public. The U.S. Department of Homeland Security Science and Technology (DHS S&T) Directorate partners with the Federal Emergency Management Agency (FEMA), the Federal Communication Commission (FCC), and commercial mobile service providers (CMSPs) to enhance public safety through the deployment of WEA, which permits emergency management organizations nationwide to submit alerts for public distribution by mobile carriers [FEMA 2015]. Alert originators can send three types of messages:

- Presidential alerts, issued by the president of the United States to reach any region of the nation or the nation as a whole
- Imminent threat alerts
- AMBER (America's Missing: Broadcast Emergency Response) alerts

CMSPs relay these alerts from FEMA's Integrated Public Alert and Warning System (IPAWS) to mobile phones using cell broadcast technology, which does not get backlogged during times of emergency, unlike wireless voice and data services. Customers who own WEA-capable mobile phones automatically receive these alerts during an emergency if they are located in the affected geographic area.

Alert originators already have extensive alert dissemination capability through the Emergency Alert System, highway signage systems, Internet websites, and telephone dialing systems, just to mention a few widely used alerting channels. The WEA system, shown in Figure A.1, expands these options to mobile devices. FEMA established the message structure along with the approvals needed to have the Alert Aggregator system disseminate messages to mobile devices. Many alert originators plan to integrate this capability with systems already in place for other dissemination channels.

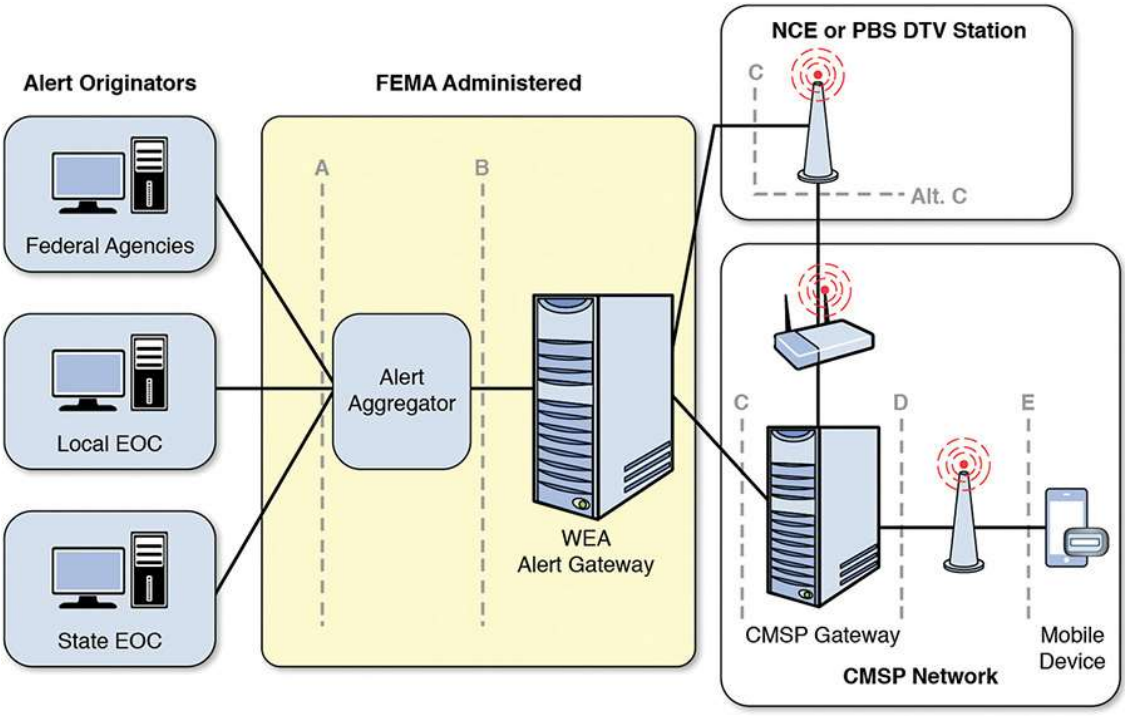

Figure A.1 *WEA System of Systems*

The *Systems Engineering Handbook* describes the following challenges for the development (and sustainment) of systems of systems [Haskins 2010]:

- Each participating system operates independently.

- Each participating system has its own update, enhancement, and replacement cycle.

- Overall performance of the desired functionality depends on how the various participating systems can interact, which is not always known in advance.

- Missing or conflicting standards can make the design of data exchanges among the participating systems complex and difficult to sustain.

- Each participating system has its own management, and the coordination of requirements, budget constraints, schedules, interfaces, and upgrades can have a major impact on the expected capability of the system of systems.

- Fuzzy boundaries can cause confusion and error; no one really owns the interface, but one of the participants needs to take leadership to ensure some level of shared understanding.

- The system of systems is never finished because as each system grows, expands, and ages, there is a constant need for adjustment.

Public safety officials and alert recipients want to be able to rely on WEA capabilities and need to have confidence that the alerts are accurate and timely. Effective security is required to support this confidence. The risk that an attacker could create false alerts or cause valid alerts to be delayed, destroyed, or modified is a critical issue. Such actions could place the alert-originating organization's mission—and the lives and property of the citizens it serves—at risk.

DHS S&T asked a team of security experts at the Carnegie Mellon University's Software Engineering Institute (SEI) to research this problem and identify a means for evaluating WEA alert originator security concerns. The team selected mission thread analysis as a means for developing a view of the system of systems that could be used for evaluating security risks.

An analysis approach was needed to prepare alert originators to address the following critical security questions [Allen 2008]:

- What do alert originators need to protect? Why does it need to be protected? What happens if it is not protected?

- What potential adverse consequences do alert originators need to prevent? At what cost? How much disruption can they stand before they take action?

- How do alert originators determine and effectively manage the residual risk?

In addition, alert originators needed to consider the local, state, and federal compliance standards that the organization must address to ensure that the planned choices for security also meet other mandated standards.

Preparing for Mission Thread Analysis

Drawing on SEI security expertise, initial questions were assembled to assist the alert originator in gathering information about the current environment and preparing for WEA (or any new technology capability).

The alert-originating organization should compose answers to the following questions:

- What WEA capability do we plan to implement (types of alerts to issue, geographic regions to cover)?

- Can we expand existing capabilities to add WEA, or do we need new capabilities?

- Are good security practices in place for the current operational environment? Is there any history of security problems that can inform our planning?

- Will we use current resources (technology and people), or do we need to add resources?

Responses to these questions begin to frame the target operational context and the critical functionality that organizations must evaluate for operational security. Each organization has a different mix of acquired technology and services, in-house development components, and existing operational capability into which the WEA capability will be woven. With the use of mission threads, responses to these questions can be described in a visually compelling form that management, system architects, system and software engineers, and stakeholders can share and refine.

A mission thread is an end-to-end set of steps that illustrate the technology and people resources needed to deliver expected behavior under a set of conditions and provide a basis for identifying and analyzing potential problems that could represent risks. For each mission step, the expected actions, outcomes, and assets are assembled. Confirmation that the components appropriately respond to expected operational use increases confidence that the system will function as intended, even in the event of an attack [Ellison 2008].

Mission threads provide a means to identify and evaluate the ways, intentional or unintentional, that component system failures could occur and how such failures would impact the mission. Next, a WEA example is provided to demonstrate how the SEI used a mission thread to analyze security.

WEA Mission Thread Example

Mission thread analysis begins with the development of an operational mission thread. For WEA, typically 25 steps take place from the determination of the need for an alert to the receipt of that alert by cell phone owners:

1. First responder contacts local alerting authority via an approved device (cell phone, email, radio, etc.) to state that an event meets criteria for us-

ing WEA to issue, cancel, or update an alert and provides information for message.

2. Local alerting authority (person) determines that the call or email from the first responder is legitimate.

3. Local alerting authority instructs Alert Origination System (AOS) operator to issue, cancel, or update an alert using information provided by first responder.[2]

---

2. In some cases, the alerting authority and the AOS operator may be the same person.

4. AOS operator logs on to the AOS.

5. AOS logon process activates auditing of the operator's session.

6. AOS operator enters alert, cancel, or update message.

7. AOS converts message to a format compliant with the Common Alerting Protocol (CAP, a WEA input standard).

8. CAP-compliant message is signed by a second person for local confirmation.

9. AOS transmits message to the IPAWS Open Platform for Emergency Networks (OPEN) Gateway.

10. IPAWS-OPEN Gateway verifies[3] message and returns status message to AOS.

---

3. In this list of steps, message verification includes authentication and ensuring that the message is correctly formatted.

11. AOS operator reads status message and responds as needed.

12. If the message was verified, IPAWS-OPEN Gateway sends message to WEA Alert Aggregator.

13. WEA Alert Aggregator verifies message and returns status to IPAWS-OPEN Gateway.

14. IPAWS-OPEN Gateway processes status and responds as needed.

15. WEA Alert Aggregator performs additional message processing as needed.

16. If the message was verified, WEA Alert Aggregator transmits alert to Federal Alert Gateway.

17. Federal Alert Gateway verifies message and returns status to WEA Alert Aggregator.

18. WEA Alert Aggregator processes status and responds as needed.

19. If the message was verified, Federal Alert Gateway converts message to CMAC (Commercial Mobile Alert for Interface C) format.

20. Federal Alert Gateway transmits message to CMSP gateway.

21. CMSP Gateway returns status to Federal Alert Gateway.

22. Federal Alert Gateway processes status and responds as needed.

23. CMSP Gateway sends message to CMSP Infrastructure.

24. CMSP Infrastructure sends message via broadcast to mobile devices in the designated area(s).

25. Mobile device users (recipients) receive the message.

Although many of these steps do not involve technology, they can still represent security risks to the mission. Mission thread analysis, unlike other techniques such as Failure Mode and Effect Analysis [Stamatis

2003], allows consideration of the people and their interactions with technology in addition to the functioning of a system itself. Also, most security evaluations consider only individual system execution. However, effective operational execution of a mission must cross organizational and system boundaries to be complete. The use of mission thread analysis for security provides a way to confirm that each participating system is secure and does not represent a risk to all others involved in mission execution.

Figure A.2 provides a picture of the WEA mission thread and includes step numbers from the list to link each step to the appropriate system area. Successful completion requires flawless execution of four major system areas—alert originator, FEMA IPAWS system, CMSPs, and cell phone recipients—each shown in a row of the figure. Each area operates independently, and they are connected only through the transmission of an alert.

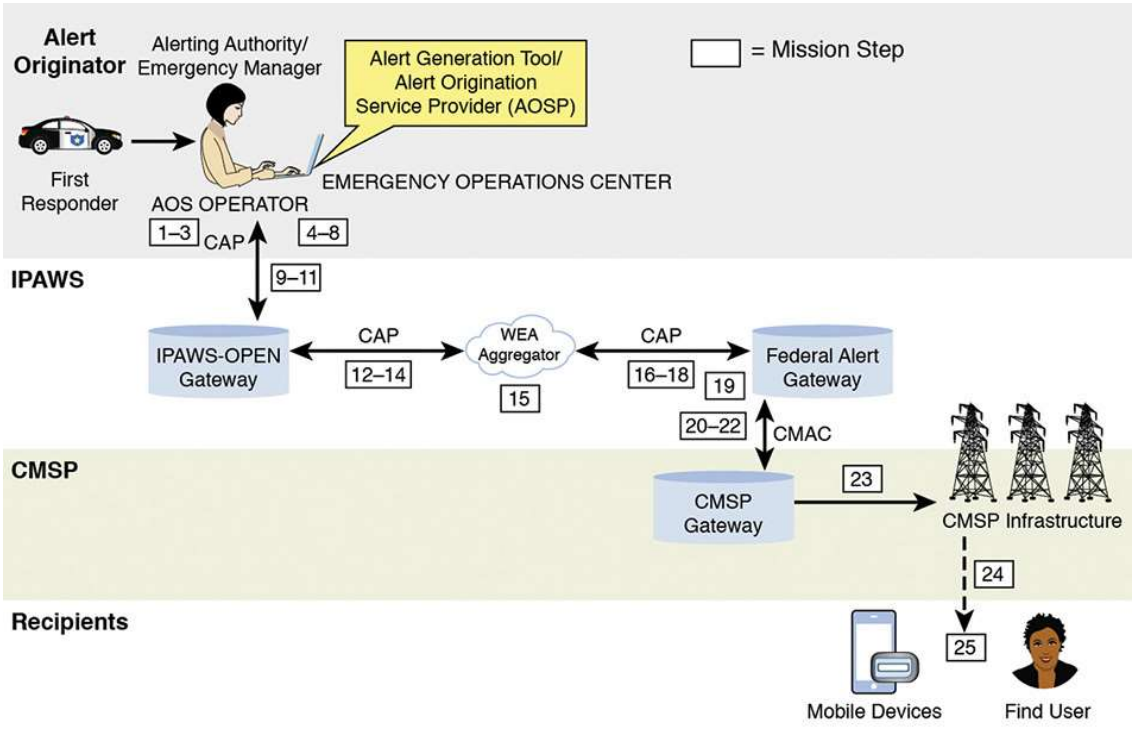

Figure A.2 *WEA Mission Thread Diagram*

WEA Security Analysis

Using the mission thread illustrated in Figure A.2, potential security concerns can be identified through possible security threats. For the WEA example, the SEI selected the STRIDE threat method for threat evaluation. STRIDE, developed by Microsoft, considers six typical categories of security concerns: spoofing, tampering with data, repudiations, information disclosure, denial of service, and elevation of privilege [Microsoft 2013]. The name of this threat method is derived from the first letter of each security concern [Howard 2006]. As an illustration of how STRIDE can be applied, focus on steps 4–9 of the mission thread, which represent the transition across two major system areas from the alert originator to the FEMA system and provide an opportunity for mission failure if interaction between the system areas is not secure. Table A.1 shows the result of the STRIDE analysis on the selected steps.

Step	Assets	STRIDE Threat Identification[a] Examples
4. AOS operator attempts to log on to the alert origination system.	• One person • Server (valid accounts/authentication information) • Logon procedure • Logon application • Username/password data in database • Communications between logon software, server, and AOS	S: Unidentified individual attempts to log on with AOS operator's information R: AOS operator denies having logged on I: Capture of logon info using key logger or packet sniffer D: AOS operator's account not registered or servers are down E: Successful logon by an unidentified and unauthorized individual
5. AOS logon activates auditing of the operator's session.	• Auditing application • Auditing procedure • Communications from accounts to auditing application • Local or remote storage	T: Logged entries added, deleted, or modified inappropriately I: Logged entries containing credential data are compromised D: Log full or server unavailable
6. AOS operator enters alert, cancel, or update message.	• One person • Alert scripts • Procedures for building scripts • GUI application • Communications between GUI application and alert generation software (including server and application)	T: Formatting errors produce incorrect message D: Scripts are unavailable or corrupted
7. AOS converts message to CAP-compliant format required by IPAWS.	• Conversion application	T: Data are changed between the AOS and the server D: Server is down
8. CAP-compliant message is signed by two people.	• Signature entry application • Signature validation application • Public/private key pair for every user	S: Digital signature is falsified R: User claims not to have signed D: Server goes down so keys cannot be distributed, or keys have expired and message cannot be sent
9. AOS transmits message to the IPAWS-OPEN Gateway.	• Application that securely connects to IPAWS • Information used to authenticate AOS and IPAWS	S: Falsified AOS CAP message or IPAWS gateway attacked and site is redirected T: Data within message are modified I: Message is not encrypted and credentials are visible D: IPAWS-OPEN Gateway is down

[a] *S:* spoofing; T: tampering with data; R: repudiation; I: information disclosure; D: denial of service; E: elevation of privilege.

Table A.1 *STRIDE Analysis for Selected WEA Mission Thread Steps*

For each step, the team analyzed technology assets critical to step execution to determine ways that STRIDE threats can compromise each asset used in that step [Howard 2006]. Security and software experts as well as individuals familiar with the operational mission must participate in this portion of the analysis. The security and software experts have an understanding of what can go wrong and the potential impact of each possible failure on the analysis. Those knowledgeable about the operational execution can ensure that the scenarios are realistic and valid. Available documentation can provide a start for the development of the mission threads, but there is a tendency to document the desired operational environment and not the real one. Effective security risk analysis requires access to realistic operational information.

Based on this input, security experts (individuals with operational security training and experience) identified at least two security risks that could lead to mission failure:

- Authentication of the individual using the AOS in step 4
- Validation and protection of the digital signatures applied to the alert approved for submission to the Alert Aggregator in step 8

To analyze these risks in greater detail and help alert originators understand how a security risk could materialize, mission threads for each specific risk were assembled. Figure A.3 provides a picture of the risk scenario that describes the second security risk (validity of the digital signature) noted from the analysis of the WEA operational mission thread. The following paragraphs provide a dialogue that describes ways in which the security threat could materialize and why the alert origination organization should consider possible mitigations.

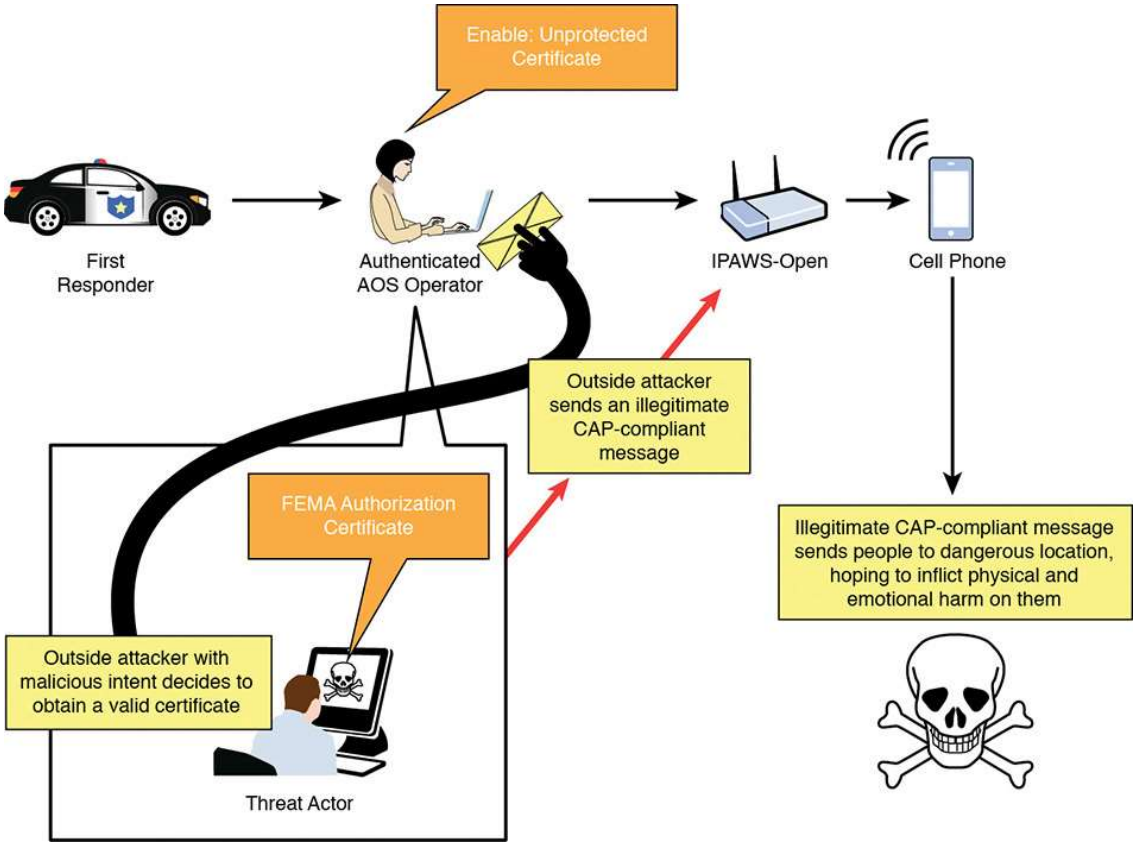

Figure A.3 *Security Risk Scenario*

An outside attacker with malicious intent decides to obtain a valid certificate and use it to send an illegitimate CAP-compliant message. The

attacker's goal is to send people to a dangerous location, hoping to inflict physical and emotional harm on them. The key to this attack is capturing a valid certificate from an alert originator. The attacker develops two strategies for capturing a valid certificate. The first strategy targets an alert originator directly. The second strategy focuses on AOS vendors. Targeting a vendor could be a particularly fruitful strategy for the attacker. The number of vendors that provide AOS software is small. As a result, each vendor controls a large number of certificates. A compromised vendor could provide an attacker with many potential organizations to target.

No matter which strategy is pursued, the attacker looks for vulnerabilities (i.e., weaknesses) in technologies or procedures that can be exploited. For example, the attacker tries to find vulnerabilities that expose certificates to exploit, such as the following:

- Unmonitored access to certificates
- Lack of encryption controls for certificates during transit and storage
- Lack of role-based access to certificates

The attacker might also explore social engineering techniques to obtain a certificate. Here, the attacker attempts to manipulate someone from the alert originator or vendor organization into providing access to a legitimate certificate or to get information that will be useful in the attacker's quest to get a certificate.

Obtaining a certificate is not a simple endeavor. The attacker must be sufficiently motivated and skilled to achieve this interim goal. However, once this part of the scenario is complete, the attacker is well positioned to send an illegitimate CAP-compliant message. The attacker has easy access to publicly documented information defining how to construct CAP-compliant messages.

The attacker's goal in this risk is to send people to a location that will put them in harm's way. To maximize the impact, the attacker takes ad-

vantage of an impending event (e.g., weather event, natural disaster). Because people tend to verify WEA messages through other channels, synchronizing the attack with an impending event makes it more likely that people will follow the attacker's instructions. This scenario could produce catastrophic consequences, depending on the severity of the event with which the attack is linked.

Through the use of mission thread analysis, security expertise can be integrated with operational execution to fully describe and analyze operational security risk situations. While there may be many variations of operational execution, an exhaustive study of all options is not necessary. Building a representative example that provides a detailed view of a real operational mission from start to finish has proven to be of value for security risk identification.

Conclusion

The process of developing a well-articulated mission thread that operational and security experts can share and analyze provides an opportunity to uncover missing or incomplete requirements as well as differences in understanding, faulty assumptions, and interactions across system and software boundaries that could contribute to security concerns and potential failure [Ellison 2008].

The mission thread analysis connects each mission step with the technology and human assets needed to execute that step and provides a framework to link potential security threats directly to mission execution. Mission thread diagrams and tables assemble information in a structure that can be readily reviewed and validated by operational and technology experts from various disciplines, including acquisition, development, and operational support. Mission thread security analysis can be an effective tool for improved identification of security risks to increase confidence that the system of systems will function with appropriate operational security.

References

[Allen 2008]

Allen, Julia; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy. *Software Security Engineering: A Guide for Project Managers.* Addison-Wesley. 2008.

[Ellison 2008]

Ellison, Robert J.; Goodenough, John B.; Weinstock, Charles B.; & Woody, Carol. *Survivability Assurance for System of Systems*. CMU/SEI-2008-TR-008. Software Engineering Institute, Carnegie Mellon University. 2008. http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=8693.

[FEMA 2015]

Federal Emergency Management Agency. Wireless Emergency Alerts. *Federal Emergency Management Agency.* June 21, 2015. http://www.fema.gov/wireless-emergency-alerts.

[Haskins 2010]

Haskins, Cecilia, ed. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities,* version 3.2. Revised by Kevin Forsberg, M. Krueger, & R. Hamelin. International Council on Systems Engineering (INCOSE). 2010.

[Howard 2006]

Howard, Michael & Lipner, Steve. *The Security Development Life Cycle.* Microsoft Press. 2006.

[Microsoft 2013]

Microsoft. The STRIDE Threat Model. *Microsoft Developer Network.* 7 June 2013 [accessed]. http://msdn.microsoft.com/en-US/library/ee823878%28v=cs.20%29.aspx.

[Stamatis 2003]

Stamatis, D. H. Failure Mode and Effect Analysis: FMEA from Theory to Execution. 2nd ed. ASQ Quality Press. 2003.