Breaking Trust

Shades of Crisis Across an Insecure Software Supply Chain

Trey Herr (presenting) + Will Loomis, Stewart Scott, June Lee, and Emma Schroeder

Atlantic Council

Methods & Definitions

139 incidents collected since October 2019 from publicly available information

Software Supply Chain Attack (SSCA): Occurs when attackers access and edit software somewhere in the complex software development supply chain to compromise a target down the chain by inserting their own malicious code.

Software Supply Chain Vulnerability (SSCV): Any software vulnerability that could be employed in a supply chain attack if exploited.

Code Location/Owner: Vendor or OSS repository associated with the codebase.

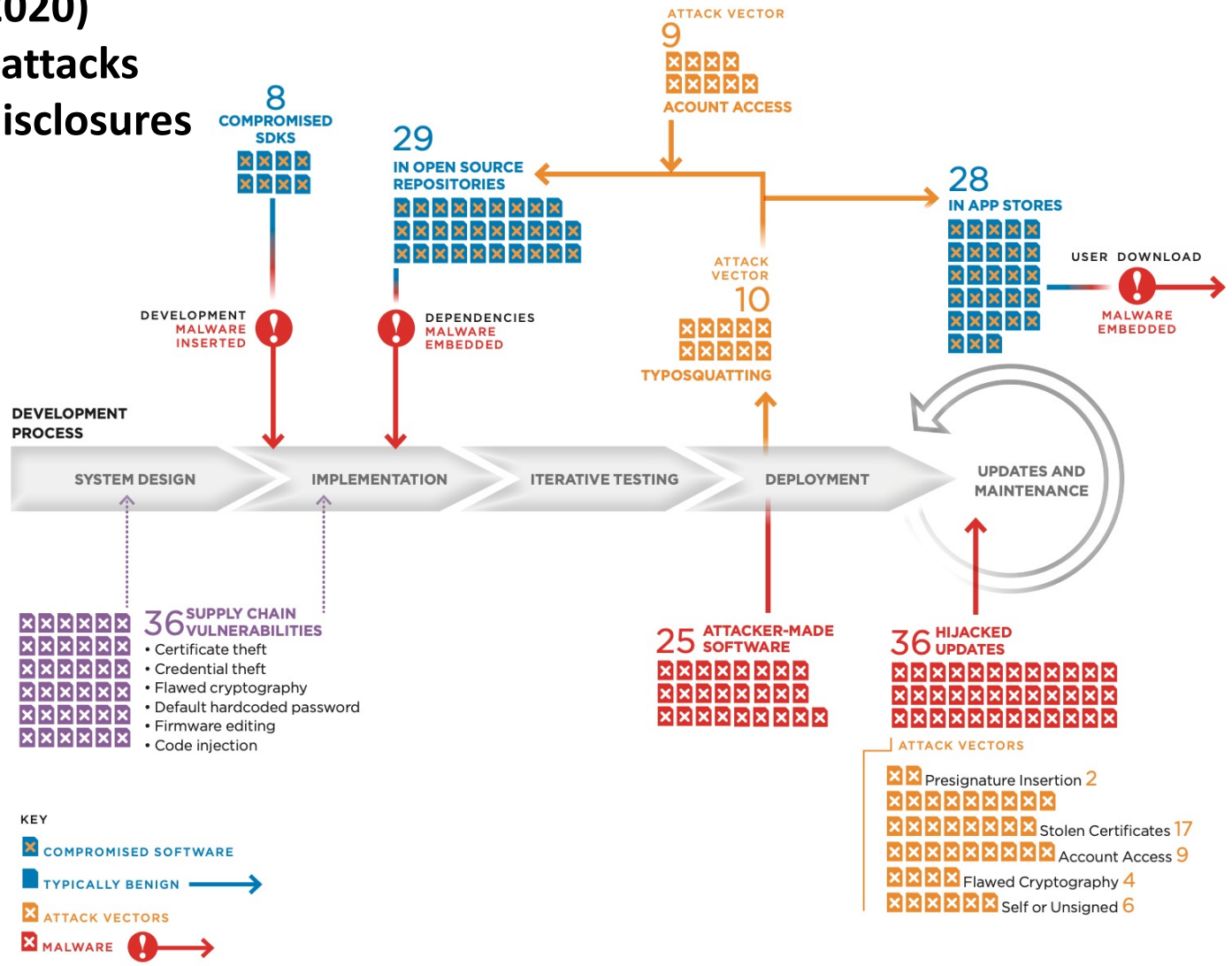
Downstream Target: The end-target of the attack.

Affected Codebase: Categories describing the codebase, product, or service modified by attackers or subject to disclosure

A Decade of Attacks and Disclosures in the Software Supply Chain

(2010-2020)

- **103 attacks**
- **36 disclosures**



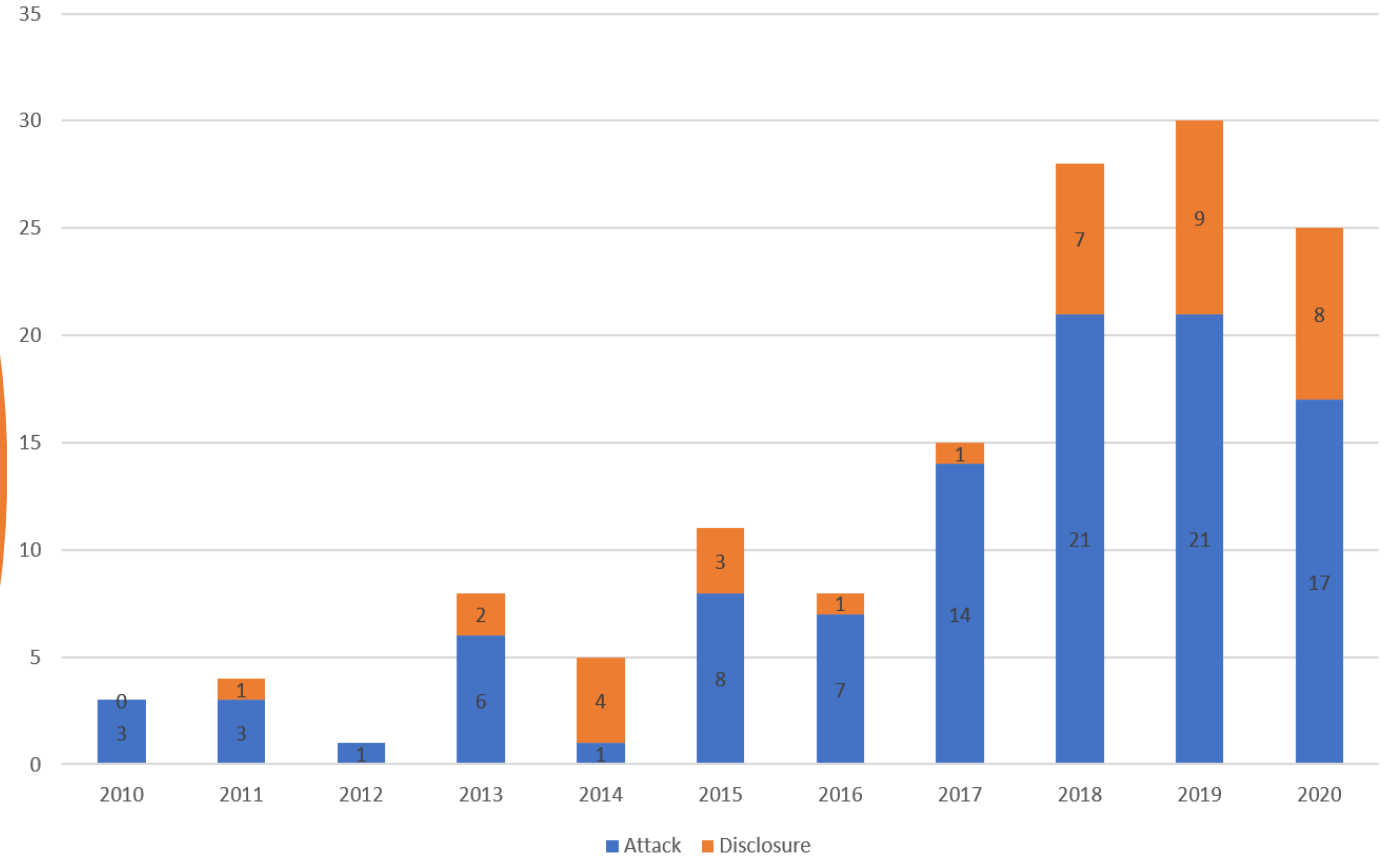
Atlantic Council

For more information, visit: https://www.atlanticcouncil.org/breaking-trust

Takeaways

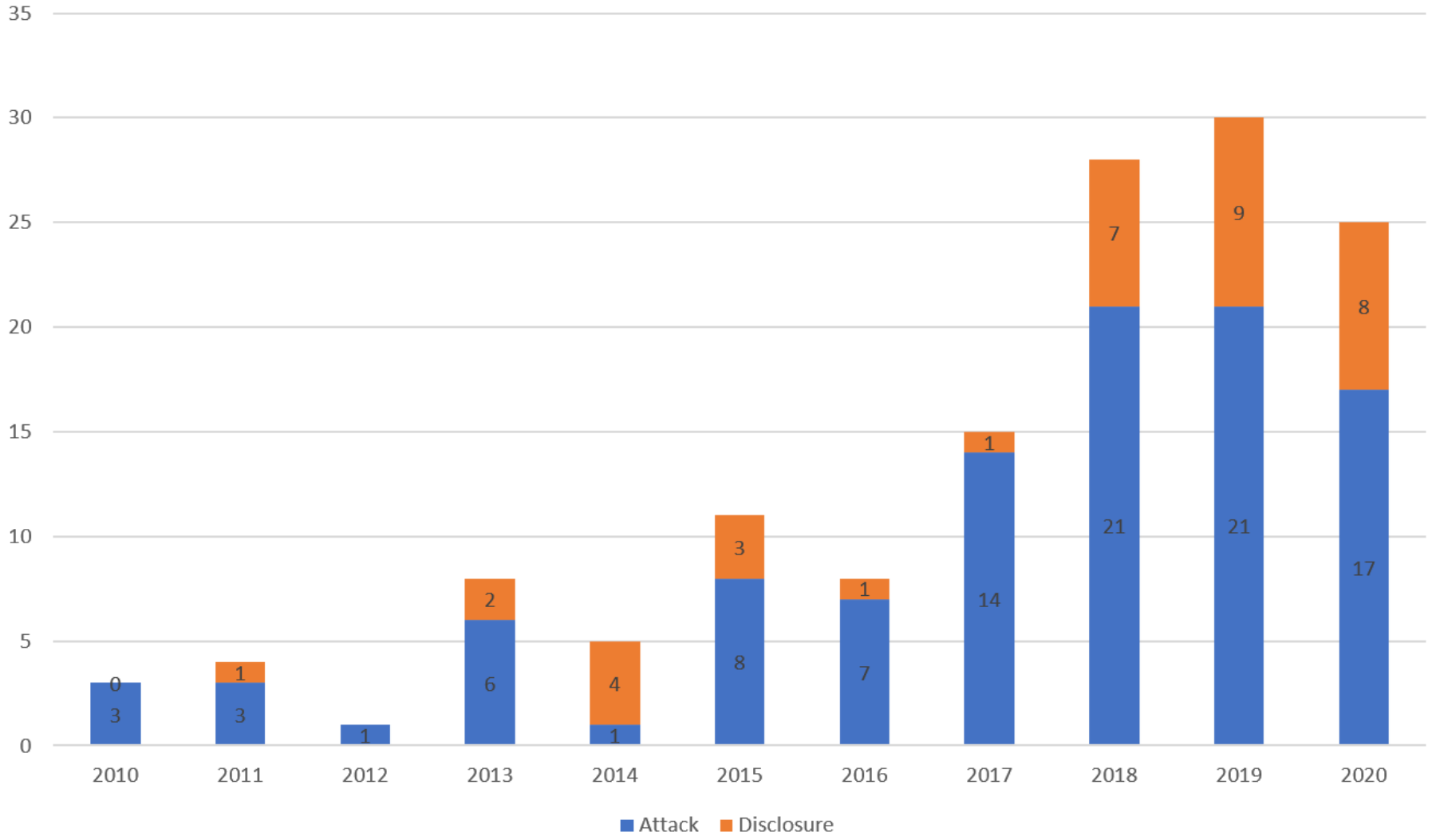
- Attacks have been **popular** and will continue to rise in prominence over time.
- Attacks that utilize software supply chain vulnerabilities are **impactful**.
- **Used to great effect** by states

Figure 1: Attacks and Disclosures by Year



Atlantic Council

Figure 1: Attacks and Disclosures by Year



Atlantic Council

Trends

- At least 30 attacks from **state actors** especially China & Russia
- Attacks undermined signing certificates & abused **public key cryptography**
- Attacks targeted **widely popular open-source projects**
- 25% of the total incidents targeted **app stores** & **developer tools**
- 26% of incidents targeted **software updates**

Figure 10: Distribution Vector by Year

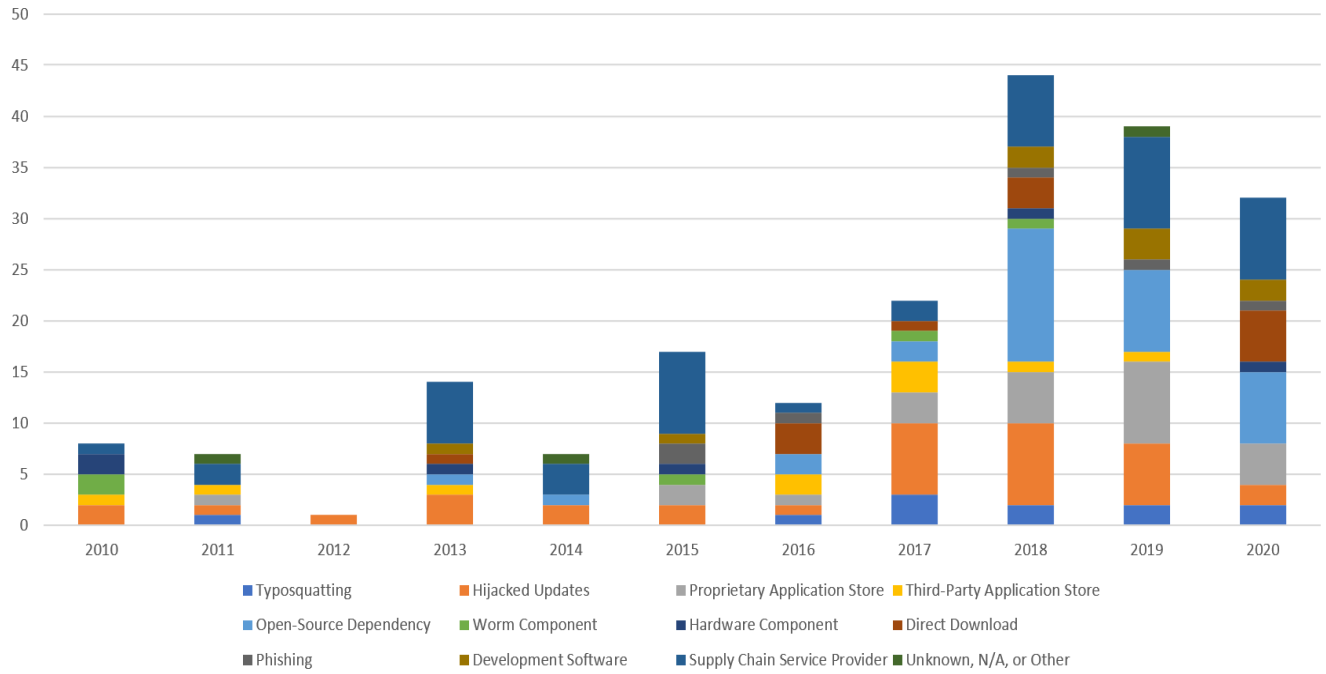
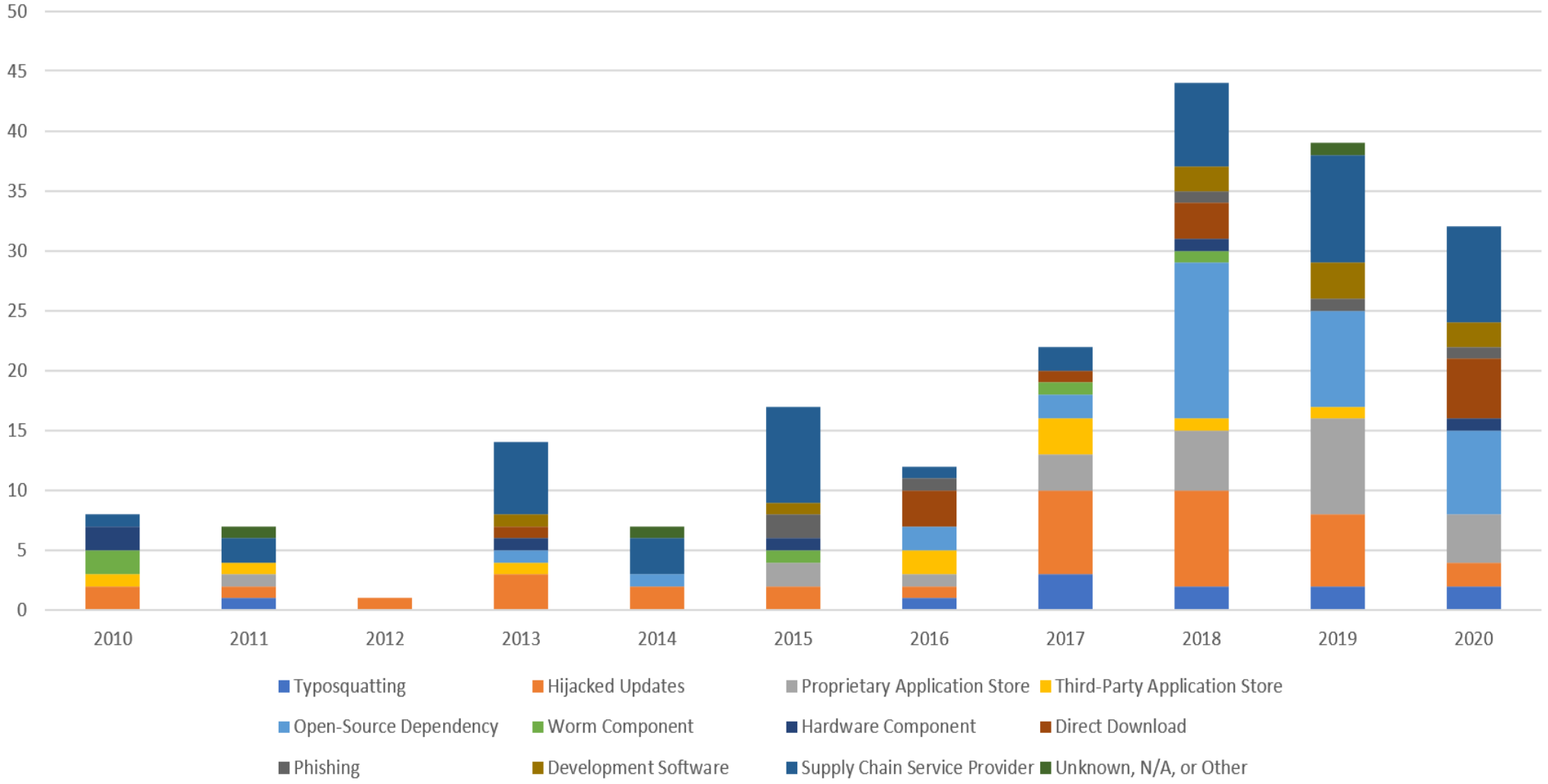
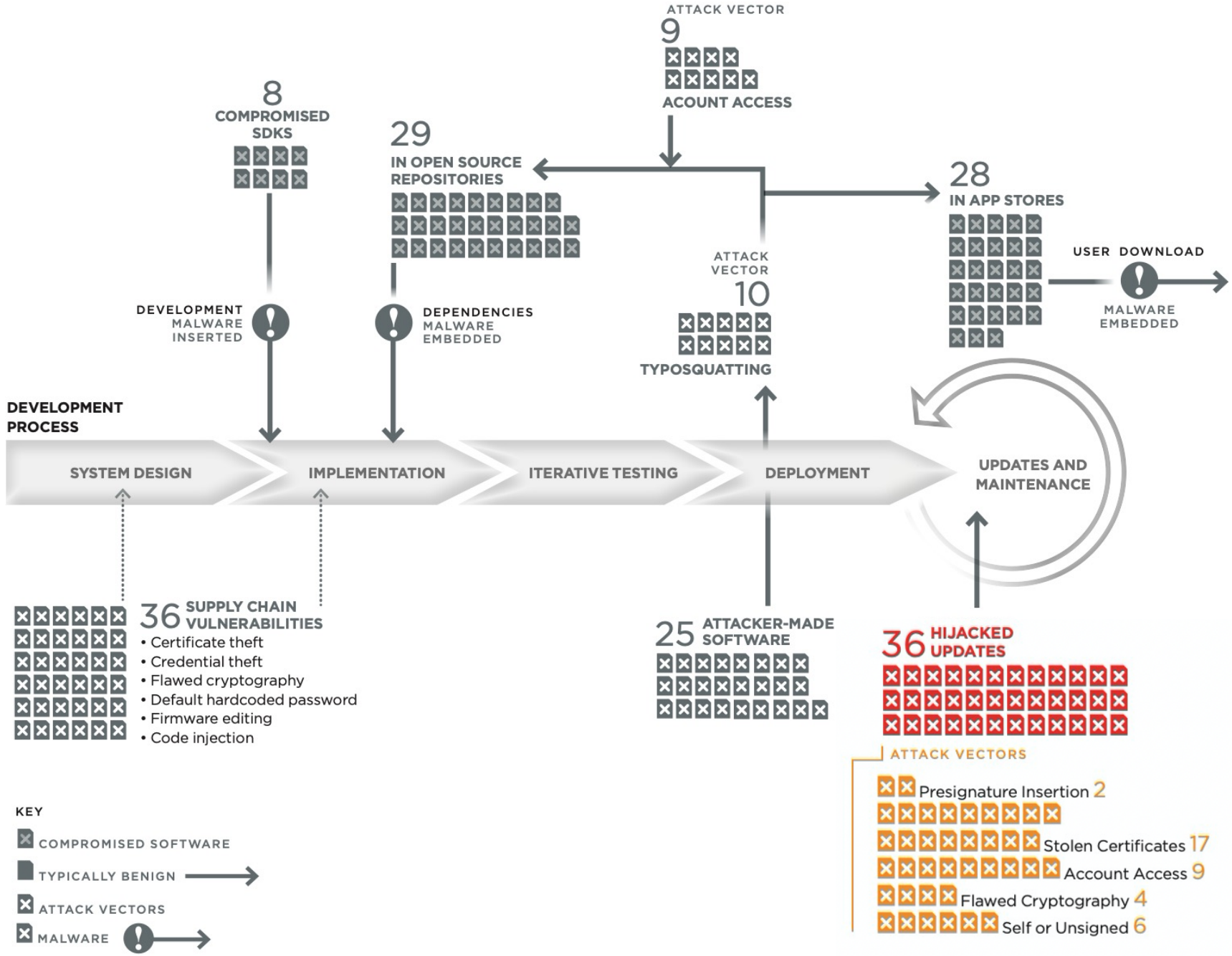
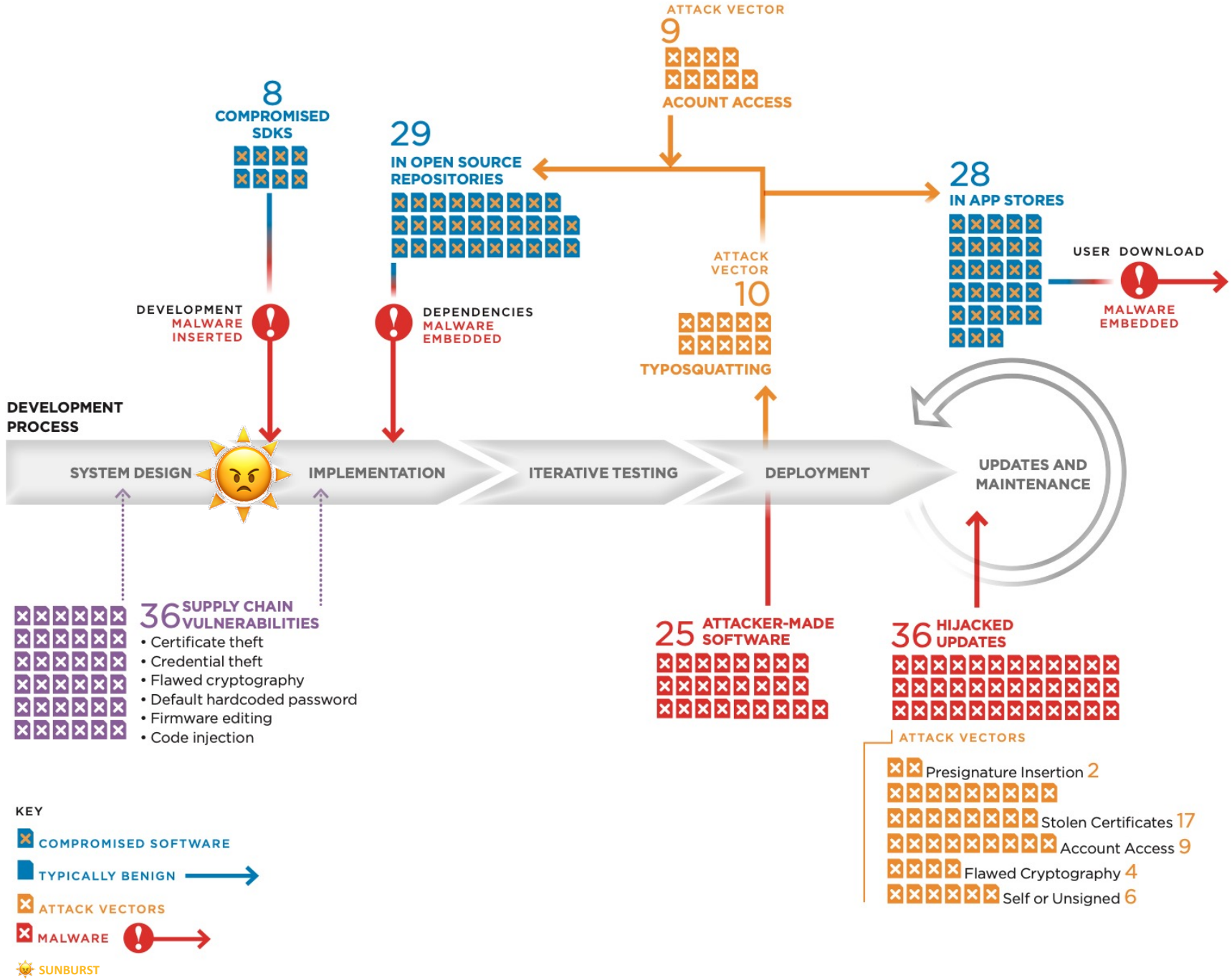


Figure 10: Distribution Vector by Year

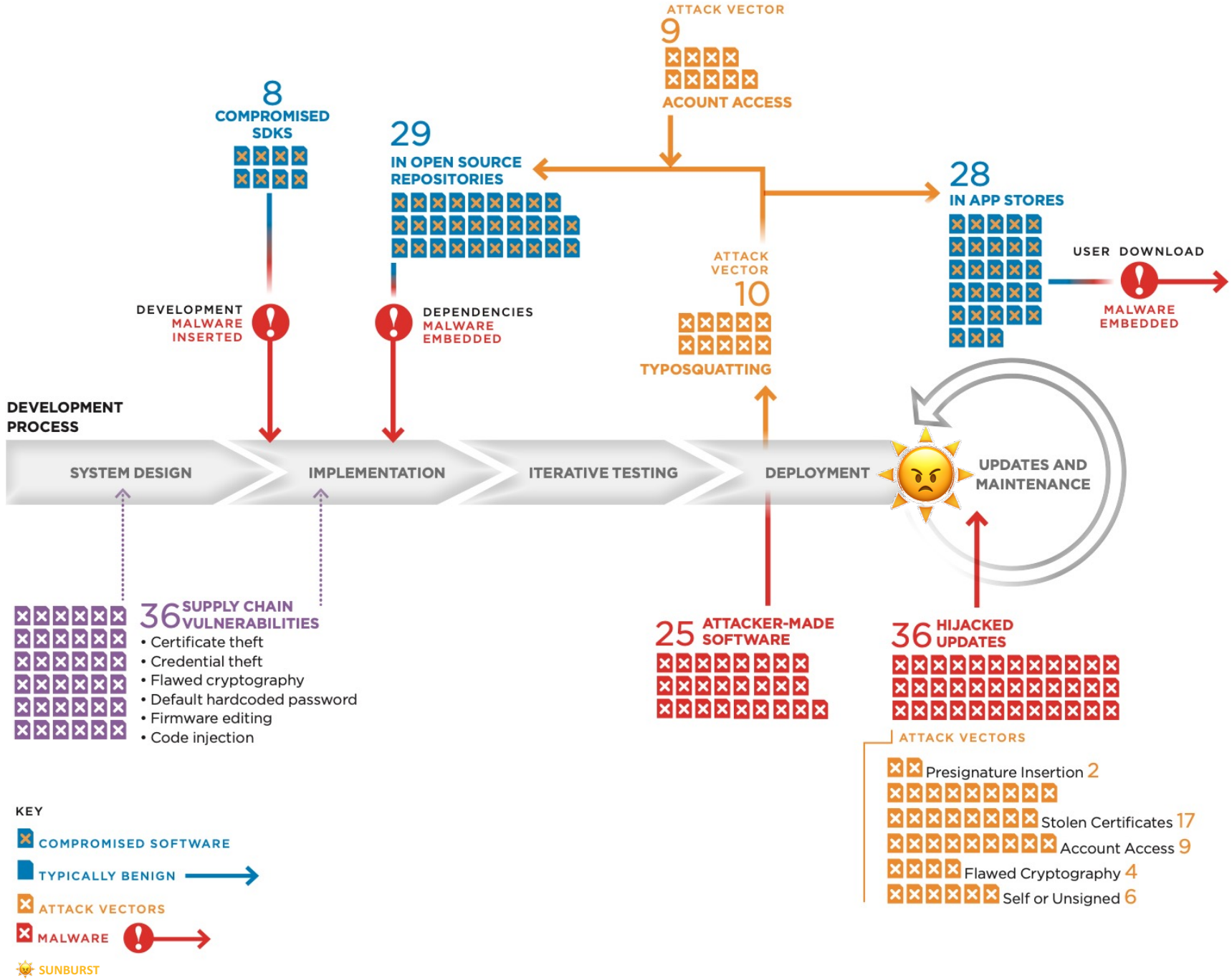




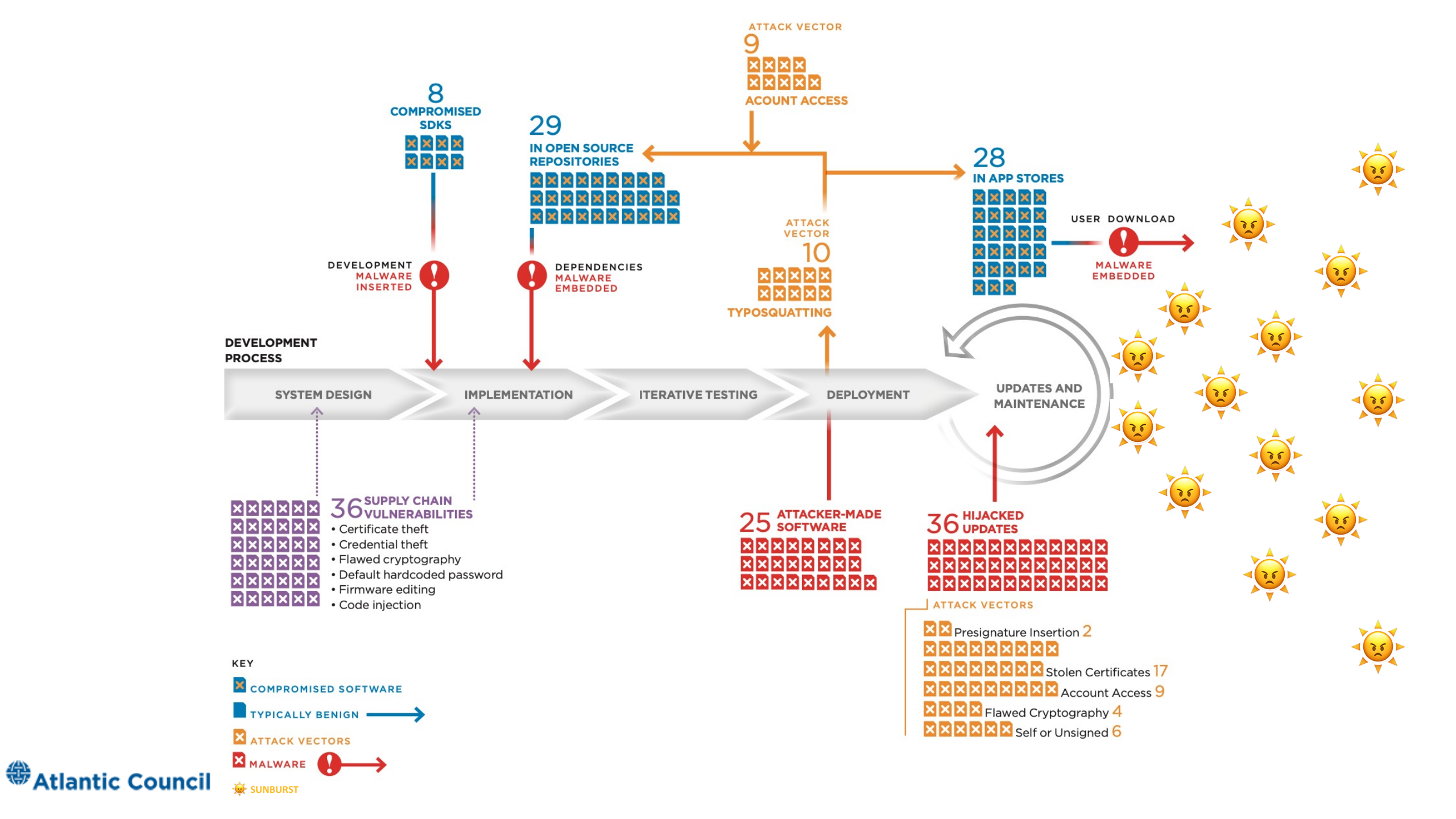
Atlantic Council



Atlantic Council

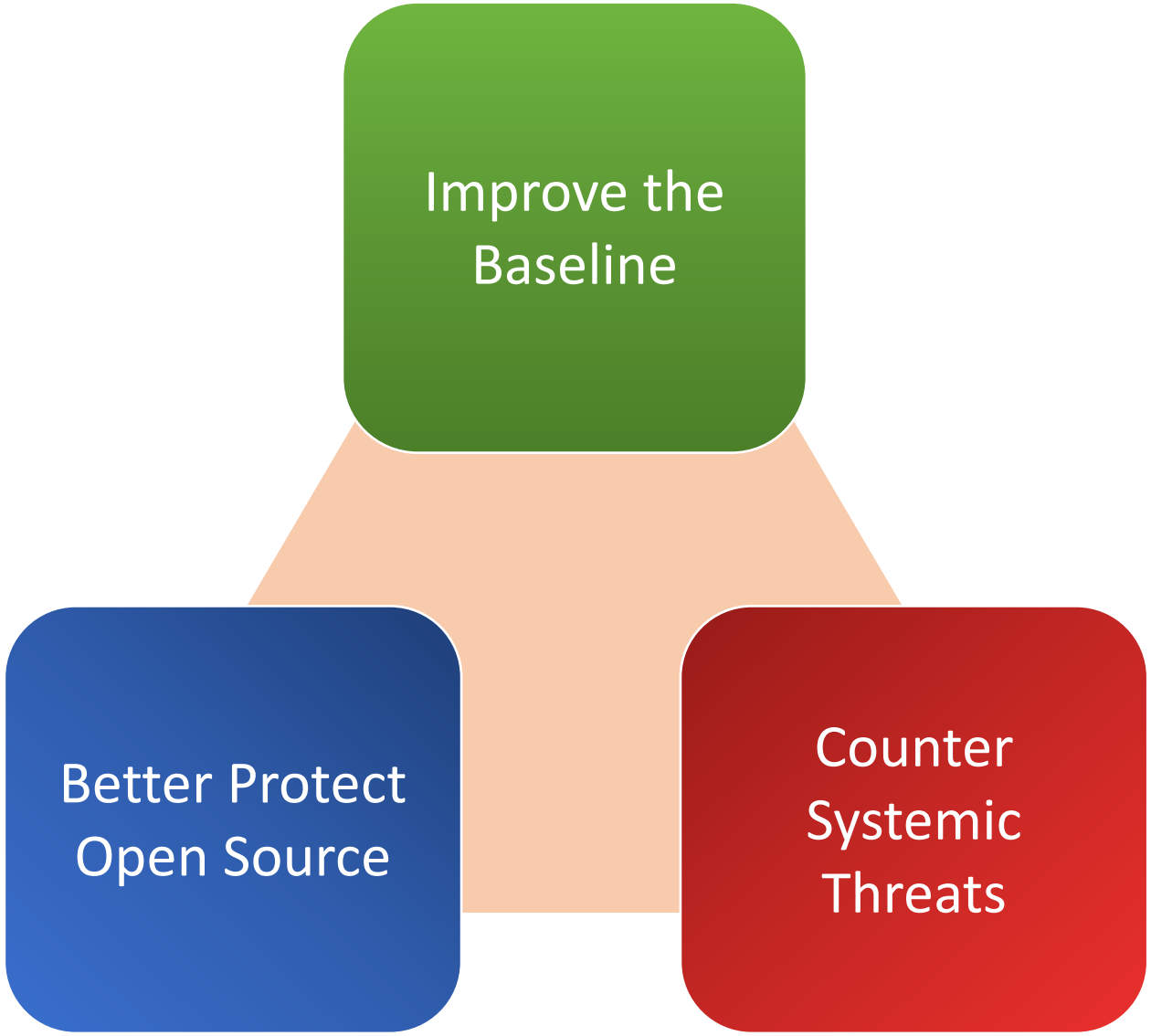


Atlantic Council



Atlantic Council

Recommendations



Thank You

therr@atlanticcouncil.org

For more on this project visit: https://www.atlanticcouncil.org/breaking-trust/

Near Term Recommendations

First

1. Develop a **Lifecycle Security Overlay** to reduce complexity of implementing NIST's SP 800-53 (*NIST, Industry)*

Next

Near Term Recommendations

First

1. Develop a **Lifecycle Security Overlay** to reduce complexity of implementing NIST's SP 800-53 (*NIST, Industry)*
2. Offer public **reference implementations** of the Overlay *(Industry, cloud service providers)*

Next

Near Term Recommendations

First

1. Develop a **Lifecycle Security Overlay** to reduce complexity of implementing NIST's SP 800-53 (*NIST, Industry)*
2. Offer public **reference implementations** of the Overlay *(Industry, cloud service providers)*

7. Provide funding (~$25M) for **baseline security improvements** to open source software (*DHS CISA)*

Next

Near Term Recommendations

First

1. Develop a **Lifecycle Security Overlay** to reduce complexity of implementing NIST's SP 800-53 (*NIST, Industry)*

2. Offer public **reference implementations** of the Overlay *(Industry, cloud service providers)*

7. Provide funding (~$25M) for **baseline security improvements** to open source software (*DHS CISA)*

13. New int'l partners **to investigate & prevent** catastrophic software supply chain attacks (*CISA, NSA, DoJ, NCSC)*

Next

Near Term Recommendations

First

1. Develop a **Lifecycle Security Overlay** to reduce complexity of implementing NIST's SP 800-53 (*NIST, Industry)*

2. Offer public **reference implementations** of the Overlay *(Industry, cloud service providers)*

7. Provide funding (~$25M) for **baseline security improvements** to open source software (*DHS CISA)*

13. New int'l partners **to investigate & prevent** catastrophic software supply chain attacks (*CISA, NSA, DoJ, NCSC)*

14. **Support strong cryptography** in standards & tools *(USG & allies)*

Next

Near Term Recommendations

First

1. Develop a **Lifecycle Security Overlay** to reduce complexity of implementing NIST's SP 800-53 (*NIST, Industry)*

2. Offer public **reference implementations** of the Overlay *(Industry, cloud service providers)*

7. Provide funding (~$25M) for **baseline security improvements** to open source software (*DHS CISA)*

13. New int'l partners **to investigate & prevent** catastrophic software supply chain attacks (*CISA, NSA, DoJ, NCSC)*

14. **Support strong cryptography** in standards & tools *(USG & allies)*

Next

10. Establish **Transatlantic Infrastructure Initiative** with European allies to resource improved security for critical OSS packages (*CISA, State*)

Near Term Recommendations

First

1. Develop a **Lifecycle Security Overlay** to reduce complexity of implementing NIST's SP 800-53 (*NIST, Industry)*

2. Offer public **reference implementations** of the Overlay *(Industry, cloud service providers)*

7. Provide funding (~$25M) for **baseline security improvements** to open source software (*DHS CISA)*

13. New int'l partners **to investigate & prevent** catastrophic software supply chain attacks (*CISA, NSA, DoJ, NCSC)*

14. **Support strong cryptography** in standards & tools *(USG & allies)*

Next

10. Establish **Transatlantic Infrastructure Initiative** with European allies to resource improved security for critical OSS packages (*CISA, State*)

15. **Annual Survey** software supply chain attacks that undermine updates, code integrity, or open-source repositories *(Cybersecurity Tech Accord or Charter of Trust)*

Near Term Recommendations

First

1. Develop a **Lifecycle Security Overlay** to reduce complexity of implementing NIST's SP 800-53 (*NIST, Industry)*

2. Offer public **reference implementations** of the Overlay *(Industry, cloud service providers)*

7. Provide funding (~$25M) for **baseline security improvements** to open source software (*DHS CISA)*

13. New int'l partners **to investigate & prevent** catastrophic software supply chain attacks (*CISA, NSA, DoJ, NCSC)*

14. **Support strong cryptography** in standards & tools *(USG & allies)*

Next

10. Establish **Transatlantic Infrastructure Initiative** with European allies to resource improved security for critical OSS packages (*CISA, State*)

15. **Annual Survey** software supply chain attacks that undermine updates, code integrity, or open-source repositories *(Cybersecurity Tech Accord or Charter of Trust)*

16. Internationalize the software bill of materials (**SBOM**) effort *(NTIA, State)*

(Selected) Recommendations

Improve the Baseline

1. Develop a Lifecycle Security Overlay to help implement 800-53 for software supply chains (*NIST, Industry)*

2. Offer public reference implementations of the Overlay (*Industry, cloud service providers)*

5. Recognize software is part of 5G (*State, NSC)*

Better Protect Open Source

7. Provide funding (~$25M) for baseline security improvements to open source software (*DHS CISA)*

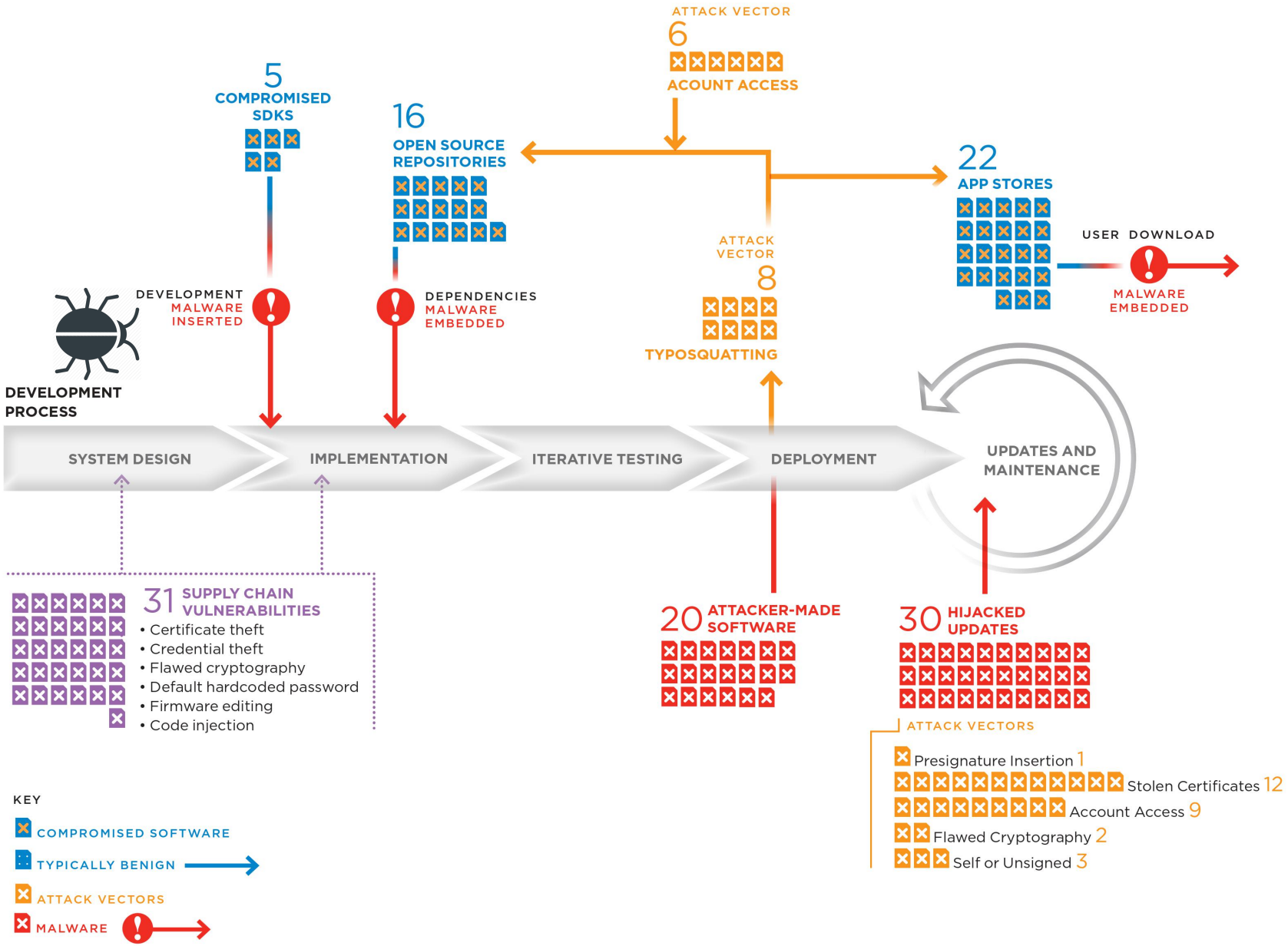
10. Establish Transatlantic Infrastructure Initiative with European allies to resource improved security for critical OSS packages (*CISA, State*)

Counter Systemic Threats

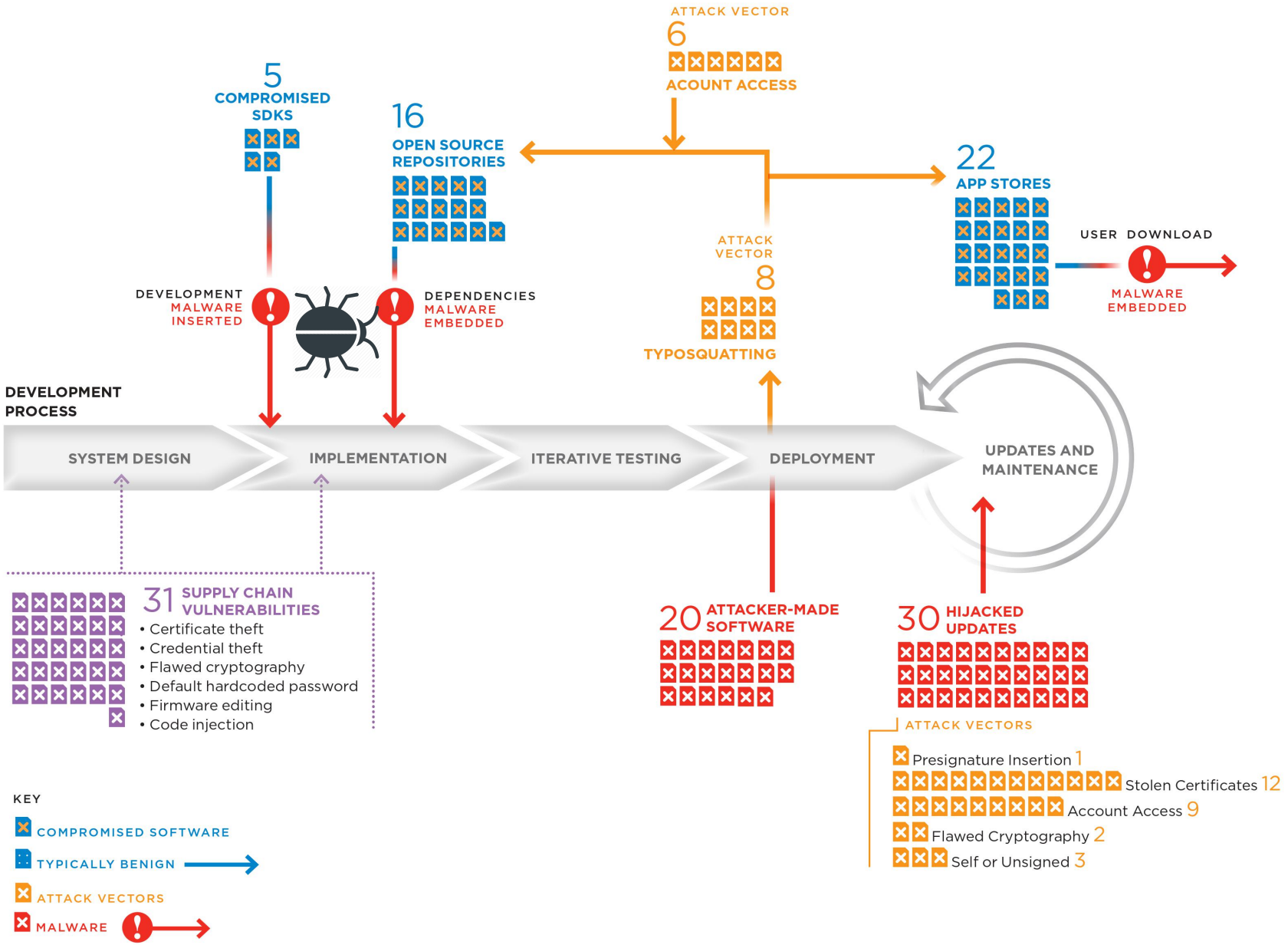
13. New intl' partners to investigate & prevent catastrophic software supply chain attacks (*CISA, NSA, DOJ, NCSC)*

14. Support strong cryptographic standards & tools (*USG & Allies)*

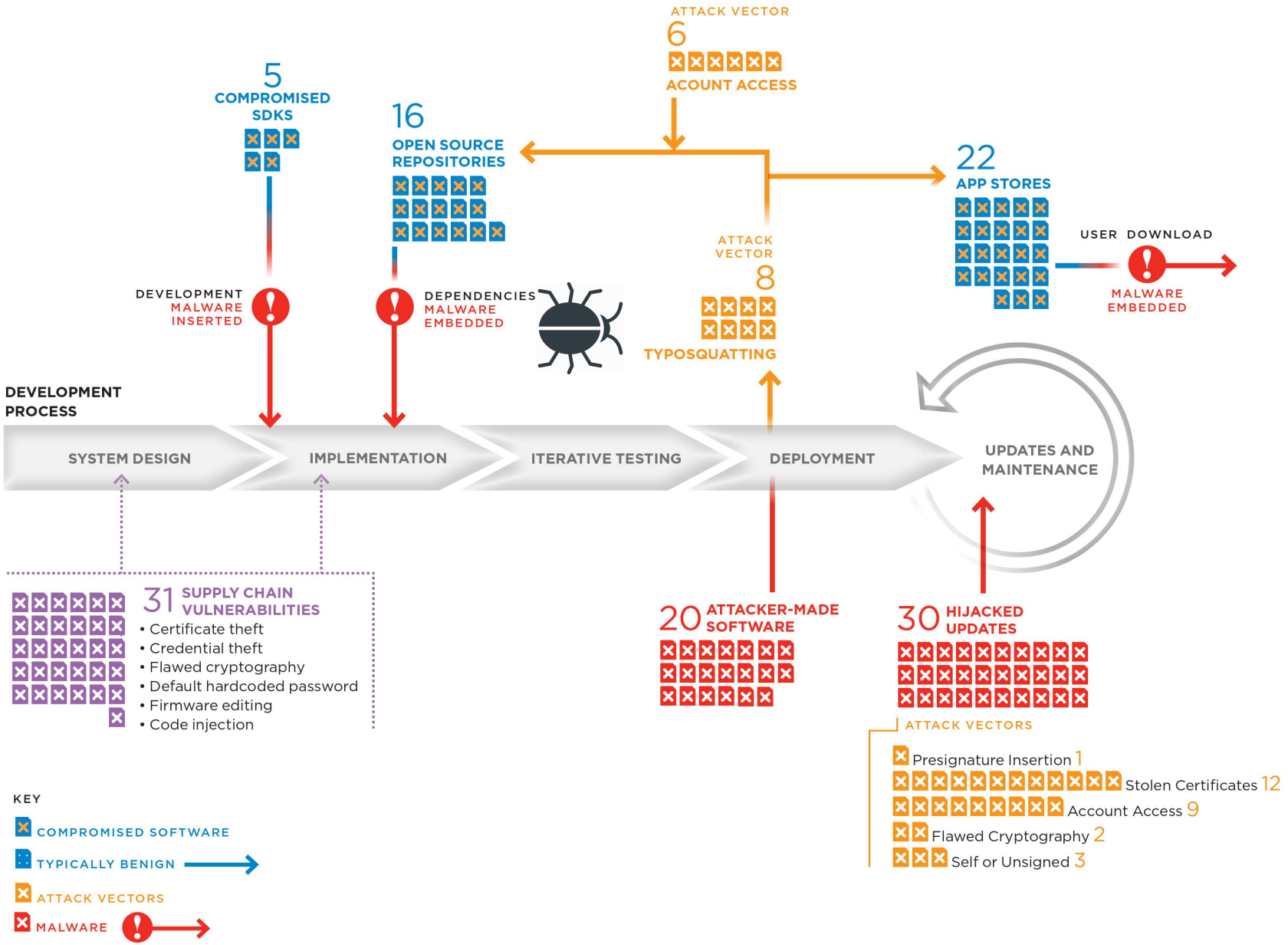
16. Internationalize the software bill of materials (SBOM) effort *(NTIA, State)*



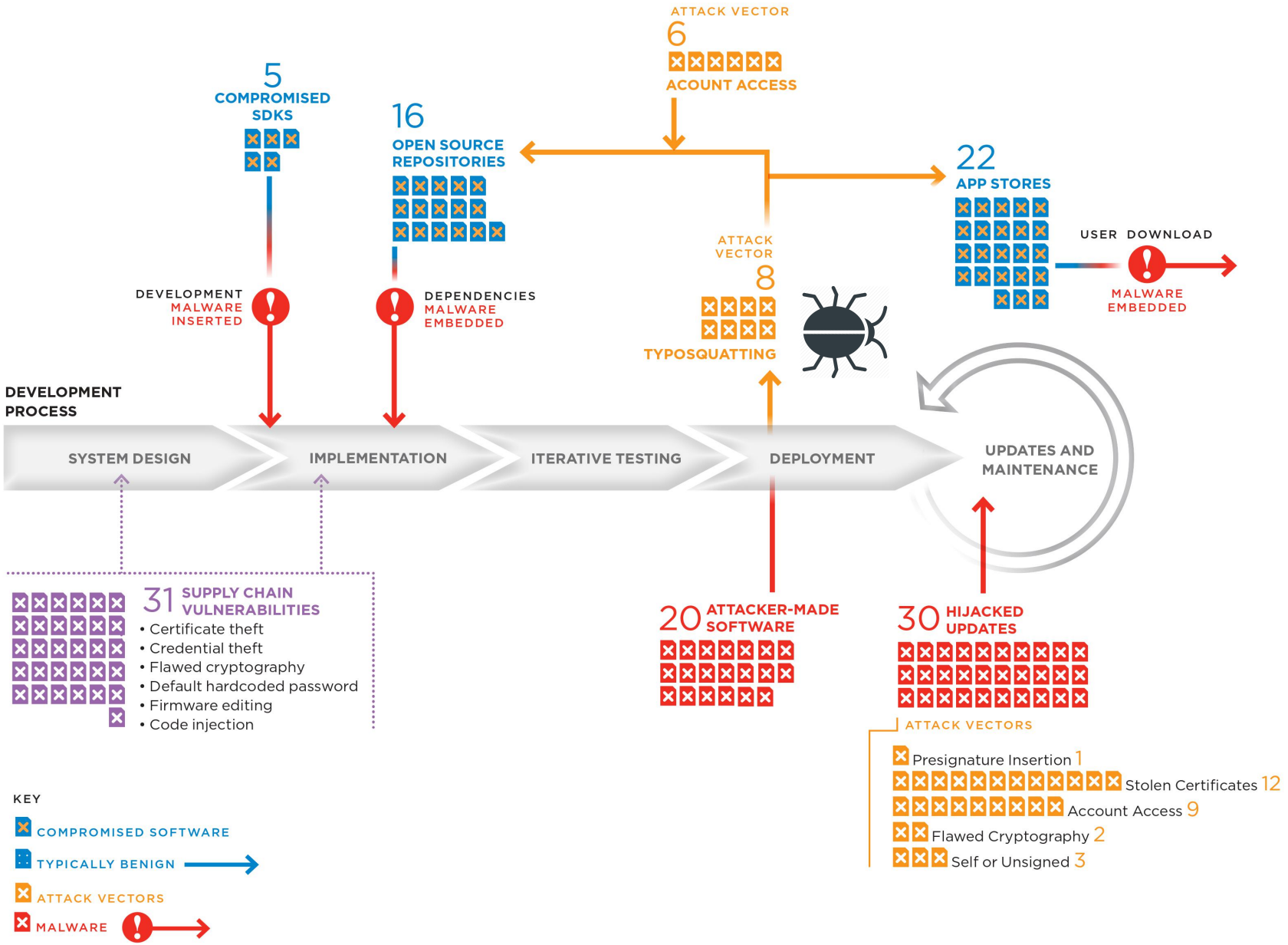
The infection starts with a malicious development kit (SDK) named RXDrioder, masquerading as an ad kit developers could use to help overlay ads in their applications.



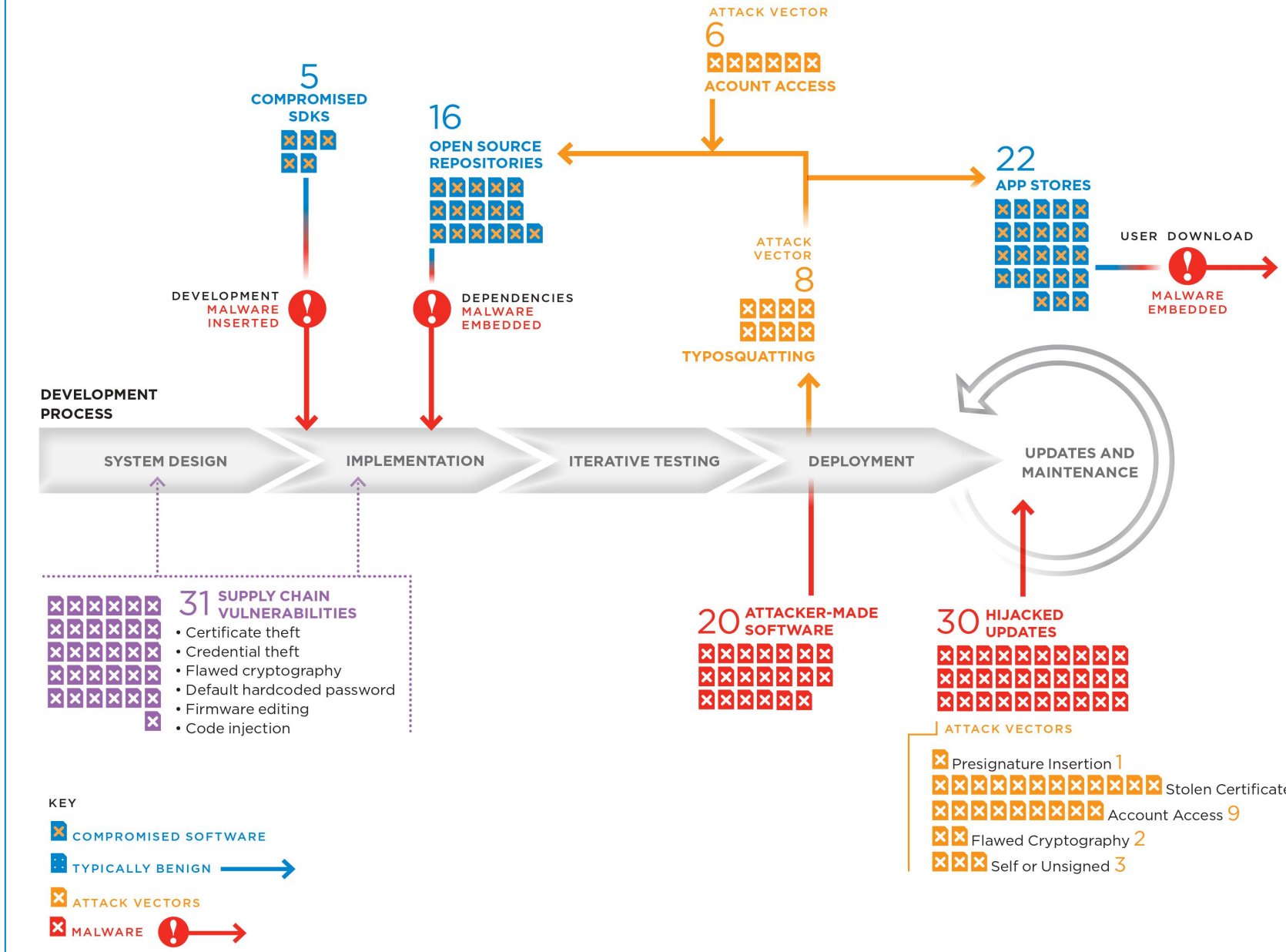
The infection spreads – via the malicious SDK used by specific app developers – to 206 Android applications during the implementation phase of development. The malicious code remains hidden during this phase.

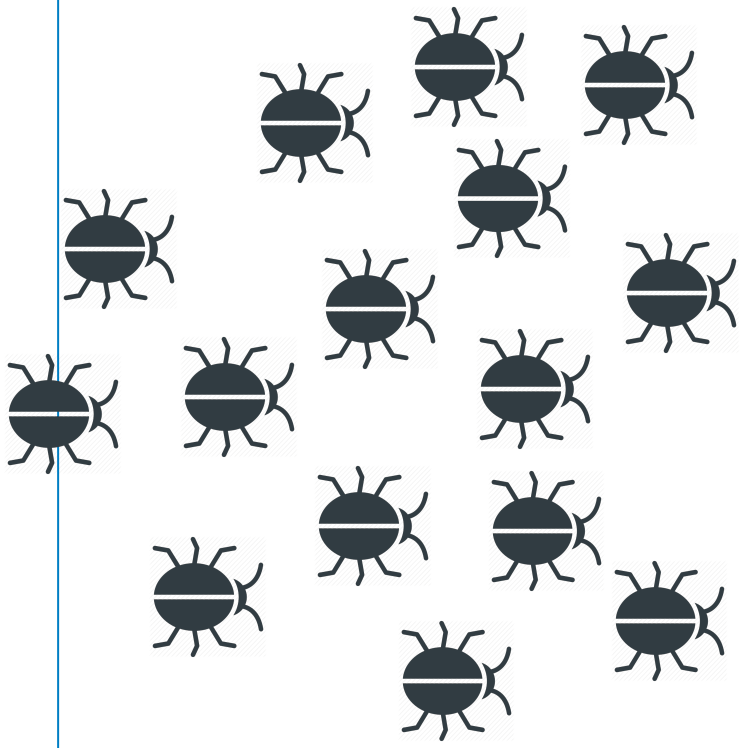


Each compromised app goes through iterative testing by its developer. The malicious code – in RXDrioder's codebase – remains undetected.



The compromised app passes through iterative testing and is ready for deployment. The finished product moves from the individual app developers to teams at app stores such as Google Play or 9Apps for review.





Once the app passes inspection by the app store teams, it is launched for download by the public. SimBad registers with a C2 server on install and takes steps to obfuscate its presence. Most of the apps targeted by SimBad were games and were downloaded by a total of 150 million people before it was identified and mitigated.