Technology
Innovation
Institute

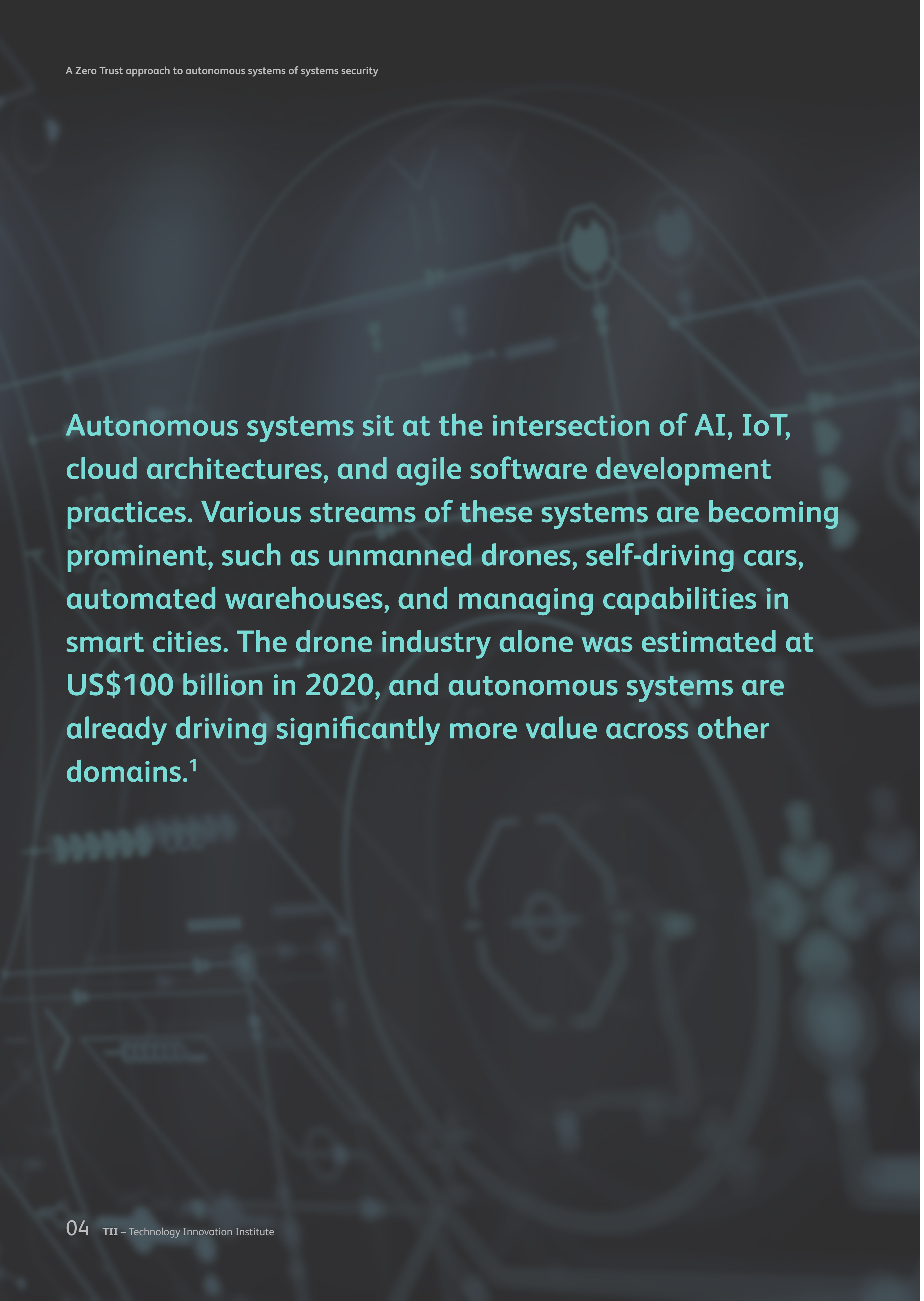
A Zero Trust approach to autonomous systems of systems security

Secure Systems
Research Center

tii.ae

Innovation for a better world

Contents



Autonomous systems sit at the intersection of AI, IoT, cloud architectures, and agile software development practices. Various streams of these systems are becoming prominent, such as unmanned drones, self-driving cars, automated warehouses, and managing capabilities in smart cities. The drone industry alone was estimated at US$100 billion in 2020, and autonomous systems are already driving significantly more value across other domains.[1]

But surprisingly little attention has been paid to securing autonomous systems as systems composed of multiple automated components. Various patchwork efforts have focused on individual components. In tandem, cloud services are starting to adopt a Zero Trust approach for securing the chain of trust that might traverse multiple systems. With that, it has become imperative to extend a Zero Trust architecture to systems of autonomous systems to protect not only drones, but also industrial equipment, supply chain automation, and smart cities.

In the near future, autonomous systems will bring a new level of digital transformation to the industry worth trillions of dollars, including automating transportation, traffic management, municipal services, law enforcement, shipping, port management, construction, agriculture, and more. Autonomous enterprise systems are further enriching these more physical aspects of autonomous systems. Gartner coined the term hyperautomation to describe tools for scaling automation using software robots that were valued at US$534 billion in 2021.[2]

Despite the importance of autonomous systems, surprisingly little research has focused on securing autonomous systems as a collection of systems. This is not to say that researchers are ignoring security – after all, security infrastructure and tools are a multi-billion dollar industry. But when it comes to securing physical components, much of the focus has been on securing individual elements such as data, software, hardware, and communications links rather than the behavior of an ensemble of autonomous systems.

Similarly, researchers are just starting to scratch the surface of protecting against swarms of autonomous things guided with malicious intent. Just last year, a half dozen precisely targeted malicious drones managed to slow oil production in Saudi Arabia for days, and more recently, several low-cost drones caused significant damage to oil tankers in the UAE. This illustrates the importance of detecting alien drones entering secure spaces.

This kind of security is just the beginning of what will be required to move towards a larger scale deployment of drones as envisioned by the US FAA's beyond visual line of sight (BVLOS) regulations.[3] These regulations promise to open immense commercial opportunities to improve industrial inspection, shipping, and remote monitoring. However, wider scale deployment will require a more systemic approach to protect against the impact of thousands of low-cost autonomous drones working in concert.

Autonomous security is also a pressing issue for industrial systems and smart city use cases. Hackers are becoming better at coordinating millions of IoT devices to launch a devastating distributed denial of service attacks on computer servers today. Similar tactics that leveraged physical and mobile autonomous things could extend the blast radius beyond IT infrastructure to destroy physical infrastructure like factories, pipelines, electric grids, or worse.

A more comprehensive approach is required to protect the security and resilience of autonomous systems and protect against cyber-physical attacks that leverage autonomous systems. The Technology Innovation Institute's Secure Systems Research Centre is leading one promising approach to building an autonomous security testbed that explores the interplay between how hardware, software, and communications systems can be exploited so that they can be hardened. The early phases of this work are focused on protecting scalable swarms of unmanned aerial vehicles controlled by the cloud. The long-term goal is to create a framework for understanding and defending against autonomous security risks across all types of infrastructure, including fleets of cars, automated warehouses, construction sites, farms, and smart cities.

[1] Goldman Sachs. "Drones: Reporting for Work." Accessed March 16, 2022. https://www.goldmansachs.com/insights/technology-driving-innovation/drones/.

[2] Gartner. "Gartner Forecasts Worldwide Hyperautomation-Enabling Software Market to Reach Nearly $600 Billion by 2022." Accessed March 16, 2022. https://www.gartner.com/en/newsroom/press-releases/2021-04-28-gartner-forecasts-worldwide-hyperautomation-enabling-software-market-to-reach-nearly-600-billion-by-2022.

[3] "Advisory and Rulemaking Committees – Unmanned Aircraft Systems (UAS) Beyond Visual Line-of-Sight (BVLOS) Operations Aviation Rulemaking Committee (ARC)." Template. Accessed March 16, 2022. https://www.faa.gov/regulations_policies/rulemaking/committees/documents/index.cfm/committee/browse/committeeID/837.

The promise of autonomous systems

Over the years, basic autonomous capabilities have grown into almost every aspect of our physical infrastructure, from automated braking in individual cars to orchestrating power flow across nationwide electrical grids with precision.

Autonomous systems are already demonstrating tremendous value today, and we are just scratching the surface. For example, Goldman Sachs estimated that unmanned autonomous vehicles (UAV) had grown into a $100 billion industry in 2021.[4] Military applications accounted for about 70% of this spending. However, commercial applications were also substantial in construction, agriculture, insurance claims, offshore oil, gas and refining, pipelines, utilities, and mining.

For example, the construction industry uses drones to automatically capture footage of construction sites before, during, and after the construction process. Drones carrying lidar and high-resolution cameras can automatically generate 3D models in minutes that would have previously taken humans days or weeks. This makes it practical to keep tabs on buildings as they are being built, track progress, and identify mistakes when they are cheaper to fix. After construction, drones can also survey physical infrastructure like bridges to identify cracks and other problems before the whole structure suffers a bigger problem.

Drones are also improving the planning and management of large farms. For example, drones with spectral imaging cameras can quickly identify nutrient deficiencies, pest outbreaks, and drought, allowing farmers to address them more precisely and cheaply. In the UAE, drones have also helped map the entire country's agricultural resources in a matter of days, which would not have been practical using physical surveys alone.[5] In another effort, UAE teams used drones to plant 6.25 million trees in only two days.[6]

[4] Goldman Sachs. "Drones: Reporting for Work." Accessed March 16, 2022. https://www.goldmansachs.com/insights/technology-driving-innovation/drones/.

[5] Ford, Georgina. "Counting Camels in The Desert - A Drone-Powered Success Story." Commercial Drone Professional (blog), September 30, 2021. https://www.commercialdroneprofessional.com/counting-camels-in-the-desert-a-drone-powered-success-story/.

[6] Douglas, Alex. "UAE to Emerge as World Leader in Using Drones, Predicts Falcon Eye." Commercial Drone Professional (blog), April 1, 2020. https://www.commercialdroneprofessional.com/uae-to-emerge-as-world-leader-in-using-drones-predicts-falcon-eye/.

Scaling autonomous systems

It is easy to get caught up in autonomous systems as a single self-driving car or individual drone. However, the real promise of autonomous systems comes when autonomous capabilities are simultaneously scaled to improve the control of individual things, the orchestration of a collection of things, and the understanding of things at scale.

The individual-level can be considered as the evolution from cruise control to automated braking and fully self-driving cars. The orchestration level entails the evolution from synchronized traffic lights to dynamically adjusted traffic lights to advanced mapping services that route cars around traffic jams. Autonomous understanding systems include traffic monitoring cameras to crowdsourcing dashcam video into dynamically updated digital twins for improving overall traffic.[7]

These same three factors of control, orchestration, and understanding play out across various use cases. A warehouse robot might reduce the need for staff. An autonomous warehouse management system could optimize the scheduling and staging of items in the warehouse. In contrast, an autonomous understanding system could help reengineer the warehouse design to further increase performance in the same space.

This combination of autonomous control, autonomous orchestration, and autonomous understanding is already showing some promise in the UAE. For example, one pilot project has created an autonomous port truck system that automates the process of shifting shipping containers from boats to trucks.[8]

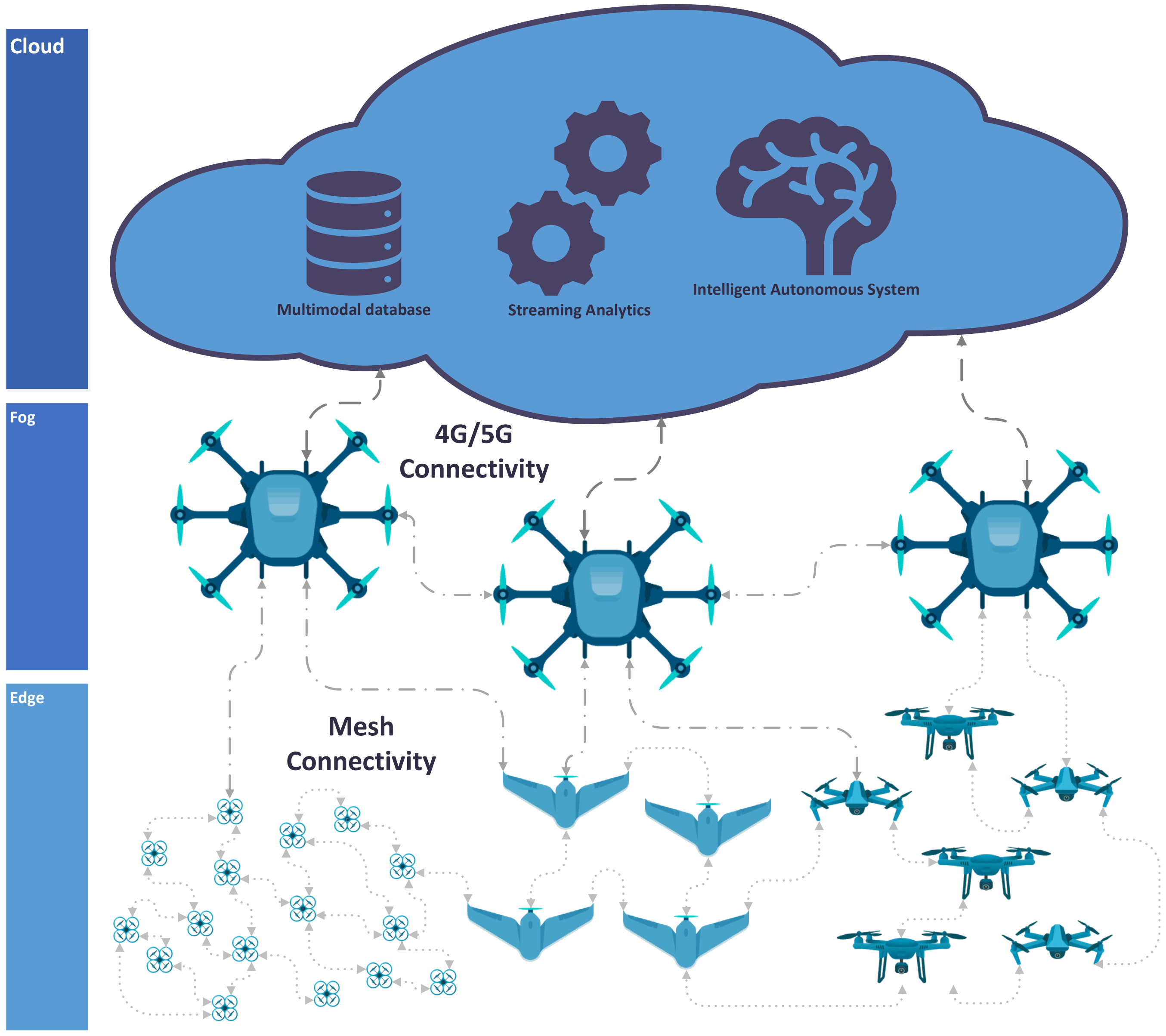
Gartner refers to the simultaneous evolution of control, orchestration, and understanding in IT systems as hyperautomation. In this case, enterprises use individual robotic process automation (RPA) software robots (called bots) to automate a collection of human tasks. Orchestration engines help organize the flow of work across multiple bots. Then process and task mining bots analyze enterprise applications or even watch over the shoulders of individuals to find further opportunities for improvement.

Researchers are just starting to explore how similar practices may be extended to include autonomous vehicles. That is one of the reasons ATRC's ASPIRE chose to focus on autonomous swarm coordination as part of its next grand challenge project.[9] ASPIRE is tasked with hosting grand challenge competitions loosely organized like the US DARPA's challenge that spearheaded research on autonomous vehicles. The upcoming challenge tasks researchers with finding the best way to orchestrate a swarm for drones to search for and retrieve objects hidden on ships that are too heavy for any individual drone.

[7] VentureBeat. "Nexar and Las Vegas Tackle Traffic with Digital Twins," September 27, 2021. https://venturebeat.com/2021/09/27/nexar-and-las-vegas-tackle-traffic-with-digital-twins/.

[8] "Region's First Autonomous Port Truck System to Be Implemented - GulfToday." Accessed March 16, 2022. https://www.gulftoday.ae/business/2021/07/06/regions-first-autonomous-port--truck-system-to-be-implemented.

[9] Defaiya, Al. "Al Defaiya | Abu Dhabi's ASPIRE Launches Over US$3 Million MBZIRC Maritime Grand Challenge," October 22, 2021. https://www.defaiya.com/news/Regional%20News/UAE/2021/10/22/abu-dhabi-s-aspire-launches-over-us-3-million-mbzirc-maritime-grand-challenge.



The need for end-to-end security and resilience

Enterprises and security researchers are just starting to struggle with protecting individual autonomous things, much less swarms. A new security approach is required for these types of swarms to scale for real-world applications.

The early generation of IoT devices were rushed to market with only basic considerations on how they might be protected against hackers or securely updated against new threats. Many of these early devices are not updateable after the fact. Consequently, they are a popular target for hackers eager to create large-scale botnets for launching distributed denial of service attacks such as the Mirai botnet.[10] This has given rise to a secondary industry of IoT security gateways designed to detect and block malicious activity outside of poorly secured appliances like lighting controllers, crockpots, TV set-top boxes, and cameras.

The security posture of the first connected cars is better, but there are still glaring vulnerabilities and gaps that need to be addressed. Some of the vulnerabilities highlighted in Upstream's 2021 Automotive Cyber Security Report[11] include:

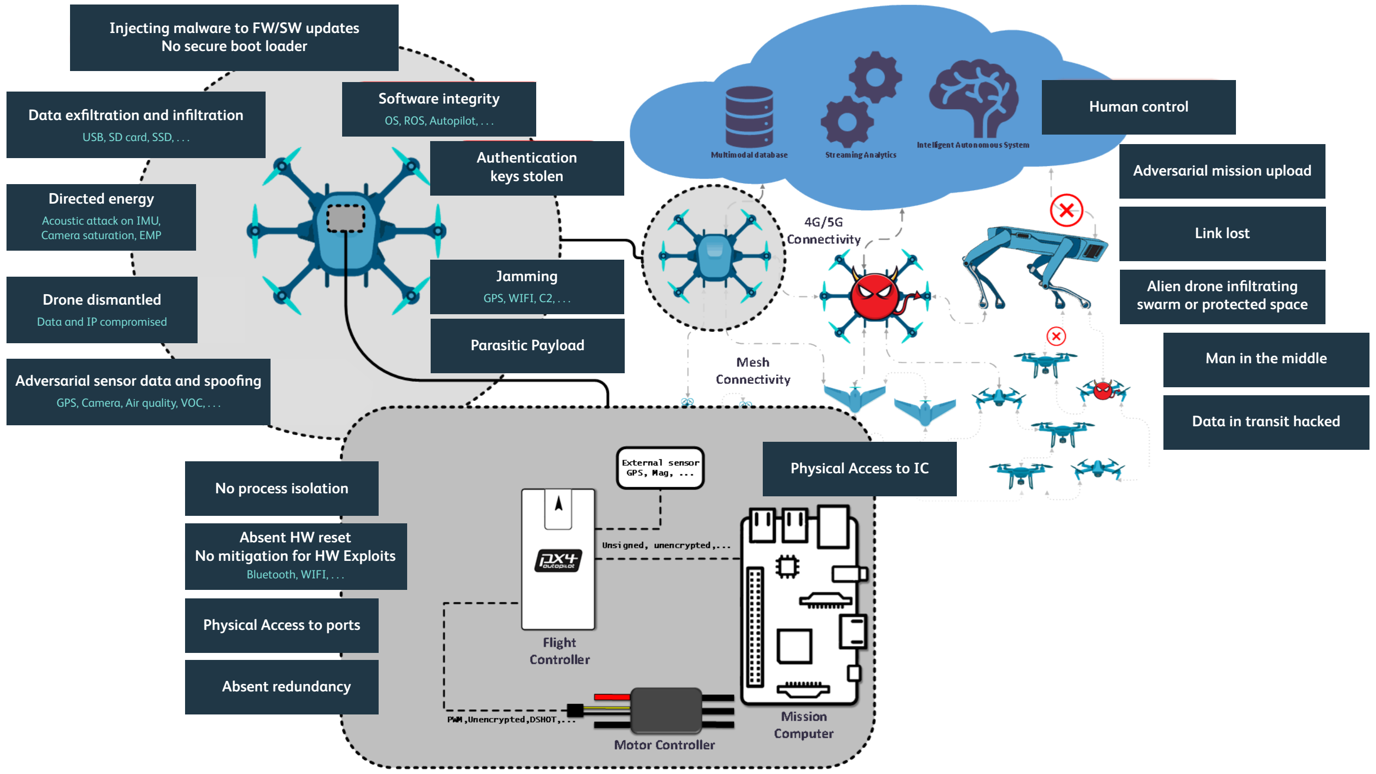
- Hackers found 19 vulnerabilities in a Mercedes-Benz E-class car that allowed them to remotely control the vehicle, open doors, and start the engine.
- Hackers took control of a car's OEM corporate network by reverse engineering a car's transmission control unit to infiltrate the network.
- Over 300 vulnerabilities were discovered in 40 popular electronic control units used in cars.
- Hackers managed to gain control over Tesla's entire connected car fleet by exploiting a vulnerability in the communications protocol.

Modern cars allow the software to be updated after the fact but generally require consumers to come to a shop for an update. Only a few leaders like Tesla, have mastered the ability to securely update software at scale.

Building secure systems will need to address hardware, software, and protocols and their interplay. Hardware security issues need to protect against attacks in which a hacker can physically update a system to compromise security or cause damage. For example, the Stuxnet[12] attack corrupted hardware in an Iranian uranium enrichment facility to send miscalibrated timing data that confused the control systems. The result was that the controller drove hundreds of expensive centrifuge systems so fast that they exploded.

There are a variety of ways hackers could launch remote hardware-directed attacks on UAVs. For example, focused beams of sound could confuse the inertial guidance unit used to control a drone. Directed EMF beams might cause a short circuit on sensitive electronics, and lasers or bright lights might confuse or destroy camera sensors.

Vulnerabilities in software systems allow hackers to spy on or take remote control of systems to launch further attacks. Early examples in IT systems included malware like the Zeus Trojan that allowed hackers to spy on banking interactions to capture credentials and steal $500 million.[13] In some cases, hackers are finding ways to infiltrate software supply chains to plant targeted malware vulnerabilities. This was how hackers managed to burrow into thousands of government, banking, and enterprise systems as part of last year's Solar Winds breach.



[10] "The Mirai Botnet Explained: How IoT Devices Almost Brought down the Internet | CSO Online." Accessed March 16, 2022. https://www.csoonline.com/article/3258748/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html.

[11] Upstream Security. "2021 Automotive Cybersecurity Report | Press Release | Upstream." Accessed March 16, 2022. https://upstream.auto/press-releases/2021-report/.

[12] Kushner, David (26 February 2013). "The Real Story of Stuxnet". IEEE Spectrum. 50 (3): 48–53. doi:10.1109/MSPEC.2013.6471059. S2CID 29782870.

[13] "$500 Million Botnet Citadel Attacked by Microsoft and the FBI | The Independent | The Independent." Accessed March 16, 2022. https://www.independent.co.uk/tech/500-million-botnet-citadel-attacked-by-microsoft-and-the-fbi-8647594.html.

Applying a Zero Trust approach to autonomous systems

The term Zero Trust model was coined by Forrester research in 2010 to denote a new paradigm for securing distributed systems.[14] Security systems have traditionally been secured by hardening a physical perimeter. But in the world of cloud computing, the perimeter is more nebulous. Zero trust security connotes the idea of always authenticating and verifying every access in order to secure around a more flexible perimeter.

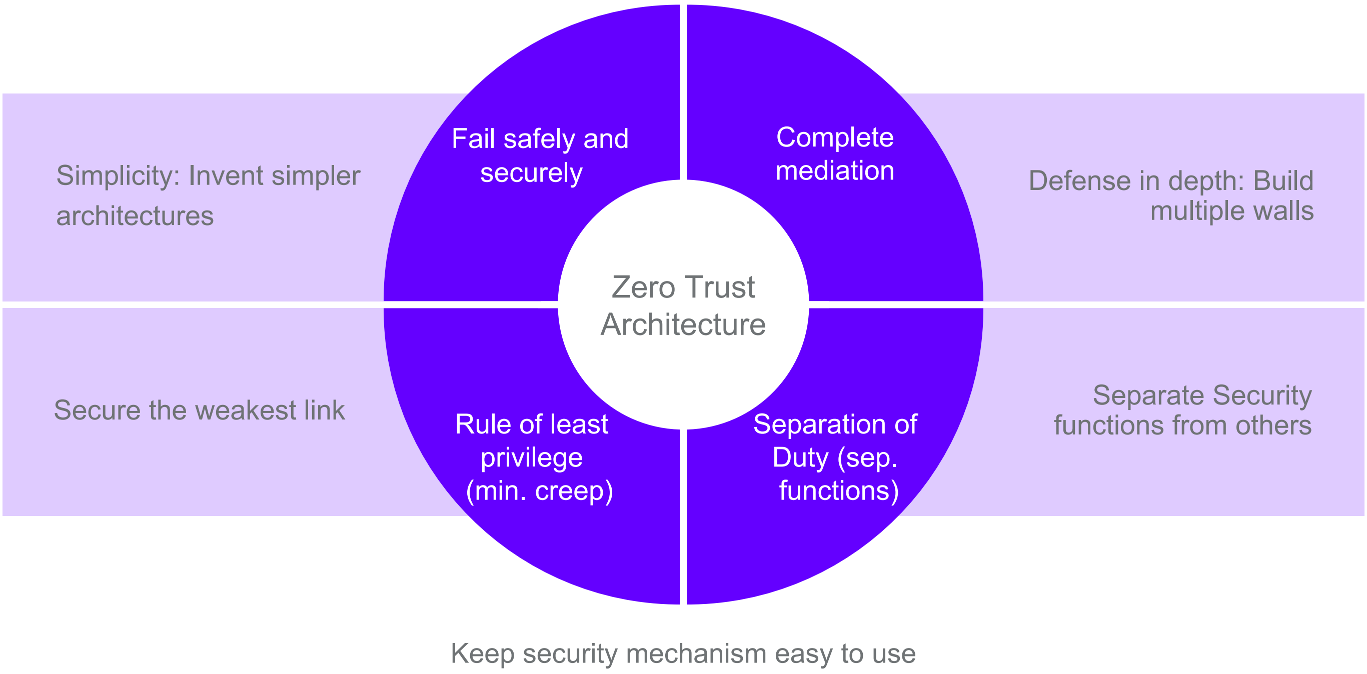
The Zero Trust paradigm allows security teams to plan for the possibility that vulnerabilities may exist throughout a chain of interactions among multiple systems, such as across several cloud services, data processes, storage services, and networks. The fundamental concept is to never trust and always verify the provenance of each request. Another basic principle is to assume that a breach has already occurred, making it essential to limit the blast radius of any breach.

Autonomous systems extend automated processes across a wider variety and physical range of hardware, communications protocols, as well as control and orchestration mechanisms. Each of these brings with them their own attack surface. Thus, security teams need to minimize the impact that a breach on one level could have on other systems. Examples include attacks on control servers, communication networks, embedded system applications, physical devices, software supply chains, and silicon supply chains.

Autonomous system security needs to be built across multiple independent security walls so that if one key or system is breached, the integrity of the whole is protected. Each system should be designed to fail safely and securely so as to minimize the impact on adjacent components. This can also make it harder for hackers to escalate an attack on a low-level system to more critical systems, as with the recent Log4J attacks.

For example, Autonomous systems need attestation schemes to ensure that only authorized software runs on the drones. An attestation scheme uses cryptographically signed software updates to ensure that only valid code can run on remote systems like autonomous drones. This prevents hackers from reprogramming a drone by simulating a legitimate program update communication or replacing legitimate updates with a bogus software upgrade staged at the command center.

Vulnerabilities in communication protocols could allow hackers to spy on drone activity or simulate control signals to take control of a drone. Such attacks could happen at any level of the communication stack, from hacking into communications within the cloud, the wireless signals between the cloud and a drone, or between multiple drones. In some cases, hackers may be able to attack systems by mimicking communications within a drone or autonomous car. For example, researchers have found ways to listen to and simulate the unprotected wireless communications involved in tire pressure monitoring.[15] This allowed them to trick the car into indicating that a good tire had a flat, which might cause a vehicle to stop.


Simplicity: Invent simpler architectures
Fail safely and securely
Complete mediation
Defense in depth: Build multiple walls
Zero Trust Architecture
Secure the weakest link
Rule of least privilege (min. creep)
Separation of Duty (sep. functions)
Separate Security functions from others
Keep security mechanism easy to use

Source: Martin Dixon: https://www.intel.com/content/www/us/en/newsroom/opinion/zero-trust-approach-architecting-silicon.html#gs.aqdrlz

[14] September 17, Kelly Jackson Higgins Editor-in-Chief and 2010. "Forrester Pushes 'Zero Trust' Model For Security." Dark Reading, September 17, 2010. https://www.darkreading.com/perimeter/forrester-pushes-zero-trust-model-for-security.

[15] BAE Systems I Cyber Security & Intelligence. "Security Challenges for Connected and Autonomous Vehicles." Accessed March 16, 2022. https://www.baesystems.com/en/cybersecurity/feature/security-challenges-for-connected-and-autonomous-vehicles.

Trust is essential in computing systems – arguably more so for chips at the heart of these systems. Unfortunately, trust is also an increasing rarity, because many chips design companies are outsourcing critical steps (fabrication, testing, assembly) to third-party companies. Such a distributed chip supply chain is financially appealing, but under normal circumstances, also necessarily untrustworthy. This distributed paradigm is susceptible to threats such as chip design reverse engineering, piracy, overproduction, and tampering. A rogue element in the fabrication with full access to the chip design blueprint can reverse engineer the functionality of the chip or its critical components, copy and pirate any hardware design intellectual property, run extra fabrication shifts to produce more chips than requested by the design house to sell them in a gray market, or insert difficult-to-catch Trojans that serve a malicious purpose (e.g., leak sensitive information) into the chips during fabrication.

Software supply chain attacks have been making the news lately. It's also essential to protect against hardware supply chain attacks in which malicious actors insert backdoors or hardware Trojans into chips. Emerging chip-to-chip authentication techniques could help mitigate such issues. The core idea is to extend zero-trust concepts applied to network security to chip-to-chip communication to mitigate the impact of attacks on the physical supply chain or malicious firmware updates. This kind of approach could involve combining public-key infrastructure, trusted computing, and secure memory management to strike the right balance between security and performance.

For example, in a drone system, designers might have a flight controller that connects many peripheral chips for motor control and sensors of various kinds. Today, there is no authentication of those chips done in real-time when the system boots. It is assumed that it is a legitimate chip because it boots up in a particular manner. We need a Zero Trust approach in which the Boot processor cryptographically verifies that these peripheral chips are from legitimate manufacturers and are running legitimate software before allowing them to connect with the main CPU on the flight controller. This chain is extended all the way from this level to applications running on the flight controller.

Researchers have been developing a technique called logic locking[16] which gives control back to the design house in the chip supply chain where they normally have almost no control. By using a logic locking tool or technique, a chip designer can insert additional logic into the design to introduce a locking mechanism that expects a secret unlock key, which is a binary vector (combination of 0s and 1s). The secret key is known to only the design house and is loaded by a trusted party (e.g., design house themselves) on the chip after fabrication. This is a one-time load operation where the key is written into the chip. Only then a fabricated chip becomes "unlocked," and thus, functional.

Logic locking serves multiple purposes. First, the design house can ensure that all of its fabricated chips can be deployed in the market under their control; any overproduced chip by the fab will remain locked and unfunctional, as it will be missing the unlock key. Second, the blueprint that is available to the fab fails to reveal all the information about the functionality of the chip and its blocks as the secret key is unknown to the untrusted entities. Any attempt to reverse engineer the chip/block functionality is thus hindered. Without the functionality of the chip fully understood, the insertion of meaningful Trojans in the foundry is also thwarted.

Modern system-on-chip designs can accelerate product development for performant and low-cost chip functionality. However, they also carry risks from the use of untrusted IP. Existing testing techniques like fuzzing and penetration tests depend on the judgment of experts. Also, they tend to be performed late in the design cycles, and it can be costly and challenging to make significant changes when problems are found. Approaches like concolic testing ("concrete" plus "symbolic), primarily used in software security testing today, could be extended to chip circuit design to detect problems much earlier in the design cycle.

It's also vital to extend confidential computing security[17] to hypervisors running on RISC-V processors that are increasingly being adopted in autonomous systems. The core idea is to isolate virtual machines from the virtual machine manager and other non-trusted software components available on the platform. This will require a combination of VM-to-VM authentication and encrypted communication. One challenge is that RISC-V processors do not currently provide hardware support for encrypted communication channels between VMs. Implementing this capability in software adds additional overhead and latency. One strategy is to create Zero Trust hardware building blocks such as IOMMU and IOPMP and ISA extensions to alleviate this overhead.

Trusted execution environments (TEE) were developed to provide a higher level of security for applications by using an encryption perimeter around program execution running on the hardware, but these were built primarily for applications confined within a CPU. Autonomous systems infrastructure needs to combine a variety of embedded computing platforms such as drone navigation systems, CPU-based architectures, and other types of dedicated hardware. Existing approaches are also fixed at design time, which leads to using untrusted software to employ peripherals in TEEs. New approaches for composite enclaves will be required to extend TEEs to more flexible designs.

It's also essential to develop new tools for detecting and responding to unknown and unexpected changes caused by novel attack techniques. A trust verification infrastructure could extend traditional API observability approaches to hardware through a combination of monitoring, logging, and tracing. These kinds of actions allow the construction of continuous verification mechanisms for anomaly and intrusion detection. Offline profiling techniques could generate trust profiles that describe how the hardware is supposed to operate. During operation, ongoing logging could ensure that the behavior adheres to the trust profile. However, this needs to be constructed to minimize the risk of logging tools being leveraged as part of a side-channel attack.

We also need to explore new hardware capabilities and advanced software techniques to compartmentalize software stacks across multiple levels. One of the most promising approaches is CHERI (Capability Hardware Enhanced RISC Instructions), which is being explored by DARPA, Google, SRI International, and the University of Cambridge.[18] Further work is required to extend this work to improve fine-grained compartmentalization at the operating system level. This could combine new middleware, OS libraries, unikernels, and various mechanisms to grant and revoke authorization in order to enforce compartmental constraints. This will be required striking the right balance between different degrees of flexibility in both configuration and determining the appropriate privileges.

We are still in the early days of building large-scale autonomous systems, but as we scale them up, new considerations like these will be required to extend zero-trust security to embedded systems, autonomous systems, and systems of autonomous systems.

[16] Yasin, Muhammad, and Ozgur Sinanoglu. "Evolution of logic locking." In 2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 1-6. IEEE, 2017.

[17] Rashid, Fahmida Y. "The rise of confidential computing: Big tech companies are adopting a new security model to protect data while it's in use-[news]." IEEE Spectrum 57, no. 6 (2020): 8-9

[18] Watson, Robert NM, Peter G. Neumann, Jonathan Woodruff, Michael Roe, Jonathan Anderson, David Chisnall, Brooks Davis et al. Capability hardware enhanced risc instructions: Cheri instruction-set architecture (version 5). No. UCAM-CL-TR-891. University of Cambridge, Computer Laboratory, 2016.

Vision for secure autonomous systems of drones

Enterprises and researchers are exploring ways to scale systems of individual autonomous systems, with the most promising research currently being focused on scaling systems of autonomous drones. In the long run, everyone wants to get to autonomous cars and factories and there is a lot of experimentation going on with unmanned vehicles that tend to require a human driver or assistant in the case of delivery vehicles.

But UAVs are already delivering value today, and regulators are starting to open the skies for more ambitious applications. The US FAA granted American Robotics the first license to fly drones beyond the visual line of sight (BVLOS) in early 2021.

Around the world, enterprises are working with regulators to develop Unmanned Traffic Management (UTM) systems. Major aerospace companies and innovative start-ups are working with regulators to show how various combinations of AI, advanced mapping, vehicle-to-vehicle communications, and encrypted communication and control could facilitate safe drone management at scale.

In the US, Boeing has partnered with SparkCognition on SkyGrid. Airbus is leading efforts to promote SESAR for the EU. Guardian Angel recently worked with UK regulators on Operation Zenith to demonstrate how a series of on-airfield tasks could be performed without endangering or disrupting airport operations. These are essential efforts and are a necessary first step in safely scaling fleets of trusted drones.

However, UTM systems generally start with the assumption that drones are all trusted. More work needs to be done to understand and analyze how these systems can be compromised and hence trusted in the first place.

Researchers around the world are exploring how individual components of these systems can be compromised and hardened. For example, researchers in Germany and Switzerland have experimented with implementing quantum-safe cryptographic algorithms to protect drone communications. They argue that long-running drones will also need to support crypto-agility that allows dynamic updating of security algorithms in response to the discovery of new vulnerabilities.[19] These researchers also explored how to implement remote attestation schemes that protect drones from software tampering.

Other researchers have explored drone cloud control mechanisms. For example, a team of researchers in Brazil has developed the Cloud-SPHERE platform as one approach for integrating UAVs into IoT and Cloud Computing paradigms

[19] Katzenbeisser, Stefan, Ilia Polian, Francesco Regazzoni, and Marc Stöttinger. “Security in Autonomous Systems.” In 2019 IEEE European Test Symposium (ETS), 1–8, 2019. https://doi.org/10.1109/ETS.2019.8791552.

Bringing security to the swarm

The next phase of autonomous drones will require developing architecture to scale drone control and security to support autonomous swarms. For example, a collection of low-cost drones can be orchestrated into drone swarms controlled by the cloud to explore new use cases like search and rescue, disinfecting public spaces, and coordinating tasks such as lifting heavy equipment beyond the capacity of any one drone.

One big shift will be the need for more distributed control mechanisms. Architectures that attempt to control each drone or autonomous system directly will run into scalability challenges as the number of individuals in the swarm grows. One approach pursued by the TII's Secure Systems Research Centre (SSRC) is the development of a dynamic hierarchy composed of drones with different capabilities for control and task execution. Similar organization of drones has been described before, and our focus is going to be on security and resilience in such a hierarchy.

In this scheme, a tier of Fog Drones acts as intermediaries between less sophisticated Edge Drones and the cloud. The Fog Drone can also offload many tasks such as summarizing input from many Edge Drones to reduce the amount of communication required with the cloud and between drones. This can also reduce the amount of processing required on each Edge Drone. This work is also exploring how mesh networks can further optimize and secure communications between drones operating in constrained situations such as a cave, fallen building, or hostile environment.

More importantly, SSRC is working with a cross-disciplinary team of researchers at leading research institutions worldwide to develop a comprehensive Zero Trust autonomous security testbed to explore security implications spanning hardware, software, and communications at the systems level. Partners include Georgia Institute of Technology, Purdue University, University of Applied Sciences and Arts of Southern Switzerland, Tampere University, University of Turku, Khalifa University, Imperial College, University of Manchester, TU Graz, University of New South Wales, University of Modena and Reggio Emilia, University of Bologna, Sapienza University of Rome, University of Milan, University of Minho, University of Waterloo, McMaster University, NYU Abu Dhabi, and UT Dallas.

This research explores ways to synthesize lessons learned from physical testbeds into useful and actionable security models. Ultimately, these security models could help autonomous teams identify and improve autonomous systems development that spans drone hardware, software implementations, and communications choices earlier in the release cycle.

Several testbeds have also been developed at Masdar in the UAE and Purdue. One goal is to develop machine learning methods at both the drone and cloud levels to detect security issues and enable resilience. Another goal is to develop tools for testing these systems in augmented reality environments for urban settings. The teams are also exploring ways to improve the ability to capture security-related data into digital twins that reflect the security implications of drones. This will help automate the ability to reflect new security vulnerabilities discovered in the real world in the models shared with researchers.

These researchers are also finding ways to harden open-source hardware, software, and communication protocols for developing and deploying drone systems. This approach opens the architecture to a wide range of security and drone researchers to find vulnerabilities sooner. This open-source approach could also benefit from the rapid innovation that the open-source robotics community is already seeing.

Some of the underpinnings of the current platform include the PX4 advanced autopilot, NuttX real-time operating systems, and the Robot Operating System 2 (ROS2). The team has also developed and implemented an open-source RISC-V processor and system on chip with specialized security features baked in. The various teams are currently exploring the security implications of different scenarios, and these explorations are informing best practices for hardening against these kinds of issues.

Here are examples of some of these scenarios:

Hijacking a high-value cargo

A drone is attempting to transport an organ between hospitals. The attacker's objective is to hijack the drone and force it to land in another location to sell the organ in the underground economy. Possible attack strategies include spoofing the sensors, jamming GPS or optical sensors, injecting fake visual location data, or a complete takeover using the control protocol. The data from successful attacks will inform modern designs or help train machine learning algorithms.

Perimeter defense against stealthy UAV

A ground-based monitoring system uses radar, lidar, and cameras to protect a building from a hostile vehicle while disregarding other delivery vehicles in the area. The attacker's objective is to disguise an attack drone as a delivery drone to breach the defended area. One attack strategy would be to use generalized adversarial networks to mimic the behavior of legitimate drones. The team will work on secure learning algorithms that robustly identify these fake drones.

Swarm hijacking

Drone swarms are exposed to additional vulnerabilites beyond those experienced by individual drones. For example, hackers could sieze control of the communication link used to manage the swarm. Improvements in distributed monitoring capabilities and dynamic rerouting capabilities could improve attack detection, identification, and mitigation.

Network resiliency

A wide area swarm is deployed for long-term surveillance, such as protecting a nature reserve or border. The swarm uses a mesh network protocol to communicate, and attackers attempt to jam the network to temporarily halt communication between the swarm and the control center. As a result, a distributed optimization reconfiguration scheme is designed to allow the swarm to reconfigure itself to re-establish contact. This scenario could also help improve strategies for slowing the propagation of malicious code or data among vehicles in the swarm. For example, regular communication between the control center and drones could help identify individual drones that may have been compromised, and communications could be routed around these.

Corrupt firmware update

New capabilities are updated to the swarm via firmware and conveyed to each drone via radio. The attacker attempts to upload a corrupted firmware update with malicious intent. Various mitigation strategies include different encryption and key management schemes, ensuring firmware integrity using cryptographically signed attestation schemes, and hardening the firmware update protocol.

Exploiting unused features

Drone control systems like PX4 Autopilot and ArduPilot use QGroundControl to set up and control flights in operations, a general-purpose library. One concern is that attackers could discover unused, underutilized, or obsolete software components to initiate an exploit. These features may receive less security testing as a result. For example, an attacker may discover a way to abuse a vulnerability in video streaming features that a drone might not even use in everyday operations. Research focuses on how to map features in these systems and effectively turn off all features and disable the underlying code that is not required for a given mission. Another research direction is to develop a lightweight monitoring tool to assure the desired behavior at runtime.

A vision for the future of autonomous systems security

Today, almost all drone applications involve the management of individual drones. The next evolution of drone adoption will require finding ways to scale both the command-and-control infrastructure , as well as hardening the security and resilience of these systems. Ultimately, research around securing autonomous systems, and not just individual drones, will help facilitate widespread commercial deployment.

It is essential for designers of autonomous systems to adopt components that have been hardened and can be updated regularly as new problems are discovered. Many enterprises are adopting DevSecOps practices in which security considerations are undertaken as part of the software development and deployment. In these cases, various tools are used to vet code updates against known best practices and reject updates that fail basic security tests. Afterwards, software scanning tools, such as WhiteHat and Contrast OSS, build an inventory of libraries used by the apps, sending an alert when critical vulnerabilities are detected within active systems. Similar approaches will need to extend to improve the components used in developing and deploying autonomous systems that scan not only the software, but also the hardware and communications protocols used.

The first results of these kinds of collaborative drone security programs are just the beginning. Eventually, improvements in UAV architectures could also be used to improve the resilience of enterprise applications, autonomous warehouses, and smart cities. In addition, better tools for modelling drone security issues will also inform the development of strategies to protect against largescale attacks by swarms of compromised drones.

The FAA suggests that the evolution of UTM systems, which provide protection for UAVs, other infrastructure, and people, should follow a spiral approach, starting with low complexity operations and gradually building modules to support higher complexity operational concepts and requirements. Similarly, the evolution of tools for improving autonomous systems security will require a spiral approach as autonomous systems evolve.