Zero Vulnerability Technology: Leveraging Intelligent Scrambling for Next Generation Military Cybersecurity

ARMS Cyber Defense Inc.

Motivation

Think of this scenario: a terrorist cell is plotting an attack against the United States. During this time, a critical conversation is picked up giving a clue into who is involved in the operation, that an attack is imminent, and the urgent need to neutralize the threat before they reach American soil. A joint military operation is quickly dispatched to perform a covert mission to capture the bad actors. As such, every possible resource is deployed, human intelligence, special forces, and of course unmanned aerial vehicles (UAVs). Everything seems to be going according to plan until all of a sudden everything goes black. The UAV has gone down due to what seems like a power failure and a team of engineers is quickly dispatched to fix the problem, but they are too late. This lapse allows the leaders of the cell to escape and leaves martyrs at the compound to ambush the oncoming raid. It was later discovered after a lengthy investigation that the UAV was compromised from a cyber attack masked as a power failure - and it could have been prevented.

This scenario illustrates just how reliant US military missions are on technology. An operation is only as good as the intelligence that is presented and the ability to coordinate actions. UAVs provide key intelligence capabilities such as real-time imagery, intercepting attacker communications and movements, and in some cases lethal capabilities to protect our soldiers in battle. The traditional fear of UAVs dealt with the occurrence of faults or physical events such as being shot down by missiles. However, with the rise of military cyber capabilities, a more direct threat vector is popping up, and it's the possibility of getting hacked.

Adversaries are realizing the asymmetric advantage that cyberspace presents, providing a more economical and controlled strategy compared to traditional military operations. Countries that normally weren't a threat can now produce devastating damage while having a good chance of remaining undetected. Not only that, countries that we don't consider a threat could act as proxies to the highest paying bidder, leading to scenarios where the threat isn't a country on a map but rather a network of cells that operate independently of the soil they reside on. With the increased external communication channels integrated within our mission critical infrastructure in theater, attackers can hijack military assets from anywhere on Earth. It is now as critical as ever to rise to the challenge and secure our assets against new threats that previously did not exist. We are in a new era of warfare and the prosperity of our military and country depends on it.

Problem

UAVs are a prime example of a safety-critical Cyber-Physical System (CPS) where the tightly coupled nature between software and physical dynamics provides the ability for attackers to perform physical actuation through the use of cyber-attacks. The three most significant properties that normally define CPS are 1) a large amount of legacy software and outdated coding practices and languages, 2) highly predictable and reliable systems that leverage time-triggered real-time operating systems, and 3) highly available and standalone systems that behave safely without the need for human intervention [1]. Accelerated technological developments continue to leave last year's most secure systems vulnerable to the threats of next year, and often it is impossible to update the software to defend against new threats. In terms of security, the traditional standalone nature of UAVs means that designers weren't taking into account potential remote attack vectors, focusing instead on guaranteeing safety and reliability. As designs quickly shifted to include more communication channels and distributed architectures over the last decade, failure in updating the central codebases has led to a plethora of easily exploitable memory corruption vulnerabilities such as Buffer Overflows. By exploiting these vulnerabilities remotely, attackers can compromise a system and disrupt software availability, affecting the underlying safety properties.

Furthermore, due to the homogeneous architecture of software as well as significant dependencies on common third party libraries, if adversaries can discover a vulnerability, they can scale to compromise every deployed instance of that device. Lets go back to the example above with the terrorist cell mission, specifically with the UAV exploit. What wasn't noticed before the mission was that a previous UAV crashed into enemy territory, after which the firmware was extracted and reverse engineered. At this point, adversaries discovered vulnerabilities and crafted an exploit. Since the two UAVs contained the same software, the adversaries were able to successfully leverage this developed exploit in the respective mission scenario, allowing them to get away from allied troops and inflict maximum damage. This scenario isn't all that unrealistic, with high profile examples of UAVs crashing in enemy territory [2], and proof of concepts in the academic and government setting proving out the correlation between open source software in several classes of safety-critical CPS [3].

Current solutions are limited in their protection capability, mostly focusing on leveraging artificial intelligence to detect indicators of attacks. On CPS, which often are memory-constrained systems without GPU and high powered processors, these detection approaches often are unpractical. Further, at a more abstract level, by using reactionary defenses, by the time an attack is detected, it is already too late with the attacker having a foothold within the system. The priority then becomes focused on how to contain and mitigate the effects of attacks. A concept called Moving Target Defense (MTD) has been gaining a lot of attention within the academic and government communities over the last couple of years. Instead of reacting to attacks, MTD focuses on dynamically reconfiguring operating system and software properties to prevent attackers from obtaining the reconnaissance knowledge necessary for successful exploit development in the first place. Instead of having static, homogeneous systems and software, properties such as program function order, network protocols and ports, architectures, and data formats are diversified to form a "unique" system, making any previous generic exploits ineffective.

In terms of CPS and even traditional IT systems, MTD forms a good starting point in protecting against generic attacker campaigns. However, it is important to note that diversity is not the end all be all of a cybersecurity solution. It should instead be treated as another piece of the puzzle in the landscape of a defense-in-depth approach. The biggest limitation deals with the vulnerability space itself. Even though diversity prevents initial knowledge of an attack surface to be correct, it does not eliminate vulnerabilities, instead using a "security through obscurity" approach to move vulnerabilities around.

Technical Capabilities

The technical founders of ARMS have previously focused their efforts on offensive exploitation of adversary systems during their time in academia, government, and UARCs. Example proof of concepts included RUCKUS for rapid and scalable embedded firmware exploitation and geo-spatial analytics to convert communications into adversary pattern of life insights. The fundamental basis behind the success of all of these efforts comes down to the homogeneous and predictable architecture of adversary systems. Furthermore, if we can demonstrate these concepts then it is only common sense that adversaries are attempting to use the same techniques against our national security and military infrastructure. As such, the goal of ARMS is to address the problem of homogeneity head on by leveraging zero vulnerability technology and intelligent scrambling, raising the bar for attacker success and minimizing the incentive for targeting Department of Defense systems. To accomplish this, ARMS is creating solutions within three pillars: 1) Software Security, 2) Data Security, and 3) Communication Security.

1) Software Security

Motivated by the DARPA Cyber Grand Challenge (CGC), there have been a number of recent prototypes within the academic and government community demonstrating the ease of exploitation at scale through the use of heuristics focused on the similarity of software. Significant overlap of software has been found between proprietary systems such as automobiles and open source software which can lead to critical applications becoming hijacked through generalized exploits. State-of-the-art approaches to protecting against these exploitation techniques are based on static randomization, leveraging specialized passes to diversify program address space (function shuffling) at compile time. These solutions are a good start, but vulnerabilities still exist (they are just moved around). Manual reverse-engineering can reveal vulnerabilities to exploit, and because of the static program layout, adversaries have virtually unlimited time to exploit systems, especially through high availability SAAS software such as web servers.

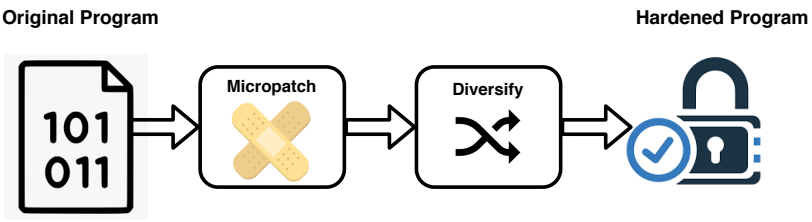

Figure 1: Software Security Process

The core technology that forms the backbone of ARMS' software security capability revolves around the concept of zero vulnerability technology. The premise behind zero vulnerability technology is that instead of solely relying on randomization with a security through obscurity approach, the goal is to leverage a hybrid approach, using micropatching to eliminate as many known vulnerabilities as possible while relying on randomization as a second layer of defense for protecting against unknown threats. As such, the goal is to target defense at the reconnaissance stage of the attack kill chain, preventing vulnerability discovery in the first place.

By leveraging the hybrid micropatching and randomization approach, we can increase the level of protection from the CGC exploit dataset from 5% (SelfRando and Multicompiler) to 75% [4]. Improvements in security protections of software applications can be broken down into two categories: 1) Minimizing the known set of vulnerabilities which limits the attack surface and 2) increase unpredictability to protect against unknown threats. We accomplish the first category by integrating micropatching to eliminate known memory corruption vectors and ROP chains, while accomplishing the later category with a vastly improved level of randomization.

Randomization is generally evaluated in terms of protection against ROP attacks, thus looking at the movement and breakup of gadgets with a program. The following proof illustrates how our technology can significantly improve protection compared to market leaders and current state of the art implementations. As such, we focus on ALSR which forms the backbone of Linux distributions, as well as the technology behind Multicompiler and SelfRando which form the backbone behind the market leading solutions provided by Polyverse and Runsafe Security. ASLR has been identified as having a gadget overlap of 15% between programs, while the remaining 85% of gadgets have the same relative offset but different base address. This means that if you find one gadget within the unmoved region, you then can infer the offset of all of the gadgets within that region and automatically adapt an exploit to be successful. This concept is known as gadget correlation and is the primary reason why current ASLR protections are not enough to provide adequate protection.With fine grained randomization such as function shuffling, the correlation between gadgets is removed and even knowing one offset within the program, doesn't get an attacker much closer to being successful. As such, each gadget is considered independent in terms of a probabilistic formulation. When evaluating Multicompiler and SelfRando against the DARPA CGC dataset, it was found that both of these technologies provide around 25% effectiveness against moving gadgets within a program [5]. Thus, even though there is a finer grained level of randomization, not as much of the program is being altered, leaving a conservative estimate of gadget overlap at 75%. Furthermore, with shuffling transformations, there is no guarantee that the number of ROP gadgets is actually decreased in the resulting process, meaning programs are potentially more vulnerable than before the transformation [6]. Finally, with ARMS' zero vulnerability approach, the gadget overlap that was evaluated over the same dataset is 2% meaning there is a 35x better randomization effectiveness compared to the market leaders. When observing how this factors into the likelihood of an exploit's success given knowledge of the location of a starting gadget, lets observe the following:

Theorem 1 *Assume ASLR has no fine grained randomization and that there is a 15% gadget overlap between programs, while the remaining 85% of gadgets are correlated, being translated by the same offset. Let the gadget overlap of SelfRando/Multicompiler be 85% while the gadget overlap of ARMS is 2%. Also, assume that the exploit is comprised of 5 gadgets which starts with gadget A and ends with subsequent gadgets sub. Assume that the adversary has discovered the location of gadget A.*

Let ***Event success*** *represent the event that the exploit can be executed successfully*
Let ***Event A*** *represent the event that gadget A resides in the gadget overlap region*
Let ***Event sub*** *represent the event that the subsequent exploit gadgets can be successfully executed*
Let ***ASLR*** *represent the likelihood of a successful attack against ASLR*
Let ***SelfRando*** *represent the likelihood of a successful attack against SelfRando/Multicompiler*
Let ***ARMS*** *represent the likelihood of a successful attack against ARMS.*
The equation to compute information disclosure resilience is given as follows:

$$P(success|gadgetA) = (P(A) * P(sub)) + (P(-A) * P(sub))$$

$$\boldsymbol{ASLR} = (.15 * .15^4) + (.85 * 1.0) = .85 = \boldsymbol{85\%}$$

$$\boldsymbol{SelfRando} = (.75 * .75^4) + (.25 * .75^4) = .31 = \boldsymbol{31\%}$$

$$\boldsymbol{ARMS} = (02 * .02^4) + (.98 * .02^4) = .02^4 = \boldsymbol{.000016\%}$$

2) Data Security

The next pillar of the ARMS technical capability suite is data security. Data security is a key area for enterprises and the military alike, ensuring the confidentiality and integrity of PII and sensitive information. Data breaches such as the OPM breach result in not just privacy loss, and potential identity fraud, but can have serious implications on mission success and the safety of government assets if there affiliation was discovered. As such, it is important whether in the Air Force's enterprise networks or on the battlefield with remote deployed equipment that data remain secure. Our view is that the main vulnerability to data is the homogeneous structure. Data sharding to a certain degree helps but if an adversary can gain access to the internal network, they potentially have full access to exfiltrate data at will. In our view, a new perspective should be taken, prioritizing the worst case scenario of mitigating adversary traversal and exfiltration once entrance is gained to military systems.

The goal of the ARMS data security capability is to provide a distributed data storage mechanism to introduce a degree of unpredictability to the attackers view. By distributing storage across multiple databases, hosts and

geographies, attacker exfiltration is limited and more fine tuned monitoring can be introduced including adding in honeypots and trap data files. Our distributed data storage is built on top of the Inter-Planetary File System (IPFS), bringing state of the art and novel academic concepts to industry. The second component of our data security capability revolves around data access vulnerabilities from web applications. By introducing taint analysis into common web server software, we can identify and mitigate potentially dangerous data access operations, consequently mitigating attacker attempts at injection based web attacks such as SQL Injection.

3) Communication Security

The final pillar of the ARMS technical capability suite is communication security. Communication security is prime in the battlefield, ensuring that key intelligence and command and control payloads aren't intercepted and reverse engineered by the adversary. There are two key categories of novel techniques that significantly improve the state of communication security: 1) Covert transmission and blending and 2) Distributed and unpredictable communications. In the first category, key research and developments have been made within the academic and government space in the area of covert communications, including by our technical founders. The problem with current communications strategies is that generally the focus is on encryption and ensuring that messages aren't encrypted. Although this is important, current communications are generally not very well disguised and transmit on single links, meaning that adversaries can sit back with unlimited time to reverse engineer encryption protocols. The key idea of our covert communication strategy is that in addition to encrypting current communications, new vectors should be leveraged to blend communications into "normal" traffic so that the adversary is unaware of communication attempts in the first place. Furthermore, by converting communications from a single link to distributed architecture similar to the Tor network, message routes can be randomized introducing a degree of unpredictability to the adversary. Our communications capabilities are currently at the research proof of concept stage but have been demonstrated on military and smart cities scenarios within a UARC setting.

Technology Readiness

ARMS capabilities have been built upon state of the art techniques within the academic, government, and military communities. The software security pillar is currently at a technology readiness level (TRL) of 5, having been demonstrated and proven on realistic infrastructure within the commercial sector. The data security capability is currently at a TRL 3 having been proven out within the laboratory environment, while the communication pillar is also at TRL 3. The underlying technology is architecture agnostic and can be easily transformed to meet the needs of the military.

Military Application

Next-generation warfighter information systems will be used to power multi-domain operations. Emphasized modernization efforts will be targeted towards advancing long-range precision fires, next-generation combat vehicles, future vertical lift, network, air, and missile defense and soldier lethality. To give us an advantage in all of our modernization efforts, it will be innovation in information systems infrastructure to ensure confidentiality, integrity, and availability that will truly allow warfighters to execute operations with groundbreaking applications and no concern of operational compromise. As legacy software runs the majority of existing systems, adversarial intelligence services seek to find vulnerabilities that can later be patched by existing vendors when they are found. This cycle of response lacks the proactive and real-time cyber defense needed to protect the interconnected battlefield of tomorrow. We propose moving target defenses with an impact that is two-fold for military technology. 1) Dramatically reducing the economy of scale for attackers and taking away their adversarial advantage and 2) Providing an on-system application intelligent mitigation capability in line with the current state of the art technologies, giving systems additional mitigation against zero-day attacks with in between elongated patching cycles. Such an approach will combine autonomy with security, allowing for detection with zero false positives, isolation, and remediation actions to take place on-board before operators on the ground even realize an attack took place.

The scope will continue to expand while the traditional "patch when a problem is found" methodology for security will be outpaced as adversaries continue to develop cyber capabilities. Additionally, the problem is amplified by the push for an interconnected battlespace with effective real-time command and control. Ultimately, we aim to impact operations in the defense against adversarial nation-states in the domains of long-, mid-, and short-range weapons systems, conventional forces, integrated air defenses, cyber-attacks, and denials of space-based capabilities, such as reconnaissance, navigation, and communications, as well as an array of informational tools.

Dual Use Market Opportunity

Through the National Science Foundation's ICORPS program, a product-market fit was identified for product managers at medium-sized companies ($20 Million - $100 Million annual revenue) with cloud-based Software-as-a-Service (SAAS) products. These companies primarily leverage Platform-as-a-Service (PAAS) solutions to aid in preventing vulnerabilities and bugs within their products. As a result, this reduces the amount of time and effort that is spent reacting to cyber-attacks and issues versus building new features. Furthermore, these solutions can further be categorized into Enterprise and Internet-of-Things (IoT) applications where IoT devices are most at risk due to the set and forget deployment mentality. This opportunity provides a total addressable market of $50 Billion in the United States with a 9.3% CAGR, a $2 Billion service addressable market within our target market, and an estimated $34 Million service obtainable market with a 1.7% penetration rate.

Commercialization Pathway

ARMS capabilities serve as the foundation for several delivery vectors for the customer. For the software security pillar, ARMS provides three solutions: 1) Container Security, 2) Endpoint Security, and 3) Ransomware Protection client. It is important to note in contrast to existing container and endpoint security commercial solutions which focus on monitoring, ARMS solutions build security into the respective entities, ensuring that the underlying software is protected. For data security, ARMS provides a web application security plugin as well as a distributed database storage tool. Finally, for communication security, ARMS provides custom developed algorithms for next generation SDN communication strategies. These solutions are marketed to both the enterprise domain as well as directly to the developer to integrate security at the beginning stages of the supply chain.

Team

Our team is comprised of individuals with impeccable backgrounds that includes inventing CNO capabilities within the NSA and serving in the military in combat zones in Afghanistan and Iraq. The technical founders include multiple cybersecurity PhD graduates and candidates and are thought leaders in the field, having spent the last 5 years in the academic space building out the respective MTD capabilities with funding from the NSA and AFRL. They have published dozens of papers, presented at numerous conferences, received two best paper awards from the NSA, and served as SMEs to high level US Government stakeholders. The rest of the founding team includes a MBA graduate and startup veterans who have experience developing and scaling cybersecurity SAAS products within multiple industries. Finally, our advising team includes key thought leaders within the Defense innovation ecosystem, as well as former executives with 9 figure exits.

References

[1] B. Potteiger, F. Cai, Z. Zhang, A. Dubey, and X. Koutsoukos, "Security in mixed time and event triggered cyber-physical systems using moving target defense," in *2020 International Symposium on Real Time Distributed Computing (ISORC), Nashville, TN*. IEEE, 2020.

[2] S. Shane and D. E. Sanger, "Drone crash in iran reveals secret us surveillance effort," *The New York Times*, vol. 7, 2011.

[3] B. Potteiger, J. Mills, D. Cohen, and P. Velez, "Ruckus: a cybersecurity engine for performing autonomous cyber-physical system vulnerability discovery at scale," in *Proceedings of the 7th Symposium on Hot Topics in the Science of Security*, 2020, pp. 1–10.

[4] D. Kelly, C. Wellons, J. Coffman, and A. Gearhart, "Automatically validating the effectiveness of software diversity schemes," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks–Supplemental Volume (DSN-S)*. IEEE, 2019, pp. 1–2.

[5] M. S. Ahmed, Y. Xiao, G. Tan, K. Snow, F. Monrose, and D. Yao, "Poster: Quantifying the impact of fine-grained code randomization on attack surface reduction," *IEEE SecDev*, 2019. [Online]. Available: https://secdev.ieee.org/wp-content/uploads/2019/09/aslr-poster.pdf

[6] J. Coffman, D. M. Kelly, C. C. Wellons, and A. S. Gearhart, "Rop gadget prevalence and survival under compiler-based binary diversification schemes," in *Proceedings of the 2016 ACM Workshop on Software PROtection*, 2016, pp. 15–26.